Text Classification using Naïve Bayes Algorithm

August 23, 2018

Instructions

- This document is **confidential** and should not be shared publicly.
- The primary aim of this exercise is to **understand your thoughts**. Code optimization and your skills at programming will be secondary.
- In case you are stuck, write a **pseudocode** describing your approach.
- Any text in **blue** indicates a url and is clickable.
- You are free to use *any* programming language to solve the problem.
- I encourage you to use **only the basic libraries** for the solutions. For example, in case of python I suggest using just the libraries like numpy, scipy, matplotlib, etc. and avoid scikit-learn, pandas, etc.
- In case of any questions please feel free to reach out to me on yash@helpshift.com, cc:dinesh.kandhari@pubmatic.com

helpshift

Data Science

Bayesian Spam Filter

The aim of this problem is get an idea about how to use Naïve Bayes Algorithm to build a text classifier from **scratch**. I recommend **not** to use any advanced libraries for this assignment.

- Read the essay `A plan for spam` by Paul Graham. It will help you appreciate the intent of this assignment.
- Download and un-compress the 6 `Enron-spam in pre-processed form` folders from `The Enron Dataset`.
- Separate out the dataset based on the tags `spam` and `ham`.
- Tokenize the data using some very basic assumptions. You can treat `whitespace/newline` as word delimiters.
- Remove punctuation characters like `.,;:!?&\/$%‘‘’’()` etc. You may choose to keep other characters such as `_-’` etc.
- Now we will reduce our vocabulary(or the dimensionality of our problem). Remove stopwords from the texts. Use `this` very basic stopwords list for your reference.
- Get token-counts for the two tags[1]. Use appropriate data-structure to store this information.
- Get Term Frequency(tf)[2] and Document Frequency(df)[3] for all tokens. These are basic statistics that will help us identify important words. Remove the tokens which have tf or df less than a certain threshold. Choice of tf or df depends on you.
- *(Optional)* Use some ideas either from `A plan for spam` or your own to reduce the vocabulary further.
- Build a Naïve Bayes classifier. Use the first 5 folders as training dataset, and the 6^{th} folder (`Enron6`) as the testing dataset.
- Report `Accuracy, Precision, Recall` for the classifier.
- Write a note describing your work, and summarizing the performance of your classifier.

---

[1] Number of times a token appears in the corresponding corpus for a given tag
[2] Total number of times a token appears in the entire corpus
[3] Number of documents in the entire corpus that contain the token