Naïve Bayes and Evaluating Text Classifiers

Natalie Parde

UIC CS 421

This Week's Topics

Tuesday

- N-gram language modeling
- Evaluating LMs
- Improving n-gram LMs

Thursday

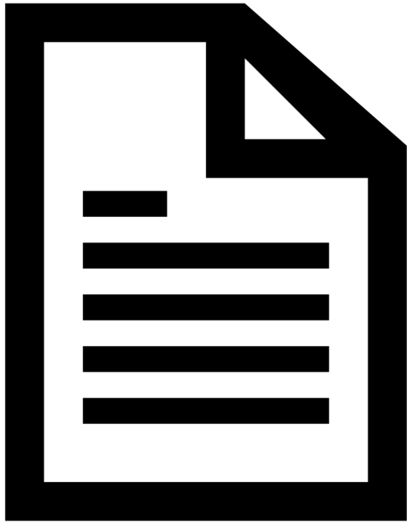
- Text classification
- Naïve Bayes
- Evaluating text classifiers

What is text classification?

The process of deciding the **category** of an **instance**

- **Instance:** A document, sentence, word, image, transcript, or other individual language sample

Fundamental to many NLP tasks



Common Applications of Text Categorization

- Spam detection

Dear Dr. Parde Natalie,

Journals of [redacted] are devoted to the principles and core ethics of Open Access. Our goal is to create an egalitarian platform to enable unrestricted knowledge exchange among researchers, experts, and curious minds alike. We are breaking away from old traditions to open the doors wide open to people from all corners of the world. First and foremost, we respect the author's right of ownership to the articles they create.

Our publications provides a global platform and a targeted source for publishing original research. Do you have an article/ebook ready for submission? We are accepting submissions ***with 139 USD as pocessing charges for the Open Access Week 2019***. Please feel free to revert with any further questions about the special themes.

Looking forward!

[redacted]
Editorial Coordinator
[redacted]

Spam

Not Spam

Common Applications of Text Categorization

- Spam detection
- Authorship attribution

"What can be the meaning of that emphatic exclamation?" cried he. "Do you consider the forms of introduction, and the stress that is laid on them, as nonsense? I cannot quite agree with you _there_. What say you, Mary? For you are a young lady of deep reflection, I know, and read great books and make extracts."

Mary wished to say something sensible, but knew not how.

"While Mary is adjusting her ideas," he continued, "let us return to Mr. Bingley."

"I am sick of Mr. Bingley," cried his wife.

"The world is full of obvious things which nobody by any chance ever observes. Where do you think that I have been?"

"A fixture also."

"On the contrary, I have been to Devonshire."

"In spirit?"

Voltaire

Sir Arthur Conan Doyle

Jane Austen

Common Applications of Text Categorization

- Spam detection
- Authorship attribution
- Sentiment analysis

Natalie's poem about Halloween was really dreadful. The word "Halloween" doesn't even rhyme with "trick or treat!" She should stick to writing NLP programs.

Natalie's poem about Halloween was a true delight! The way she rhymed "Halloween" with "trick or treat" was artful and unexpected. I can't wait to read what she writes next!

Natalie wrote a poem about Halloween. She wrote it as if the words "Halloween" and "trick or treat" rhyme with one another. It was her first poem.

Positive

Negative

Neutral

Common Applications of Text Categorization

- Spam detection
- Authorship attribution
- Sentiment analysis
- Domain identification

"What can be the meaning of that emphatic exclamation?" cried he. "Do you consider the forms of introduction, and the stress that is laid on them, as nonsense? I cannot quite agree with you _there_. What say you, Mary? For you are a young lady of deep reflection, I know, and read great books and make extracts."

Mary wished to say something sensible, but knew not how.

"While Mary is adjusting her ideas," he continued, "let us retu…

Mr. Bingley."

"I am sick of Mr. Bingley," cried his wi…

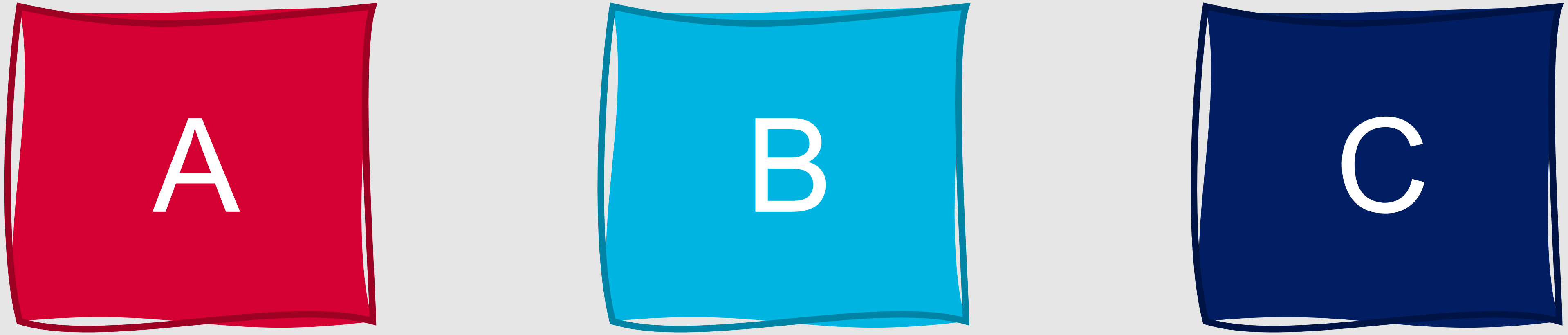
The model takes two inputs: the tokenized interview, and the corresponding POS tag list. Word embeddings for the interview text tokens are computed using pre-trained 300 dimensional GloVe embeddings trained on the Wikipedia 2014 and Gigaword 5 dataset (Pennington et al., 2014). The POS tag for each word is represented as a one-hot encoded vector. The word embeddings and POS vectors are input to two different CNNs utilizing the same architecture, and the output of the two CNNs is then flattened and given as input to a bidirectional LSTM with an attention mechanism.

Fiction

Academic

Classification

- Goal:
  - Take a single **observation**
  - Extract some useful **features**
  - Classify the observation into one of a set of discrete classes based on those features



How is classification performed?

- Rule-based methods
- Statistical methods
  - Including feature-based methods and deep learning methods

Rule-Based Classification Methods

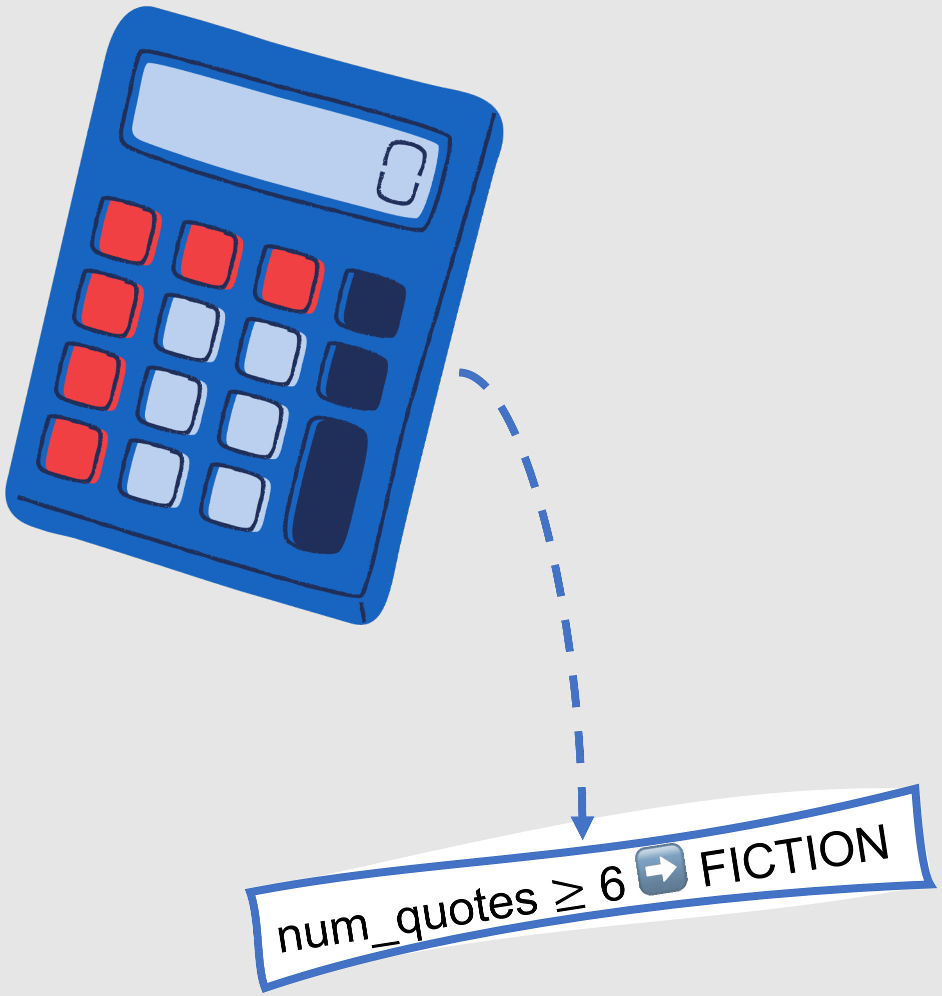
- **Manually create a set of rules** based on expected differences among features from different classes
- Use that information to classify test data

If text contains "love" ➡️ POSITIVE

If text contains quotation marks ➡️ FICTION

Statistical Classification Methods

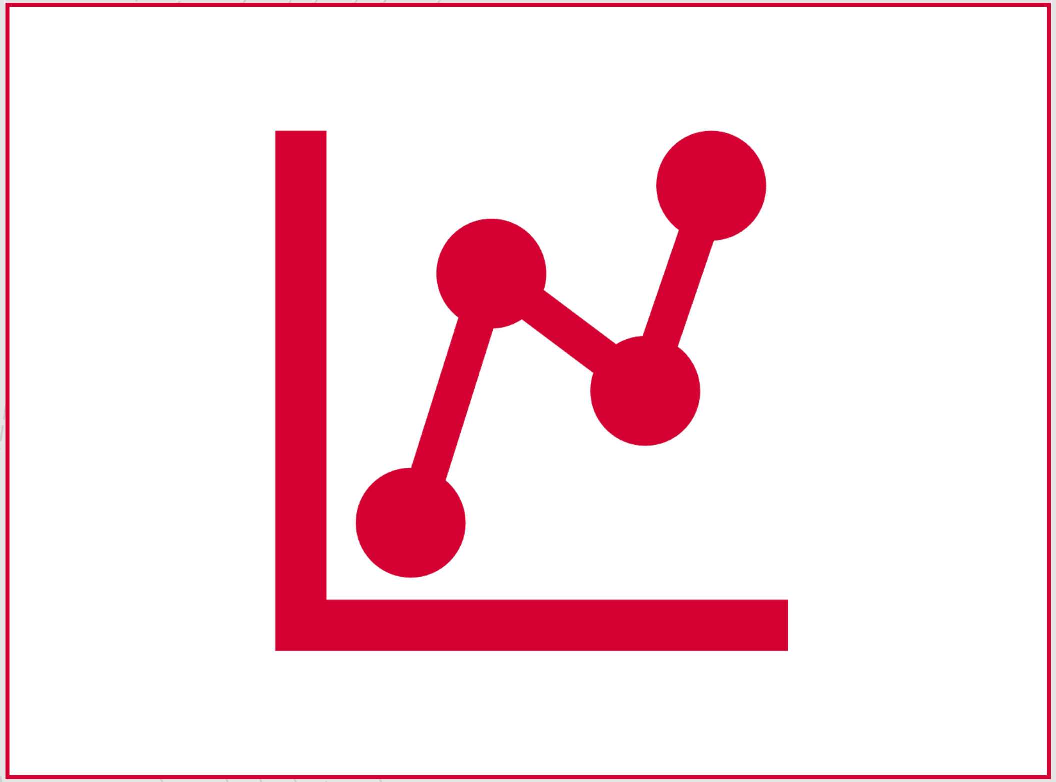
- Automatically **learn which characteristics best distinguish different classes** from one another based on a collection of training data
- Use that information to classify test data



Is rule-based or statistical classification better?

- In modern computing environments (i.e., scenarios with plentiful data), **statistical classification is generally a better choice**
- If data is really limited, rule-based methods may work better

Language is dynamic.



- Word uses can change over time, and so can data
  - He *ghosted* me
  - *Covid-19*
- With rule-based methods, we have to write new rules to accommodate changes in language
  - We also might miss some changes!
- Statistical methods can be automatically retrained when new data is available

Types of Statistical Classification Techniques

Supervised learning: Statistical classification *with* a labeled training set

Unsupervised learning: Statistical classification *without* a labeled training set

Formal Definition: Supervised Learning

- Take an input *x* from a set of inputs $x \in X$
- Consider a fixed set of output classes $y \in Y$, where $Y = \{y_1, y_2, \ldots, y_M\}$
  - In text classification, we may refer to *x* as *d* (for "document") and *y* as *c* (for "class")
- We have a training set of *N* documents, each of which have been manually labeled with a class: $\{(d_1, c_1), \ldots, (d_N, c_N)\}$
- Goal: Learn a classifier that is capable of mapping from a new document *d* to its correct class $c \in C$ (equivalently, learning to predict the correct class $y \in Y$ for an input $x \in X$)

Types of Supervised Probabilistic Classification Models

- Naïve Bayes
- Logistic regression
- Support vector machine
- K-nearest neighbors
- Multilayer perceptrons (neural networks)
- …and many more!

These classification models can be further subdivided into groups.

- **Generative classifiers** build models of how classes could generate input data
  - Given an observation, they return the class most likely to have generated it
- **Discriminative classifiers** learn which features from the input are most useful to discriminate between different possible classes
  - Given an observation, they return the best match based on these weighted features

This Week's Topics

Tuesday

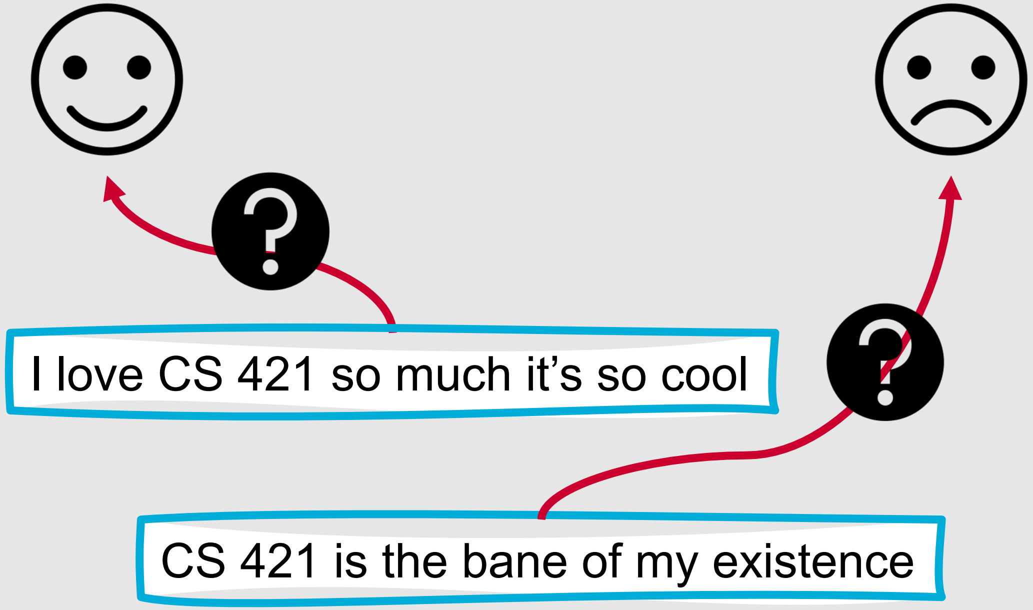
- N-gram language modeling
- Evaluating LMs
- Improving n-gram LMs

Thursday

- Text classification
- Naïve Bayes
- Evaluating text classifiers

One probabilistic, generative classifier?

- Naïve Bayes: A **probabilistic classifier** that learns to **predict labels** for new documents



Why is it “Naïve” Bayes?

- Naïve Bayes classifiers make a naïve assumption that features don’t impact each other and instead **are all independent from one another**
- Is this really the case?
  - No! As we’ve seen with language models, words are dependent on their contexts
  - However, Naïve Bayes classifiers still perform reasonably well despite this assumption

Types of Naïve Bayes Classifiers

Gaussian Naïve Bayes: Assumes the outcomes for the input data are normally distributed along a continuum

Multinomial Naïve Bayes: Assumes the outcomes for the input data follow a multinomial distribution (there is a discrete set of possible outcomes)

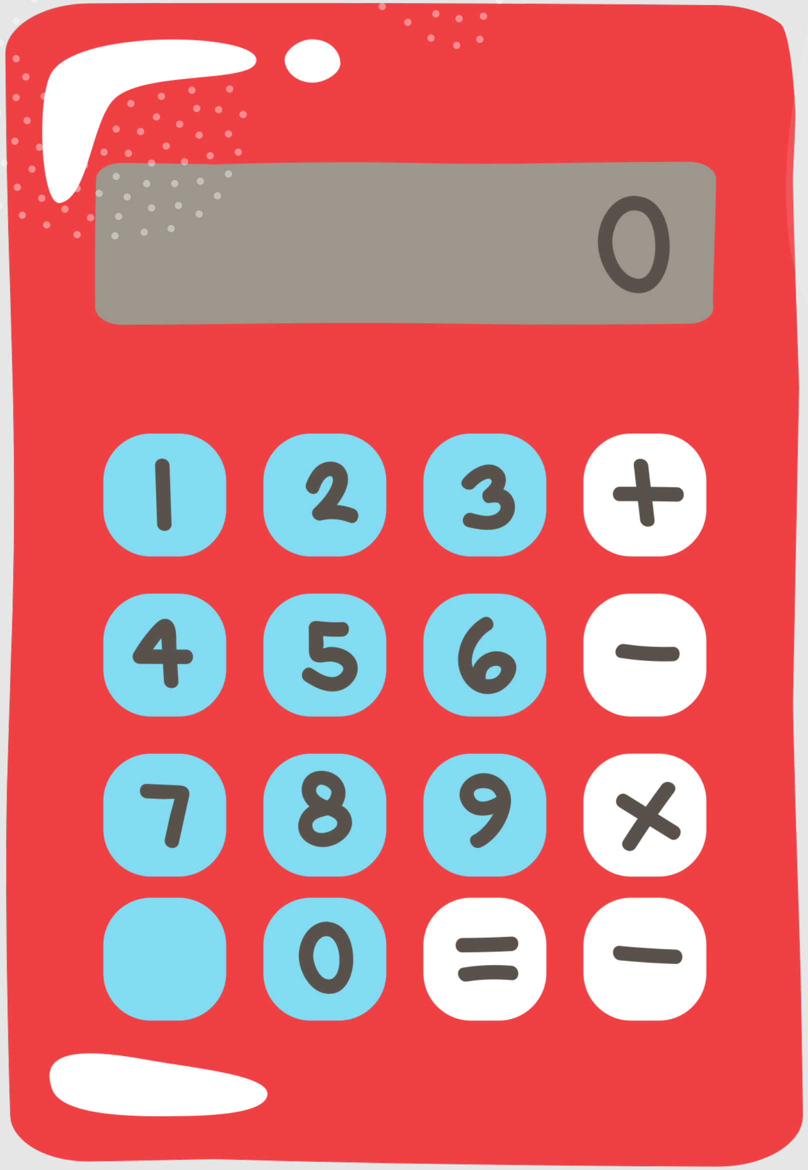
Binomial Naïve Bayes: Assumes the outcomes for the input data follow a binomial distribution (there are two possible outcomes)

How does naïve Bayes work?

- For a document d, out of all classes $c \in C$ the classifier returns the class c' which has the maximum **posterior probability**, given the document
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c|d)$

Posterior probabilities are computed using Bayesian inference.

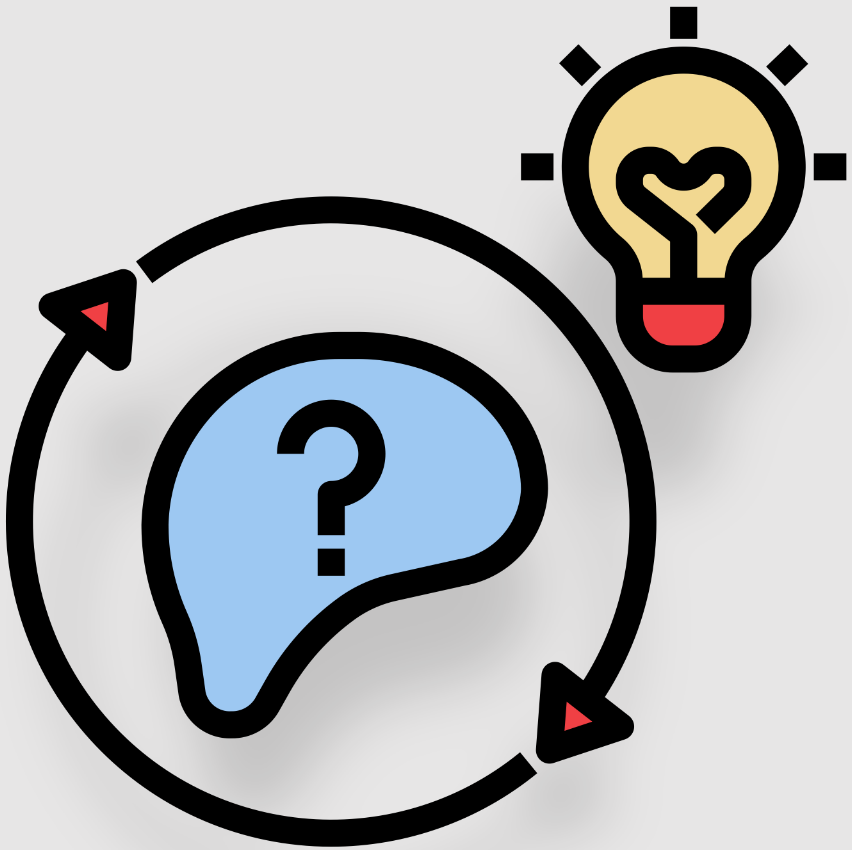
- Bayesian inference uses **Bayes' rule** to transform probabilities like those shown previously into other probabilities that are easier or more convenient to calculate
- Bayes' rule:
  - $P(y|x) = \frac{P(x|y)P(y)}{P(x)}$



Applying Bayesian inference in Naïve Bayes

- If we take Bayes' rule:
  - $P(y|x) = \frac{P(x|y)P(y)}{P(x)}$
- And substitute it into our previous equation:
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c|d)$
- We get the following:
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c|d)$

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



We can simplify this….

- Drop the denominator P(d)
  - We'll be computing $\frac{P(d|c)P(c)}{P(d)}$ for each class, but P(d) doesn't change for each class
    - We're always asking about the most likely class for the same document d
- Thus:
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$

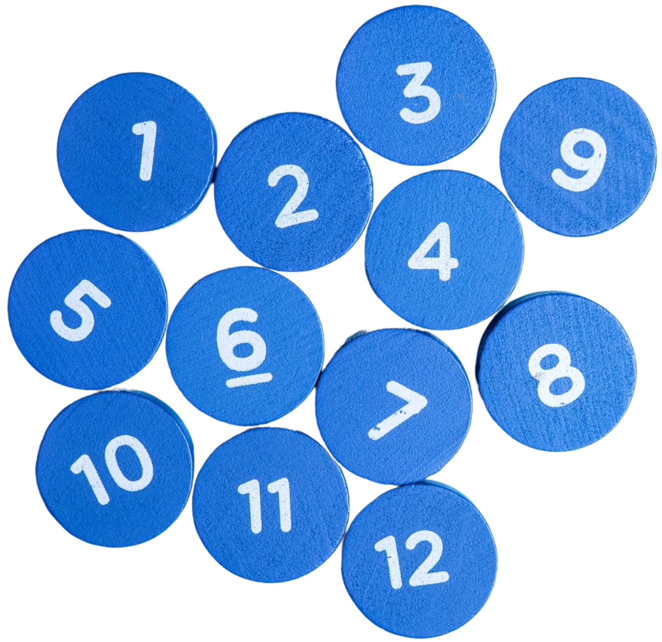
So, the most probable class *c'* given some document *d* is the class that has the highest product of two probabilities.

- **Prior probability** of the class P(*c*)
- **Likelihood** of the document P(*d*|*c*)

$$\boxed{c' = \operatorname*{argmax}_{c \in C} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}}$$

To find these probabilities….

- We need to represent our text sample using one or more numbers
- These numbers can represent different **features** of the data



Example Feature Representation

- Represent each document as a **bag of words**
  - Unordered set of words and their frequencies
- Decide how likely it is that a document belongs to a class based on its distribution of **word frequencies**

Usman's poem about Thanksgiving rivaled Natalie's notorious Halloween poem.

's	2
poem	2
Usman	1
about	1
Thanksgiving	1
rivaled	1
Natalie	1
notorious	1
Halloween	1

More formally, this means that….

- Bags of words are sets of features {f_1, f_2, …, f_n}, where each feature f corresponds to the frequency of one of the words in the vocabulary
- Therefore:
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c) = \underset{c \in C}{\operatorname{argmax}} \underbrace{P(f_1, f_2, \dots, f_n|c)}_{\text{likelihood}} \underbrace{P(c)}_{\text{prior}}$
- The Naïve Bayes assumption means that we can "naïvely" multiply our probabilities for each feature together, since they're assumed to be independent
- Therefore:
  - $P(f_1, f_2, \dots, f_n|c) = P(f_1|c) * P(f_2|c) * \cdots * P(f_n|c)$

Putting this all together….

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

$$= \underset{c \in C}{\operatorname{argmax}} P(f_1, f_2, \ldots, f_n|c)\, P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{f \in F} P(f|c)$$

How do we use our Naïve Bayes classifier?

- For a new text document, extract features and compute:
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in N} P(f_i|c)$
- To avoid underflow (generating numbers that are too tiny to be adequately represented) and increase speed, we can also do these computations in log space:
  - $c' = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i \in N} \log P(f_i|c)$

When viewed in log space, we can see that this is a linear classifier.

- A linear classifier predicts classes as a **linear function** of the input features
  - $c' = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i \in T} \log P(w_i|c)$
- Some linear classifiers:
  - Naïve Bayes
  - Logistic Regression

How do we train a Naïve Bayes classifier?

- We need to learn P(c) and P(f_i|c) based on available data
- To compute P(c), we figure out what percentage of the instances in our training set are in class c
  - Let N_c be the number of instances in our training data with class c
  - Let N_{doc} be the total number of instances, or documents
  - $P(c)' = \frac{N_c}{N_{doc}}$
- To compute P(f_i|c)….
  - **Maximum likelihood estimates!**

Naïve Bayes Model Training

- To compute P(f_i|*c*), find the fraction of times f_i appears among all documents of class *c*
    1. Concatenate all instances from class *c* into one big document of text
    2. Find the frequency of f_i in this document to find the maximum likelihood estimate:
        - $P(f_i|c)' = \frac{count(f_i,c)}{\sum_{f \in V} count(f,c)}$
            - Note: Since we're assuming features are words in a bag-of-words model, *V* is the set of all word types across all classes (not just the word types in class *c*)

To avoid having a single zero probability "zero out" the entire product, we can apply smoothing techniques.

- Simple, common solution: Laplace (add-one) smoothing
  - $P(f_i|c)'$

$$= \frac{count(f_i,c)+1}{\sum_{f\in V}(count(f,c)+1)}$$

$$= \frac{\boldsymbol{count(f_i,c)+1}}{\boldsymbol{\sum_{f\in V}(count(f,c))+|V|}}$$

Other scenarios to address:

- **Unknown words**
  - Solution: Ignore words that didn't exist in the training data (remove from test document + do not compute any probabilities for them)
- **Stopwords**
  - Ignore very frequent words like *a* and *the* in many cases using an automatically or manually defined stopword list

Example: Naïve Bayes

Natalie was soooo thrilled that Usman had a famous new poem.

She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.

Usman was happy that his poem about Thanksgiving was so successful.

He congratulated Natalie for getting #2 on the bestseller list.

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Natalie was soooo thrilled that Usman had a famous new poem.

Sarcastic

She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.

Sarcastic

Usman was happy that his poem about Thanksgiving was so successful.

Not Sarcastic

Not Sarcastic

He congratulated Natalie for getting #2 on the bestseller list.

Natalie told Usman she was soooo totally happy for him.

?

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What is the prior probability for each class?
  - $P(c)' = \frac{N_c}{N_{doc}}$

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What is the prior probability for each class?
  - $P(c)' = \frac{N_c}{N_{doc}}$
- P(Sarcastic) = 2/4 = 0.5
- P(Not Sarcastic) = 2/4 = 0.5

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What is the prior probability for each class?
  - $P(c)' = \frac{N_c}{N_{doc}}$
- P(Sarcastic) = 2/4 = 0.5
- P(Not Sarcastic) = 2/4 = 0.5
- Note: This means we have a **balanced training set**
  - Balanced: An equal number of samples for each class

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- Taking a closer look at our data, let's remove:
  - Stop words
  - Unknown words

Natalie told Usman she was soooo totally happy for him.

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

- Taking a closer look at our data, let's remove:
  - **Stop words**
  - Unknown words

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- Taking a closer look at our test instance, let's also remove:
  - Stop words
  - **Unknown words**

Natalie told Usman she was soooo totally happy for him.

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What are the likelihoods from the training set for the remaining words in the test instance?
  - $P(w_i|c)' = \frac{count(w_i,c)}{\sum_{w \in V} count(w,c)}$

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What are the likelihoods from the training set for the remaining words in the test instance?
  - $P(w_i|c)' = \frac{count(w_i,c)+1}{(\sum_{w \in V} count(w,c))+|V|}$
  - P("Natalie"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Natalie"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$

Make sure to use smoothing!

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What are the likelihoods from the training set for the remaining words in the test instance?
  - $P(w_i|c)' = \frac{count(w_i,c)+1}{(\sum_{w \in V} count(w,c))+|V|}$
  - P("Natalie"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Natalie"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$
  - P("Usman"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Usman"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What are the likelihoods from the training set for the remaining words in the test instance?
  - $P(w_i|c)' = \frac{count(w_i,c)+1}{(\sum_{w \in V} count(w,c))+|V|}$
  - P("Natalie"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Natalie"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$
  - P("Usman"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Usman"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$
  - P("soooo"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("soooo"|Not Sarcastic) = $\frac{0+1}{12+21} = 0.030$

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- What are the likelihoods from the training set for the remaining words in the test instance?
  - $P(w_i|c)' = \frac{count(w_i,c)+1}{(\sum_{w\in V} count(w,c))+|V|}$
  - P("Natalie"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Natalie"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$
  - P("Usman"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Usman"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$
  - P("soooo"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("soooo"|Not Sarcastic) = $\frac{0+1}{12+21} = 0.030$
  - P("totally"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("totally"|Not Sarcastic) = $\frac{0+1}{12+21} = 0.030$

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

- What are the likelihoods from the training set for the remaining words in the test instance?
  - $P(w_i|c)' = \frac{count(w_i,c)+1}{(\sum_{w \in V} count(w,c))+|V|}$
  - P("Natalie"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Natalie"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$
  - P("Usman"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("Usman"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$
  - P("soooo"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("soooo"|Not Sarcastic) = $\frac{0+1}{12+21} = 0.030$
  - P("totally"|Sarcastic) = $\frac{1+1}{15+21} = 0.056$
  - P("totally"|Not Sarcastic) = $\frac{0+1}{12+21} = 0.030$
  - P("happy"|Sarcastic) = $\frac{0+1}{15+21} = 0.028$
  - P("happy"|Not Sarcastic) = $\frac{1+1}{12+21} = 0.061$

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- Given all of this information, how should we classify the test sentence?
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in T} P(w_i|c)$

Word	P(Word|Sarcastic)	P(Word|Not Sarcastic)
Natalie	0.056	0.061
Usman	0.056	0.061
soooo	0.056	0.030
totally	0.056	0.030
happy	0.028	0.061

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- Given all of this information, how should we classify the test sentence *s*?
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in T} P(w_i|c)$
  - P(Sarcastic)*P(*s*|Sarcastic) = 0.5 * 0.056 * 0.056 * 0.056 * 0.056 * 0.028 = 1.377 * 10^{-7}

Word	P(Word|Sarcastic)	P(Word|Not Sarcastic)
Natalie	0.056	0.061
Usman	0.056	0.061
soooo	0.056	0.030
totally	0.056	0.030
happy	0.028	0.061

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- Given all of this information, how should we classify the test sentence *s*?
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in T} P(w_i|c)$
  - P(Sarcastic)*P(*s*|Sarcastic) = 0.5 * 0.056 * 0.056 * 0.056 * 0.056 * 0.028 = 1.377 * 10^{-7}
  - P(Not Sarcastic)*P(*s*|Not Sarcastic) = 0.5 * 0.061 * 0.061 * 0.030 * 0.030 * 0.061 = 1.021 * 10^{-7}

Word	P(Word|Sarcastic)	P(Word|Not Sarcastic)
Natalie	0.056	0.061
Usman	0.056	0.061
soooo	0.056	0.030
totally	0.056	0.030
happy	0.028	0.061

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

Training	
Document	**Class**
Natalie was soooo thrilled that Usman had a famous new poem.	Sarcastic
She was totally 100% not annoyed that it had surpassed her poem on the bestseller list.	Sarcastic
Usman was happy that his poem about Thanksgiving was so successful.	Not Sarcastic
He congratulated Natalie for getting #2 on the bestseller list.	Not Sarcastic
Test	
Document	**Class**
Natalie told Usman she was soooo totally happy for him.	?

- Given all of this information, how should we classify the test sentence *s*?
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in T} P(w_i|c)$
  - **P(Sarcastic)*P(*s*|Sarcastic) = 0.5 * 0.056 * 0.056 * 0.056 * 0.056 * 0.028 = 1.377 * 10^{-7}**
  - P(Not Sarcastic)*P(*s*|Not Sarcastic) = 0.5 * 0.061 * 0.061 * 0.030 * 0.030 * 0.061 = 1.021 * 10^{-7}

Word	P(Word|Sarcastic)	P(Word|Not Sarcastic)
Natalie	0.056	0.061
Usman	0.056	0.061
soooo	0.056	0.030
totally	0.056	0.030
happy	0.028	0.061

P(Sarcastic) = 0.5
P(Not Sarcastic) = 0.5

Natalie told Usman she was soooo totally happy for him.

Sarcastic

Optimizing for Specific Tasks

- There are a variety of task-specific ways to improve performance with this model
- You may want to specifically encode:
  - Whether a feature exists in the data (rather than how many times)
  - Whether specific types of words (e.g., **negation**) are present

What if we don't have enough information to train an accurate Naïve Bayes classifier for a task?

- We can derive alternate or additional features (not word counts) from external **lexicons**
  - For example, add a feature that is counted whenever a word from a specific lexicon occurs
- **Lexicons** generally contain annotated characteristics (e.g., sentiment labels) for a list of words
- For sentiment analysis:
  - Linguistic Inquiry and Word Count (http://liwc.wpengine.com/)
  - Opinion Lexicon (https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon)
  - MPQA Subjectivity Lexicon (https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/)

Whether this works well may depend on data sparsity.

Large dataset:	Small dataset:
• Using many features will work better than just using a few binary features (allows for the classifier to learn more complex ways to discriminate between classes)	• Using a smaller number of more general features may work better (allows for the classifier to learn meaningful differences, rather than making predictions based on one or two occurrences of a given feature)

This Week's Topics

Tuesday

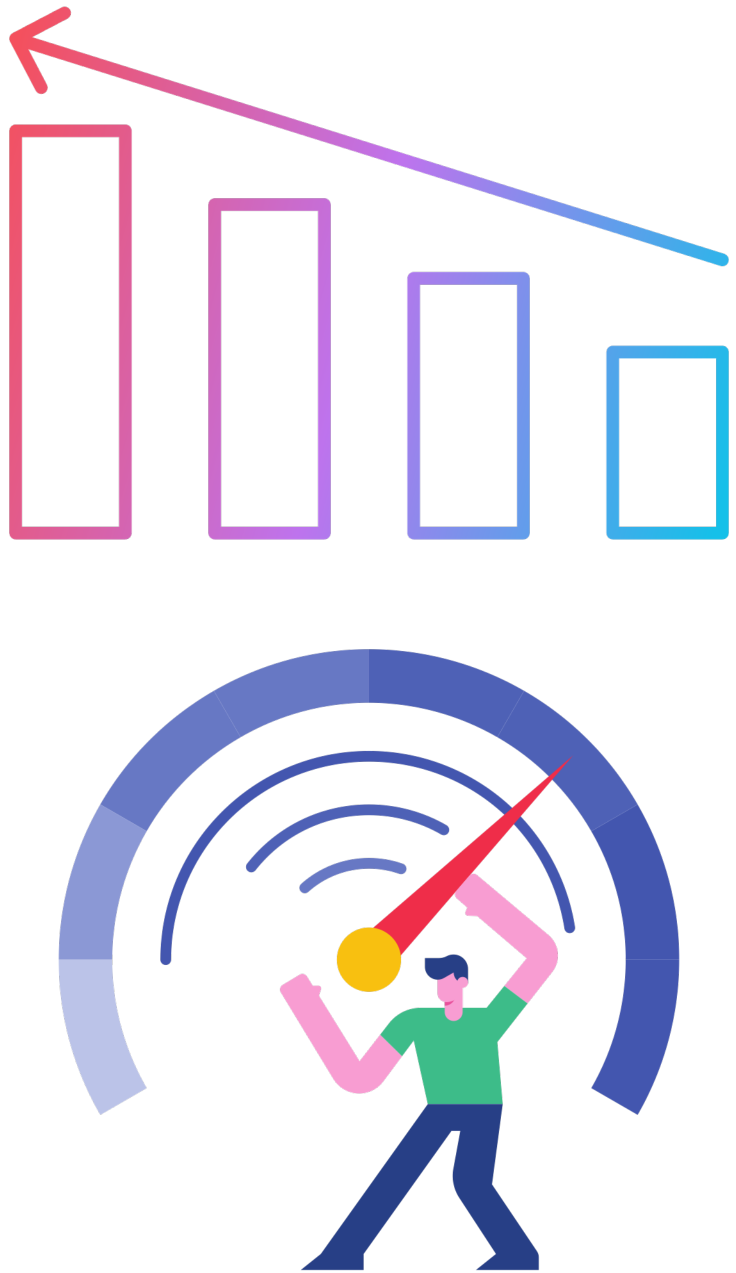
- N-gram language modeling
- Evaluating LMs
- Improving n-gram LMs

Thursday

- Text classification
- Naïve Bayes
- Evaluating text classifiers

We've learned a bit about text classification now….

How can we measure the performance of our models?



Gold Labels

- Before determining anything, we need some sort of basis upon which to make our comparisons
  - *Is "Sarcastic" the correct label for "Natalie told Usman she was soooo totally happy for him." ?*
- We can acquire **gold standard labels** from human annotators

Does it matter who our annotators are?

- Depends on the task
- For complex tasks, you may want to recruit experts
  - Rating translation quality
  - Labeling pedagogical strategies in teacher-student interactions
- For simpler tasks, you can probably recruit non-experts
  - Deciding whether text is sarcastic or non-sarcastic
  - Deciding whether a specified event takes place before or after a second event

Contingency Tables

- Once we have our gold standard labels (either from an existing dataset, or after collecting our own), we can begin comparing **predicted** and **actual** labels
- To do this, we can create a **contingency table** or **confusion matrix**

Contingency Tables

- In a contingency table, each cell labels a set of possible outcomes
- These outcomes are generally referred to as:
  - **True positives**
    - Predicted true and actually true
  - **False positives**
    - Predicted true and actually false
  - **True negatives**
    - Predicted false and actually false
  - **False negatives**
    - Predicted false and actually true

Actual

Predicted		
	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

We can compute a variety of metrics using contingency tables.

Precision

Recall

F-Measure

Accuracy

Actual

Predicted		
	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Accuracy

- **Accuracy:** The percentage of all observations that the system labels correctly
- Accuracy = $\frac{tp+tn}{tp+fp+tn+fn}$

Why not just use accuracy and be done with it?

- This metric can be unreliable when dealing with unbalanced datasets!
  - Imagine that we have 999,900 non-sarcastic sentences, and 100 sarcastic sentences
  - Our classifier might decide to just predict “non-sarcastic” every time to maximize its expected accuracy
    - 999900/1000000 = 99.99% accuracy
  - However, such a classifier would be useless …it would never tell us when a sentence *is* sarcastic

What are some more useful alternative metrics?

Precision

Recall

F-Measure

	Actual	
Predicted	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Precision

- **Precision:** Of the instances that the system predicted to be positive, what percentage actually are?
- Precision = $\frac{\text{tp}}{\text{tp}+\text{fp}}$

Actual

Predicted		
	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Recall

- **Recall:** Of the instances that actually are positive, what percentage did the system predict to be?
- Recall = $\frac{tp}{tp+fn}$

Precision and recall both emphasize a specific class of interest.

- For example:
  - **Sarcastic** or Non-Sarcastic
  - Positive or **Negative**
- In our problematic example case, precision and recall for the positive (sarcastic) case would both be 0
  - Precision = 0/(0+0) = 0
  - Recall = 0/(0+100) = 0

Actual

Predicted

TP: 0	FP: 0
FN: 100	TN: 999,900

Which is more useful: Precision or recall?

- Depends on the task!
- If it's more important to maximize the chances that all predicted true values really are true, at the expense of predicting some of the true values as false, focus on precision
- If it's more important to maximize the chances that all true values are predicted to be true, at the expense of predicting some false values to be true as well, focus on recall

What if both are important?

- **F-measure** combines aspects of both **precision** and **recall** by computing their weighted harmonic mean
  - $F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R}$
- The β parameter weights the importance of precision and recall, depending on the needs of the application
  - $\beta > 1$ means that recall is more important
  - $\beta < 1$ means that precision is more important
  - $\beta = 1$ means that the two are equally important

F-Measure

- Most commonly, researchers set β = 1 to weight precision and recall equally
- In this case, the metric is generally referred to as F_1
  - $F_1 = \frac{(1^2+1)PR}{1^2P+R} = \frac{2PR}{P+R}$
  - Note: With this equation, the lower of the two numbers will factor slightly more heavily into the final score

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	
Oh yay more things to grade!!!	Sarcastic	
Oh yay my new subscription box arrived!!!	Not Sarcastic	
Where is the closest coffee shop?	Not Sarcastic	
I just love large group meetings.	Sarcastic	

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow’s 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

TP: ?	FP: ?
FN: ?	TN: ?

Positive Class: Sarcastic

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

	Actual	
Predicted	TP: 1	FP: ?
	FN: ?	TN: ?

Positive Class: Sarcastic

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted		
	TP: 1	FP: 1
	FN: ?	TN: ?

Positive Class: Sarcastic

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

TP: 1	FP: 1
FN: 3	TN: ?

Positive Class: Sarcastic

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

TP: 1	FP: 1
FN: 3	TN: 2

Positive Class: Sarcastic

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

TP: 1	FP: 1
FN: 3	TN: 2

Positive Class: Sarcastic

Precision = $\frac{tp}{tp+fp} = \frac{1}{1+1} = 0.5$

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Predicted \ Actual		
	TP: 1	FP: 1
	FN: 3	TN: 2

Positive Class: Sarcastic

Precision = $\frac{tp}{tp+fp} = \frac{1}{1+1} = 0.5$

Recall = $\frac{tp}{tp+fn} = \frac{1}{1+3} = 0.25$

Example: Precision, Recall, and F_1

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted		
	TP: 1	FP: 1
	FN: 3	TN: 2

Positive Class: Sarcastic

Precision = 0.5

Recall = 0.25

$$F_1 = \frac{(1^2+1)PR}{1^2P+R} = \frac{2PR}{P+R} = \frac{2*0.5*0.25}{0.5+0.25} = 0.333$$

Example: Same, but what if the positive class is Not Sarcastic?

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

Predicted	TP: ?	FP: ?
	FN: ?	TN: ?

Positive Class: Not Sarcastic

Precision = ?

Recall = ?

$$F_1 = \frac{(1^2+1)PR}{1^2P+R} = \frac{2PR}{P+R} = ?$$

Example: Same, but what if the positive class is Not Sarcastic?

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

	TP: 2	FP: 3
	FN: 1	TN: 1

Positive Class: Not Sarcastic

Precision = ?

Recall = ?

$$F_1 = \frac{(1^2+1)PR}{1^2P+R} = \frac{2PR}{P+R} = ?$$

Example: Same, but what if the positive class is Not Sarcastic?

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted		
	TP: 2	FP: 3
	FN: 1	TN: 1

Positive Class: Not Sarcastic

Precision = 0.4

Recall = ?

$F_1 = \frac{(1^2+1)PR}{1^2P+R} = \frac{2PR}{P+R}$ = ?

Example: Same, but what if the positive class is Not Sarcastic?

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

	TP: 2	FP: 3
	FN: 1	TN: 1

Positive Class: Not Sarcastic

Precision = 0.4

Recall = 0.667

$F_1 = \frac{(1^2+1)PR}{1^2P+R} = \frac{2PR}{P+R}$ = ?

Example: Same, but what if the positive class is Not Sarcastic?

Instance	Actual Label	Predicted Label
I was absolutely thrilled that my smoke alarm broke.	Sarcastic	Not Sarcastic
I was absolutely thrilled that my paper was accepted!	Not Sarcastic	Not Sarcastic
I am soooo sad that tomorrow's 8 a.m. meeting is cancelled.	Sarcastic	Sarcastic
Oh yay more things to grade!!!	Sarcastic	Not Sarcastic
Oh yay my new subscription box arrived!!!	Not Sarcastic	Sarcastic
Where is the closest coffee shop?	Not Sarcastic	Not Sarcastic
I just love large group meetings.	Sarcastic	Not Sarcastic

Actual

Predicted

	TP: 2	FP: 3
	FN: 1	TN: 1

Positive Class: Not Sarcastic

Precision = 0.4

Recall = 0.667

$$F_1 = \frac{(1^2+1)PR}{1^2P+R} = \frac{2PR}{P+R} = \frac{2*0.4*0.667}{0.4+0.667} = 0.50009$$

Summary: Naïve Bayes Essentials

- **Naïve Bayes** is a **probabilistic, supervised classification algorithm**
- When making predictions, a classifier takes a test observation, extracts a set of features from it, and assigns a label to the observation based on similarities between its feature values and those of observations in the training dataset
- **Multinomial Naïve Bayes** assumes that there is a discrete set of possible classes for the data
- Naïve Bayes is "naïve" because it makes the simplifying assumption that **all features are independent of one another**
- Naïve Bayes classifiers generally use **bag of words** features, but may use other features (e.g., those from external **lexicons**) depending on the task
- Classification model performance is determined by comparing the model's predictions to a set of **gold standard labels**
- The similarities and differences between predicted and actual labels can be summarized in a **contingency table** containing **true positives**, **false positives**, **true negatives**, and **false negatives**
- Four common metrics can be computed from values in this table
  - **Precision:** Of the observations predicted to be true, how many actually are?
  - **Recall:** Of the observations that are true, how many were predicted to be?
  - **F-Measure:** What is the harmonic mean between precision and recall?
  - **Accuracy:** What percentage of observations did the model label correctly?