9. Feature Engineering

Extension Note: Project 1 due date has been extended by 2 days to **July 4 23:59UTC** (Note the UTC time zone).

Frequently, the way the data is represented can have a significant impact on the performance of a machine learning method. Try to improve the performance of your best classifier by using different features. In this problem, we will practice two simple variants of the bag of words (BoW) representation.

Remove Stop Words

1/1 point (graded)
Try to implement stop words removal in your feature engineering code. Specifically, load the file **stopwords.txt**, remove the words in the file from your dictionary, and use features constructed from the new dictionary to train your model and make predictions.

Compare your result in the **testing** data on Pegasos algorithm using $T = 25$ and $L = 0.01$ when you remove the words in **stopwords.txt** from your dictionary.

Hint: Instead of replacing the feature matrix with zero columns on stop words, you can modify the `bag_of_words` function to prevent adding stopwords to the dictionary

Accuracy on the test set using the original dictionary: 0.8020

Accuracy on the test set using the dictionary with stop words removed:

0.808 ✔ **Answer:** 0.8080

Solution:

- The original dictionary size is 13401, while the dictionary size after stop word removal is 13276.

Submit You have used 1 of 20 attempts

ℹ Answers are displayed within the problem

Change Binary Features to Counts Features

1/1 point (graded)
Again, use the same learning algorithm and the same feature as the last problem. However, when you compute the feature vector of a word, use its count in each document rather than a binary indicator.

Hint: You are free to modify the `extract_bow_feature_vectors` function to compute counts features.

Accuracy on the test set using the dictionary with stop words removed and counts features:

0.77 ✔ **Answer:** 0.7700

Try to compare your result to the last problem, and see the discussion in solution after answering the question.

Solution:

- The performance is 0.7700, which is worse than the previous problem.
- Even if you use the original feature sets (without stop word removal), it will still decrease performance.

- It is possible that giving models more information leads to worse performance, when model overfits irrelevant information.
- From the learning perspective, it is helpful to somehow “regularize" feature representations (e.g., binarizing them).

Submit You have used 1 of 20 attempts

Answers are displayed within the problem

Some additional features that you might want to explore are:

- Length of the text
- Occurrence of all-cap words (e.g. “AMAZING", “DON'T BUY THIS")
- Word embeddings

Besides adding new features, you can also change the original unigram feature set. For example,

- Threshold the number of times a word should appear in the dataset before adding them to the dictionary. For example, words that occur less than three times across the train dataset could be considered irrelevant and thus can be removed. This lets you reduce the number of columns that are prone to overfitting.

There are also many other things you could change when training your model. Try anything that can help you understand the sentiment of a review. It's worth looking through the dataset and coming up with some features that may help your model. Remember that not all features will actually help so you should experiment with some simpler ones before trying anything too complicated.

Discussion

Show Discussion

Topic: Unit 1 Linear Classifiers and Generalizations (2 weeks):Project 1: Automatic Review Analyzer / 9. Feature Engineering