CSE 250B: Machine learning **Fall 2016**

Homework 2 — Sparse generative models

Overview

The multinomial naive Bayes model is a quick-and-dirty way to do text classification. In some situations, it would be helpful to have a *sparse* version of this model – that is, one that focuses on a small subset of the full vocabulary. This would offer two clear advantages:

- The class-conditional probability of a document would be the product of the probabilities of important words, rather than the product of probabilities of mostly-irrelevant words.
- Sparser models are more easily interpretable.

It is in any case common practice to discard a few dozen (or few hundred) "stopwords": words like `a`, `the`, `and`, and so on. Here we have a far more drastic reduction in mind.

In this assignment, you will develop your own ideas for for selecting a small subset of the vocabulary for multinomial naive Bayes. The final goal is good classification accuracy.

You will then implement your method and test it on the 20 Newsgroups dataset.

Some further details

- There are several versions of 20 Newsgroups on the web. You should download "20news-bydate.tar.gz" from

  `http://qwone.com/~jason/20Newsgroups/`

  Unpack it and look through the directories at some of the files. Overall, there are roughly 19,000 documents, each from one of 20 newsgroups. The label of a document is the identity of its newsgroup. The documents are divided into a training set and a test set.

- The same website has a processed version of the data, "20news-bydate-matlab.tgz", that is particularly convenient to use. Download this and also the file "vocabulary.txt". Look at the first training document in the processed set and the corresponding original text document to understand the relation between the two.

- The words in the documents constitute an overall vocabulary V of size 61188. Recall that to build a multinomial naive Bayes model (over the full vocabulary), you would begin by computing the following statistics for each of the 20 classes $j = 1, 2, \ldots, 20$:

  - π_j, the fraction of documents that belong to that class; and
  - P_j, a probability distribution over V that models the documents of that class. To estimate this, imagine that all the documents of class j are strung together. For each word $w \in V$, let P_{jw} be the fraction of this concatenated document occupied by w. Well, almost: you will need to do smoothing (just add one to the count of how often w occurs).

  You will instead be focusing upon a carefully-chosen subset of V, and you need to decide what to do with the remaining words: ignore them altogether, or treat them in some uniform way?

- When classifying a new document, remember to work with logs to avoid underflow.

What to turn in

On the due date, turn in a **typewritten** report containing the following elements (each labeled clearly).

1. *A short, high-level description of your idea for vocabulary selection.*

   A few sentences should suffice. These should be crystal clear: they should communicate the key idea to the reader.

2. *Concise and unambiguous pseudocode.*

   (Please do not submit any actual code.) Once again, clarity and conciseness are of the essence. Your scheme should take as input the full vocabulary, the labeled training set, and a number M, and it should return a subset of M vocabulary items along with a multinomial naive Bayes model that estimates parameters only for these features.

   Also indicate how classification will be done. What happens if a test document contains none of the selected words?

3. *Experimental results.*

   Try out your method for a few different values of M, including at least 5000, 10000, and 20000 (the total vocabulary size for this dataset is over 60000). Compare the classification accuracy with that of multinomial naive Bayes on (1) the full vocabulary and (2) a randomly selected subset of M vocabulary words. This comparison should be summarized in tables or (even better) graphs, all clearly marked. Remember to provide error bars for any method that is randomized.

   The pseudocode and experimental details must contain all information needed to reproduce the results.

4. *Inspection of models.*

   Choose a smaller value of M, say $M = 1000$, and out of the selected words, show 10-20 "representatives" from each class.

   (a) How did you choose these representatives?

   (b) Do the selected words make sense to you?

5. *Critical evaluation.*

   Is your method a clear benefit? Is there further scope for improvement? What would you like to try next?