Training a Naïve Bayes Classifier using BOW Features

Natalie Parde

UIC CS 421

How do we train a Naïve Bayes classifier?

- More specifically, how do we learn P(*c*) and P($f_i|c$)?
- 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!**

Remember, in our scenario we're assuming that a feature is just a word in a document's bag of words.

- Thus, to compute P(f_i|c), we'll just need P(w_i|c)
  - Fraction of times w_i appears among all words in all documents of class c
- How do we do this?
  - Concatenate all instances from class c into a big super-document of text
  - Find the frequency of w_i in this super-document to find the maximum likelihood estimate of the probability:
    - $P(w_i|c)' = \frac{count(w_i,c)}{\sum_{w \in V} count(w,c)}$
    - Note that V is the set of all word types across all classes, not just the words in class c

Recall, zero probabilities can be very problematic.

- Naïve Bayes naïvely multiplies all the feature likelihoods together
- This means that if there is a single zero probability when computing the word likelihoods, the entire probability for the class will be 0
  - $c' = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in T} P(w_i|c)$

How do we fix this issue?

- Smoothing!
- Simplest solution: Laplace (add-one) smoothing
  - $P(w_i|c)' = \frac{count(w_i,c)+1}{\sum_{w \in V}(count(w,c)+1)} = \frac{\boldsymbol{count(w_i,c)+1}}{\boldsymbol{(\sum_{w \in V}(count(w,c))+|V|)}}$

What about unknown words?

- Some words will inevitably occur in the test data despite never having occurred in the training data
- Easy solution for Naïve Bayes?
  - Ignore words that didn’t exist in the training data (remove from test document + do not compute any probabilities for them)

What about stop words?

- **Stop words** are very frequent words like *a* and *the*
- In some scenarios, it may make sense to ignore those words
  - Stop words may occur with equal frequency in all classes
  - However, this isn't always the case (e.g., spam detection)
- Stop words can be defined either automatically or using a predefined stop word list
  - Automatically:
    - Sort the vocabulary by frequency in the training set
    - Define the top 10-100 vocabulary entries as stop words
  - Predefined List:
    - Search online, or see if the package you're using (e.g., NLTK) already has one

Final, Formal Algorithm

Train Naïve Bayes

```
for each class c∈C: # Calculate P(c)
    N_doc ← |D|
    N_c ← number of d∈D from class c
    logprior[c] ← log(N_c/N_doc)
    V ← vocabulary of D
    superdoc[c] ← d∈D from class c
    for each word w in V:
        count(w,c) ← superdoc[c].count(w)
        loglikelihood[w,c] ← log(count(w_i,c)+1 / (Σ_{w∈V}(count(w,c))+|V|))
return logprior,loglikelihood,V
```

$$\text{loglikelihood}[w,c] \leftarrow \log\left(\frac{count(w_i,c)+1}{(\sum_{w\in V}(count(w,c))+|V|)}\right)$$

Test Naïve Bayes

```
for each class c∈C:
    sum[c] ← logprior[c]
    for each position i in testdoc:
        word ← testdoc[i]
        if word∈V:
            sum[c] ← sum[c]+loglikelihood[word,c]
return argmax_c sum[c]
```

$$\text{return } \underset{c}{\operatorname{argmax}}\, sum[c]$$

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

He congratulated Natalie for getting #2 on the bestseller list. Not Sarcastic

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 test instance, 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 test instance, 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.	?

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

- Taking a closer look at our test instance, 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.	?

- 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) = $\frac{1+1}{12+21} = 0.061$

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.	?

- 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) = $\frac{1+1}{12+21} = 0.061$

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

Natalie told Usman she was soooo totally happy for him.

Example: Naïve Bayes

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

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.	?

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