Assignment 4

Kun Qian

UCDAVIS
UNIVERSITY OF CALIFORNIA

Assignment 4

- Instructions will be on canvas
- Download the code and data from canvas
- Due Jun 5 at 11:59pm

Sentiment Analysis

- Using Naive Bayes we will be classifying movie reviews
- Homework will compare implementations of Naive Bayes with 3 modifications
  - Using StopWords, Binarized, and Custom

Naive Bayes

- Naive Bayes depends on the underlying assumption that there is independence among predictors
- "An apple is considered to be an apple if it is red, round, and 4 inches in diameter"

Pros and Cons Naive Bayes

- **Pros**
  - Easy and fast to predict classes of test data
  - If independence between features is true, than Naive Bayes classifier performs better than most models ie logistic regression with less training data
    - Rather than tuning parameters on each learned step, Naive Bayes just calculates them

Pros and Cons Naive Bayes

- **Cons**
  - Cannot categorized for a category not observed well in training.
    - Zero Frequency problem: smoothing helps
  - Bad estimator
  - Assumption of independence is rare in most data sets

Intro: Scripts and Data

- NaiveBayes.py: The starter code file for you to implement three types of naive bayes classifier for movie review sentiment analysis.
- data/imdb: 1000 positive reviews and 1000 negative reviews.
- data/english.stop: a set of English stop words.

Task 1

- Implement a basic naive bayes classifier to predict the sentiment for movie review.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(w_i \mid c_j)$$

- What is the probability of a particular sentiment given a feature?
  - How does this feature relate to other features (weighting features)

Task 1

- Implement a basic naive bayes classifier to predict the sentiment for movie review.

Let N_c be number of documents with class c

Let N_{doc} be total number of documents

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

Task 1

- Implement a basic naive bayes classifier to predict the sentiment for movie review.

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

Task 1

- Run:

Python NaiveBayes.py data/imdb

- What is the prediction accuracy you get?
  - Is the distribution of positive to negative reviews balanced for training?

Adding Reviews to Model

```
def addDocument(self, classifier, words):
    """
    Train your model on a document with label classifier (pos or neg) and words (list of strings). You should
    store any structures for your classifier in the naive bayes class. This function will return nothing
    """
    # TODO
    # Train model on document with label classifiers and words
    # Write code here

    pass
```

- What information is worth retaining?
  - # of classifications per across corpus?
  - Word frequency?
  - Word frequency per sentiment?

- Maintain these structures within your Naive Bayes class

Adding The Classifier

```
def classify(self, words):
    """
    Classify a list of words and return a positive or negative sentiment
    """
    if self.stopWordsFilter:
        words = self.filterStopWords(words)

    # TODO
    # classify a list of words and return the 'pos' or 'neg' classification
    # Write code here

    return 'pos'
```

- You will need to calculate the following:
  - Probability of positive and negative reviews in the corpus
    P(Pos) = #Pos/(#Pos + # Neg)
    P(Neg) = #Neg/(#Pos +# Neg) or 1-P(Pos)

  - More on the next slide...

Adding The Classifier

- You will need to calculate the following:
  - Score for positive and negative sentiment (hint: do one at a time)

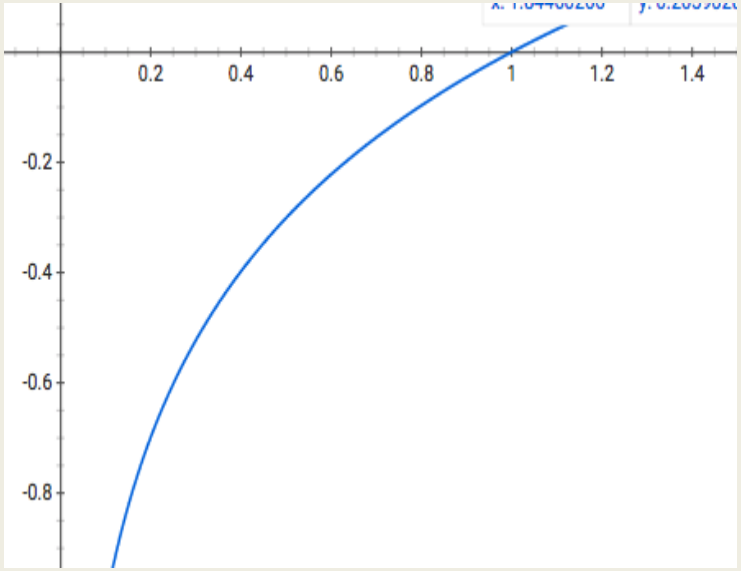
  - What we want to do (C_j is our positive/negative sentiment)
    -log(P(C_j)*P(w1|C_j)*P(w2|C_j)*...P(wN|C_j))

    This is equal to saying
    P(sentiment)*[P(w1, sentiment) * P(w2, sentiment) … P(wN, sentiment)]

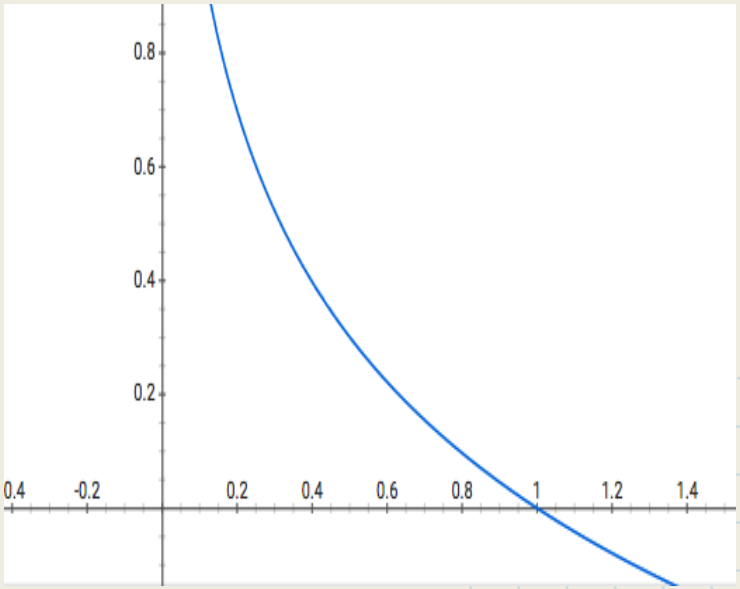
  - Using -log we can continuously decrement each probability (word probability for a sentiment) and finally multiple/decrement the probability of the sentiment itself.

Adding The Classifier

- Using -log we can continuously decrement each probability (word probability for a sentiment) and finally multiple/decrement the probability of the sentiment itself.
- Start at score of 0, decrement by -log(Prob(Words, Sentiment))



log(x)



-log(x)

Task 2

- Evaluate the model with the stop words removed.

- Run

Python NaiveBayes.py -f /data/imdb

- Does this approach affect average accuracy? Explain why removing stop words helped?

Task 3

- Implement a binarized version of the Naive Bayes Classifier.
- Clip all the word counts in each document at 1.

For each word w_k in *Vocabulary*

$n_k \leftarrow$ # of docs belong to c_j that contain w_k

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

Task 3

- Run

Python NaiveBayes.py -b data/imdb

- Did you get improvement, when compared to basic naive bayes classifier?

Task 4

- Design new features and heuristics to enhance your movie review sentiment predictor.

- Run

Python NaiveBayes.py -m data/imdb

Example of new heuristics:
Add "NOT_" to all the words after you detect a negation in the document.

Task 4

Constraints:

1. Only change features, but not the classifier method. You can only use naive bayes for this assignment.

1. You cannot use any external python packages other than the default ones that python have.

1. Limit your changes to addDocument() and classify() as much as you can. Changes beyond addDocument() and classify() needs to be done with cautions.

Task 4

Goals:

1. Minimum requirement: (1) Your best model should be achieve higher probability than your basic naive bayes classifier and the binary version naive bayes classifier. (2) It will need to achieve at least 83% average accuracy with the 10-fold cross validation on imdb dataset. (3) It will need to achieve a higher accuracy than a TA model on the hold out dataset.

1. Competitive Task: Improve your classifier as much as you can! We will test all your classifiers with a hold out dataset. The top 10% classifier with the highest prediction accuracy will be awarded 5 bonus points for this assignment.

Task 4

Some suggestions on how to improve:

1. Properly handle negation and turns in the sentence:

I had heard this movie was very good, but I found it bad.

The movie is neigher inspired nor realistic.

2. Appropriate feature selection. Removing stop words is a simple version of this process. Consider what words contribute to a certain sentiment more.