Tyler Altenhofen, Rohan Alur

Code Intro

There is a file called **run_all.m** that contains a script for training and running each of our models on the supplied validation set. Each models training and prediction implementation is in if a file with p̈redictsäs a prefix (eg. **predict_knn.m**). We then find the parameters using cross validation in scripts with a postfix of ẍval(̈eg **knn_xval.m**).

Generative Methods

Naive Bayes

ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!

To run the algorithm on the validation set run

```
load('train.mat');
load('validation.mat');
load('vocabulary.mat');
Y_hat = predict_naivebayes(X_train_bag, Y_train, X_validation_bag)
```

K-Means Clustering

In this model we first run a basic k-means clustering algorithm. We then determine the appropriate class for each of the clusters by using the class with the maximum number of occurrences with in the cluster. We use this to predict values by finding which cluster it fits into and guessing the class appropriately.

To train and then run our implementation of k-means on the supplied validation set you can run

```
load('train.mat');
load('validation.mat');
load('vocabulary.mat');
Y_hat = predict_k_means(X_train_bag, Y_train, X_validation_bag)
```

Discriminative Methods

Logistic Regression

We used liblinear to run logistic regression on the data using an L2 loss. We then multiply the cost matrix by the predicted probabilities and predict the class with the smallest expected cost.

ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!

To run the algorithm on the validation set run

```
load('train.mat');
load('validation.mat');
load('vocabulary.mat');
Y_hat = predict_logistic(X_train_bag, Y_train, X_validation_bag)
```

Random Forest

ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!
ROHAN FILL THIS IN !!!!!!!!!!!

To run the algorithm on the validation set run

```
load('train.mat');
load('validation.mat');
load('vocabulary.mat');
Y_hat = predict_rf(X_train_bag, Y_train, X_validation_bag)
```

Instances Based Methods

K-Nearest Neighbors

This algorithm runs k-nearest neighbors and finds the 100 closest points. It then uses the ratio of points in each class to predict a probability for each class. Finally, it uses the cost matrix to predict the class with the lowest expected cost.

To run the algorithm on the validation set run

```
load('train.mat');
load('validation.mat');
load('vocabulary.mat');
Y_hat = predict_knn(X_train_bag, Y_train, X_validation_bag)
```