AI201 Programming Assignment 2
Naive Bayes Spam Filter

Pros Naval

Deadline: 12:00 noon of October 12, 2024

1 Introduction

Spam detection in email may be considered as a two-category classification problem. The first category represents legitimate messages or ham and the other unsolicited messages or spam. In this assignment you will be given the opportunity to make your own Naive Bayes classifier for filtering spam. For a successful implementation you need to train your classifier on some messages labelled as ham or spam and evaluate its performance on unseen test messages.

Recall that the Naive Bayes formula is given by

$$P(\omega|x_1, x_2, ..., x_d) = \frac{\prod_{i=1}^{d} P(x_i|\omega)P(\omega)}{\sum_{\omega} \prod_{i=1}^{d} P(x_i|\omega)P(\omega)} \tag{1}$$

where d is the vocabulary size $|V|$ and ω is the spam or ham class (i.e., $\omega \in \{\text{ham, spam}\}$) ,. For our spam filter, the class conditional likelihood for word x_i is as follows:

$$P(x_i|\omega) \quad = \quad \frac{\sum_{D \in D_\omega} I(x_i \in D)}{|D_\omega|}$$

where $I(\cdot)$ is the indicator function, D_ω is the set of documents belonging to class ω in the training set, and $|D_\omega|$ is its cardinality.

Note that when a word does not occur in a document class in the training data, it does not mean that it will never appear in any document of that class. We need to modify the formula to avoid a zero denominator. We circumvent this problem by using Lambda Smoothing which uses a modified formula for the class conditional likelihood:

$$P(x_i|\omega) = \frac{\left[\sum_{D \in D_\omega} I(x_i \in D)\right] + \lambda}{|D_\omega| + \lambda|V|} \tag{2}$$

where $|V|$ is the vocabulary size. When $\lambda = 1$, this formula is called Laplace Smoothing.

In order to avoid loss of precision during multiplication of the likelihoods, I suggest that you instead add the logarithm of the likelihoods and exponentiate the result.

2 Naive Bayes for the TREC06 Corpus

You will design a classifier for a subset of the TREC06 Public Spam Corpus which is a dataset for benchmarking spam algorithms. The link to this dataset will be provided to you. After downloading and unzipping the dataset, you should see a subdirectory containing subdirectories of email messages and a label file named `labels`. The files within the subdirectories contain ham and spam messages which you will use for training and testing.

Note: Except for the matplotlib and csv libraries, you are not allowed to use any other library that implements the Bayesian Classifier. You must code the classifier on your own, otherwise the rest of your submission will not be checked and you will get zero credit.

2.1 Classifier Construction and Evaluation

1. Write a script that makes disjoint training and test sets containing ham and spam. Use a 70-30 partitioning of your data.

2. Parse the documents in the training set. For simplicity, define a word as any sequence of alphabetic characters [a-zA-Z] delimited by a white space in front and a white space, comma, or period at the end. Form the vocabulary V of unique words in the training data, count their statistics and report the prior probabilities for spam and ham.

3. Construct and train a Naive Bayesian Classifier from the count statistics above. You are not allowed to use any library (such as scikit-learn) that has a Naive Bayes implementation nor copy code from anywhere.

4. Implement the code for classifying an unknown message and try it on the testset.

5. Write a function that computes the precision and recall measures.

   The formulas for precision and recall are as follows:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

   where

   TP (true pos) : num of spam messages classified as spam

TN (true neg) : num of ham messages classified as ham
FP (false pos) : num of ham messages misclassified as spam
FN (false neg) : num of spam messages misclassified as ham

2.2 Lambda Smoothing

1. Write another function that uses λ smoothing.

2. Use λ smoothing for 5 different values of λ (e.g. $\lambda = 2.0, 1.0, 0.5, 0.1, 0.005$) and print the precision and recall for these values. What value of λ yields the best precision and recall ?

2.3 Improving your Classifier

Your Naive Bayes Classifier treats each word in exactly the same way. Some words however such as "a, of, the" are less informative than others. These words tend to occur more frequently in both classes than words that tend to suggest that the message is spam.

1. Find a way to identify the most informative words for spam filtering. Using the best value of λ found above, print the top 200 informative words for spam and ham messages. (Hint: Read Hovold's paper [1])

2. Evaluate the precision and recall of your classifier using this smaller vocabulary of 200 words.

2.4 Writing your Report

Prepare a 2,000 word (minimum) report consisting of the following:

1. Description of the Naive Bayes Algorithm you used in your implementation

2. Description of your implementation including the modification suggested by Hovold

3. Analysis and discussion of results

4. Conclusions and suggestions for future work on how to improve your classifier

3 Deliverables

Deliverables for this Machine Problem:

1. 2,000 word (minimum) conference-style report using prescribed Latex template

2. Jupyter notebook containing Python code, figures, plots, etc.

3. training and test sets

4. All other files needed to reproduce your experimental results

The deadline for submission is **12:00 noon of October 12, 2024**. Email soft copy of report, source codes and files to `submit2pcnaval@gmail.com` with "[AI201: PA2 Submission] Your Name " on the subject line. Do not email links from DropBox, Google Drive, etc.

4 References

[1] Johan Hovold, Naive Bayes Spam Filtering using Word Position Attributes, In *Conference on Email and Anti-Spam*, Stanford University, 2005.