CSC411 Fall 2017
Assignment 3 Report

Tianbao Li

2017/12/04

1 Q1: 20 Newsgroups prediction

1.1 Data summary

Here is the detailed information of the dataset:

- data set: The 20 newsgroups text
- data amount: 11314
- feature amount: 101631
- data representation: tf-idf
- baseline: Bernoulli Naive-Bayes classifier

1.2 Model training

To train model to outperform the baseline, here we useopen-souece code of scikit-learn. Here, I picked several different algorithms and trained their hyper-parameters with k-fold validation. The detailed better hyper-parameters, train losses (0-1 loss), test losses (0-1 loss) are shown in Table 1. From the result of the validation losses, the best three models are Neural Networks, Multinomial Naive Bayes, Logist Regression, and has the same best results in test dataset.

Model	Hyper-pramater	Train loss	test loss
BernoulliNB		0.598727240587	0.457912904939
MultinomialNB	$\alpha = 0.01$	0.958900477285	0.700212426978
Logist Regression	$C = 500$	0.974721583878	0.683483802443
SGD	$\alpha = 0.0001$	0.962877850451	0.671136484334
SVM	$C = 1$	0.895704436981	0.67750929368
KNN	$K = 1$	0.973749337104	0.113382899628
Decision Tree	$K = 601$	0.974721583878	0.4026818906
Neural Networks	$\alpha =$		

Table 1: Model comparison

1.3 Hyper-parameters training

To train models by choosing better-fitting parameters, I splited training data by KFold and run cross valiadation by cross_val_score.

Here, taking multinomial Naive Bayes as an example. Hyper-parameter α is used for smoothing. I picked several possible α value and compared by validation loss, then chose a better α.

```
splits = 5
kf = KFold(splits, shuffle = True, random_state = 0)
As = [1e-10, 1e-8, 1e-6, 1e-4, 1e-2, 0.1, 0.5, 1.0, 2, 5, 10]
scores = []
for a in As:
    model = MultinomialNB(alpha = a)
    score = cross_val_score(model, tfidf_train, train_labels, cv = kf)
    scores.append(np.mean(score))
opt_A_index = int(np.argmax(scores))
opt_A = As[opt_A_index]
```

1.4 Model selection

For the three well-working models, they have their advantages for solving this problem.

- **Neural Networks** can work well for complex models, especially when it comes deeper.
- **Multinomial Naive Bayes** implements the naive Bayes algorithm for multinomially distributed data, text classification usually holds such structure.
- **Logistic Regression** is famous for linear classification, which the newsgroup data has similar distribution.

1.5 Class confusion