Generative Model: Naive Bayes

This directory contains the code used to generate our generative model for the competition. We decided to use a simple Naive Bayes classifier to represent our generative model. Please follow the instructions below to run the code.

About

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with naive, or strong, independence assumptions between the features. In our case, we use a binary multinomial Naive Bayes classifier to classify tweets. With the *bag of words assumptions*, we assume that position between the words in our tweets does not matter. Furthermore, instead of training on the frequency of a word occurring in a tweet, we train on whether a word occurs or not. Thus, we clip the word counts to 1 in order to improve performance.

Instructions to run code

The instructions below will guide you in training and testing the model.

Training the model

To train the model, simply run the following command in MATLAB's command window:

```
naive_bayes
```

This will run the training script, `naive_bayes.m` for the model. The training script takes care of loading and preprocessing all of the training data. At the end, it will output the amount of time taken to train the model, general loss (error) of the trained model, and the cost associated with testing the model on the training data.

Note: If you would like to save the trained model, uncomment line 23 in `naive_bayes.m`. It will store the trained model as `naive_bayes.mat`. Keep in mind that this will overwrite the model already provided in this directory.

Testing the model

To test the model, simply call the `test_nb_model` function. This function takes in 3 parameters:

1. The test bag of words (sparse matrix, `X_test_bag`)
2. The raw set of test tweets (cell matrix, `test_raw`)
3. The true test labels (`Y_test`)

`test_nb_model` will return the cost of predicting test data using our trained model. The function

should be called in MATLAB's command window as the following:

```
test_nb_model(X_test_bag, test_raw, Y_test)
```

Note: Make sure your `X_test_bag` is a sparse matrix of size Nx10000, where N denotes the number of samples in the data set. Also make sure `test_raw` is cell matrix of size Nx1, and `Y_test` is double matrix of Nx1 as well.

An example

Say we want to test our trained model on the training data. Make the following call in MATLAB's command window:

```
test_nb_model(X_train_bag, train_raw, Y_train)
```

This assumes that you have already loaded the training data into the workspace. If not, simply enter the following into MATLAB's command window:

```
load('../data/train.mat')
```