Statistics 5525: Homework 5

Man Tang, Linjun Li, Pinyi Lu, Man Zhang, Yafei Zhang

Problem 1

Introduction

The naive Bayes classifier is very popular over the years. It is appropriate when the dimension of the feature space is very high. The naive Bayes model assumes that the features are independent for a given class. The dimension of features in our dataset is 57. Thus, we use naive Bayes classifier to classify the observations by tuning the kernel width.

Methods

Our dataset is from spam.data.txt. The dimension of features p is 57. The number of observations n is 4601. For the labels, spam is represented by 1 and email is represented by 0. Since the last two features have larger spreads than the others, we first standardize the features so that each feature has mean 0 and standard deviation 1. In the classification, we use select the kernel widths and report cross validated error rates. Then, we select a single kernel width for all features and classes and calculate the optimal cross-validated error rates (1 kernel width). Next, we tune the optimal kernel widths (both classes will use the same kernel width) and calculate cross-validated error rates (57 kernel widths). For each feature/class pair, we tune optimal kernel widths and also calculate the cross-validated errors (57 × 2 widths). At last, we repeat the above exercises and calculate the perceived error rates for comparison.

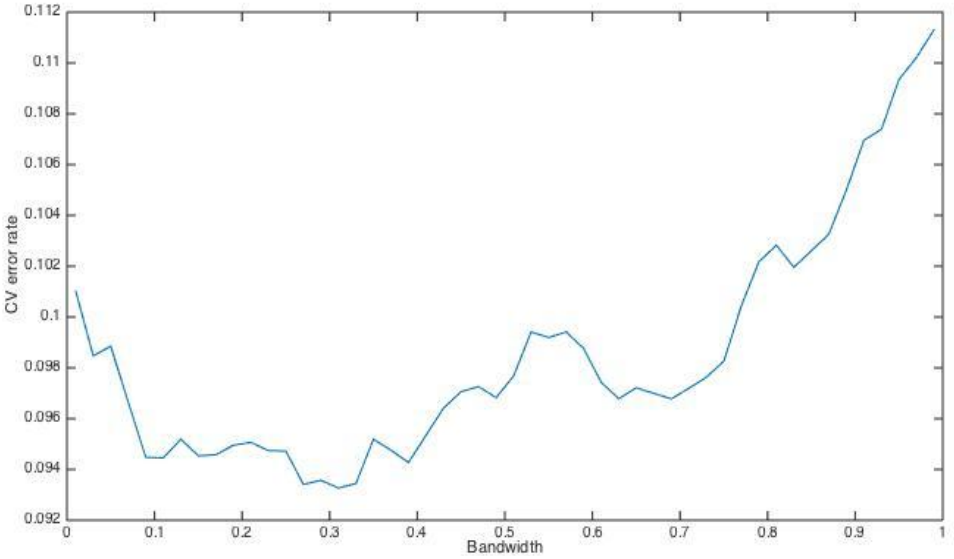
Results

Part a

For a given width 1, the cross-validated error rate is 0.11282.

Part b

We used 5-folds cross validation. When width=0.31, the cross-validated error rate is smallest, 0.0932.



Part c

Using 5-fold cross validation. The smallest cross-validated error rate is 0.0681 with width equals:

```
[1.5   0.3   1.8   0.2   0.4   1.7   0.4   0.4   0.1   0.8   0.1
 0.5   1.8   1.3   0.2   0.2   0.5   1.9   1.2   0.8   1.3   0.5
 0.2   1.9   0.1   0.4   0.1   0.7   0.1   0.6   1.3   1.9   0.8
 0.2   0.1   0.1   0.1   1.9   1.2   0.4   0.1   0.3   0.6   0.2
 0.5   0.1   0.2   0.3   0.4   0.8   1.2   0.1   0.2   0.3   0.2
 0.9   0.1]
```

Part d

Using 5-fold cross validation. The smallest cross-validated error rate is 0.0761 with width equals:

```
Columns 1 through 16
  1.0   0.3   0.5   0.2   0.4   0.3   0.3   0.4   0.2   0.2   0.4   0.4   0.3   0.3   0.3   0.3
  0.7   0.2   0.1   0.6   0.8   0.3   0.3   0.1   0.2   0.4   0.3   0.2   0.3   0.4   0.3   0.3
Columns 17 through 32
  0.2   0.5   0.8   0.3   0.3   0.2   0.2   0.6   0.4   0.3   0.3   0.2   0.4   0.2   0.5   0.3
  0.3   0.3   0.3   0.3   0.1   0.4   0.3   0.3   0.3   0.2   0.3   0.3   0.4   0.2   0.3   0.4
Columns 33 through 48
  0.3   0.2   0.3   0.4   0.3   0.3   0.3   0.3   0.3   0.4   0.2   0.4   0.6   0.3   0.2   0.2
  0.3   0.2   0.5   0.2   0.3   0.4   0.3   0.5   0.2   0.2   0.3   0.3   0.3   0.2   0.5   0.3
Columns 49 through 57
  0.3   0.4   0.3   0.1   0.3   0.3   0.3   0.4   0.2
  0.4   0.2   0.3   0.2   0.6   0.2   0.4   0.1   0.4
```

Part e

The perceived error rates are 0.08324 (1 width), 0.0572 (57 kernel widths), 0.0602 (57 × 2 widths).

Conclusion

Comparing these three perceived error rates in part e, the model with 57 kernel widths is the smallest, indicating that this model performs best in prediction.

Problem 2

Introduction

In this problem, we used three classification methods to fit three different classification models, using the training data "X_train.txt" (with 561 variables and 7352 observations) and the labels in "y_train.txt". The three classification methods include: Support Vector Machine, Classification Tree and Random Forest. With the the three fitted models we obtained three versions of labels for the records in the testing dataset "X_test.txt".

Methods

For each of the three classification model, we run five-fold cross-validation to find the the optimal model parameters. For the SVM, we search for the optimal "cost" which

penalize the misclassification of the training data; for the Classification Tree, we search for the optimal number of terminal nodes "T"; whereas for Random Forest, we search for the optimal number of variables randomly sampled as candidates at each split, mtry. With the optimal parameters we can calculate the averaged error (misclassification) rate and make comparison of the performance of the three classification algorithms.

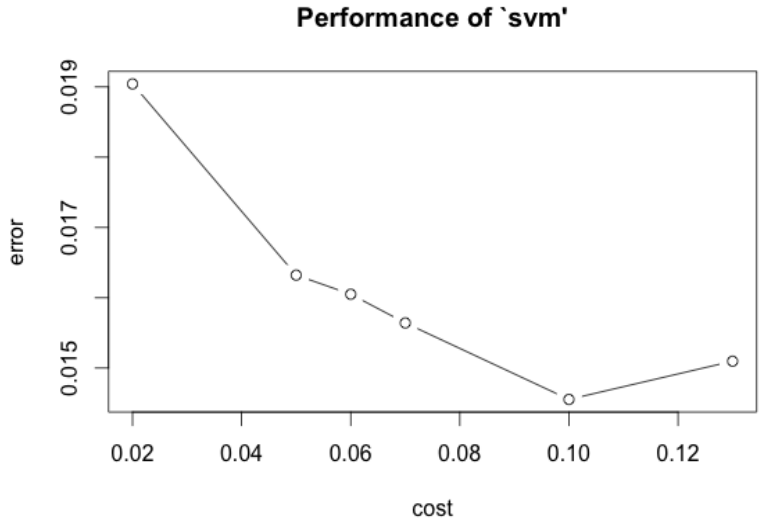
Results

Part a

I, Support Vector Machine (SVM) with Linear Kernel.

In this section, we show our classification results using SVM with a linear kernel. Five-fold cross validations are performed for different values of the cost parameter "C" (the constant of the regularization term in the Lagrange formulation, which determines how much we penalize the misclassification of the data in the training set). After trying a series of C values (i.e. C=0.02, 0.05, 0.06, 0.07, 0.1, 0.13), we obtain the following five-fold averaged error rate.

Cost (C)	0.02	0.05	0.06	0.07	0.10	0.13
Error Rate	0.0190	0.0163	0.0161	0.0156	0.0146	0.0151



The best performance happens at C=0.1, where the error rate is the lowest (0.01455). With C=0.1, the SVM model 'svmfit' is obtained with the training data. In this model, 903 records are used as support vectors and an averaged total accuracy is 98.49201, the error rates in the five-fold cross validations calculated again, which are: 0.0136054, 0.0142857, 0.0142760, 0.0149660 and 0.0183549. Also, since the decision hyperplane of the linear SVM is formed by the support 903 vectors, we only need to study the variations of the covariates with these support vectors in order to find the useful features (covariates). This can be done by calculating the variances of the covariates and select the covariates with large variances, such as $var(x_i)>1$. The covariates selected as useful features with different thresholds of variance can be found as follows (the followings are

the numberings of the covariates):

```
> which(vars>1)
[1]   1   2   3  18  23  25  28  31  32  33  36  37  39  40  44  45  46  47  48
[20]  49  56  60  61  62  63  64  65  66  67  68  69  78  79  80  83 107 108 110
[39]  111 112 113 114 115 116 117 119 120 121 122 123 144 148 149 152 158 159 160
[58]  161 170 177 188 189 191 195 197 198 199 200 210 211 212 213 223 224 225 226
[77]  236 237 238 251 262 263 264 265 280 291 294 296 301 302 322 324 327 328 336
[96]  337 338 342 358 373 374 375 379 389 401 406 417 436 438 454 462 463 464 465
[115]466 467 468 470 471 472 474 507 513 514 515 526 551 552
> which(vars>1.25)
[1]   2   3  46  49  60  62 389 468 472 552
> which(vars>1.5)
[1]   3  46  49  62 389 468
> which(vars>1.75)
[1]  49  62 389
```

Comment: The reason why SVM can achieve so small error rate is that, in the training data with 7352 observations and 561 variables, the feature space is actually very sparse, which makes the SVM with linear Kernel methods a viable classification method.

II, Classification Tree.

5-fold cross-validation was conducted by tuning the number of terminal node (T) for classification tree. After trying a series of terminal node value, the minimal cross-validation error were found to be 0.0812.

Node	112	114	116	118	120
Error rate	0.120	0.112	0.0921	0.081	0.102

And the covariates considered as important are:

```
[1]  390  53  560  50  15  130  58  43  12  160  180  165  374  410  185  181  59
[18]  51  187  56  372  140  4  5  41  452 198  193  13  162  44  192  199  133
[35]  150  65  349  158  236  31  2  87  14  398  505  75  449  276  38  42  34
[52]  40  118  66  451  435  504  72  16  90  203  297  210  19  1  275  355  10
[69]  527   70   23   24   92   509   54   3   46   296   68   380   233
```

III, Random Forest.

In this section, we show our classification results using Random Forest. Five-fold cross validations are performed for different number of variables randomly sampled as candidates at each split (mtry). After trying a series of mtry values (i.e. mtry = 561, 280, 140, 70, 35, 18, 9, 4, 2, 1), we obtain the following five-fold averaged error rate. When mtry equals to 280, the lowest error rate was obtained.

mtry	561	280	140	70	35	18	9	4	2	1
error	0.0189	0.0172	0.0207	0.0350	0.0586	0.0858	0.168	0.306	0.471	0.637

Using the extractor function, "importance", for variable importance measures as produced by randomForest, the most useful features for making the predictions are identified including 53, 560, 41, 42, 272, 266, 509, 70, 57, and 599.

Discussion

The following is a comparison of the averaged 5-fold cross validation error rate of the optimal model from the three classification methods.

Classification Method	Linear SVM	Classification Tree	Random Forest
Error Rate	0.0146	0.081	0.0172

From the comparison of the error rates produced by different classification methods, we find SVM with the Linear Kernel achieves the smallest error rate. we suggest to use this method to predict the labels for the records in the testing dataset "X_test.txt".

Part b

I, Support Vector Machine (SVM) with Linear Kernel.

Using the fitted model 'svmfit', one can easily do the predictions for the testing data. The predictions for 2974 instances are saved in the file 'svm.txt'. The predicted number of instances that falls into each category (label=1,2,3,4,5,6) are: 517, 469, 402, 454, 568 and 537 respectively.

II. Classification Tree.

Using the fitted model in part a, the predictions on the testing data for each instances are: 550 for 1, 422 for 2, 415 for 3, 468 for 4, 555 for 5, and 537 for 6. The predicted labels using Classification Tree is saved in the file 'fitree.txt'.

III, Random Forest.

Using the fitted model in part a, the predictions on the testing data for each instances are: 537 for 1, 447 for 2, 403 for 3, 452 for 4, 571 for 5, and 537 for 6. The predicted labels using Random Forest is saved in the file 'RandomForest.txt'.