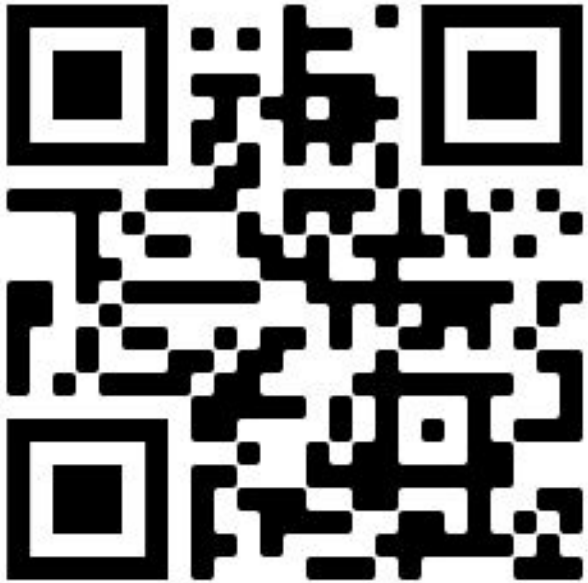
SIGNLL Meeting 3

Join Discord!



Agenda (9/26)

- Join the Discord (hopefully it works this time)
- Naive Bayes!
- Start thinking of project ideas

Naive Bayes!

- A probabilistic classifier!
- Out of all classes $c \in C$, it returns the class c-hat which is most likely.
- Uses:
  - Sentiment analysis!
  - Spam detection!
  - More generally, **text categorization**:
    - Tasks in which we label or categorize entire documents
- Let's look at Naive Bayes in the context of sentiment analysis for the next few slides: our classes will be either positive or negative.

Bag of Words

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

What makes Naive Bayes “naive” is that we don’t care about word order. We can represent a document as an unordered **bag of words**, as shown above.

What is Naive Bayes?

Bayes' rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

A bunch of stats and algebra later...

P(c) is the prior probability of a class among all documents.
P(w_i | c) is the probability of a particular word given that a document is of a certain class (formula for this on the next slide).

$$c_{NB} = \arg\max_{c \in C} \log P(c) + \sum_{i \in positions} \log P(w_i|c)$$

Laplace Smoothing

The added +1 and +|V| in the numerator and denominator are a result of Laplace smoothing. What if the word “great” appears in a negative document but not a positive one, in our data? Then P(great|positive) = 0, so the entire probability would be 0. Not ideal.

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

Try it out!

- Workshop: Use Naive Bayes and NLTK to classify tweets as either coming from Joe Biden or Kanye West