CS 534 Machine Learning homework 4

Si Chen

Problem 1

(a) Proof: $k_\beta(x,z) = (1+\beta x\cdot z)^2-1$

$= (1+\beta(x_1 z_1+x_2z_2))^2-1$

$= 2\beta(x_1 z_1+x_2z_2) +\beta^2(x_1 z_1+x_2z_2)^2$

$= 2\beta x_1 z_1+2\beta x_2z_2 + \beta^2x_1^2 z_1^2+ \beta^2x_2^2 z_2^2+ \beta^2 x_1 z_1x_2 z_2$

$=(\sqrt{2\beta}x_1, \sqrt{2\beta}x_2, \beta x_1^2, \beta x_2^2, \beta x_1 x_2) \cdot (\sqrt{2\beta}z_1, \sqrt{2\beta}z_2, \beta z_1^2, \beta z_2^2, \beta z_1 z_2)$

$=\emptyset(x) \cdot \emptyset(z)$,

Where $\emptyset(x)= (\sqrt{2\beta}x_1, \sqrt{2\beta}x_2, \beta x_1^2, \beta x_2^2, \beta x_1 x_2)$

Thus, $k_\beta(x,z)$ is a kernel.

(b) Let $f(X) = \frac{1}{||X||_2}$, $f(Z) = \frac{1}{||Z||_2}$ f(X) and f(Z) are constant, transpose do not change.

$K_1(X, Z)=(\frac{X}{||X||_2})^T (\frac{Z}{||Z||_2})= f(X) \cdot f(Z) \cdot X^T \cdot Z= f(X) \cdot f(Z) \cdot k(X,Z)$

According to (i), $K_1(X, Z)$ is a kernel.

$K_2(X, Z)=1$, $\emptyset(x)=1$. So $K_2(X, Z)$ is a kernel.

$K_3(X, Z)=1+(\frac{X}{||X||_2})^T (\frac{Z}{||Z||_2})= K_2(X, Z) + K_3(X, Z)$

According to (ii), $K_3(X, Z)$ is a kernel.

$K_{new}(X, Z) = (1 + (\frac{X}{||X||_2})^T (\frac{Z}{||Z||_2}))^3= K_3(X, Z) \cdot K_3(X, Z) \cdot K_3(X, Z)$

According to (iii), $K_{new}(X, Z)$ is a kernel.

Problem 2

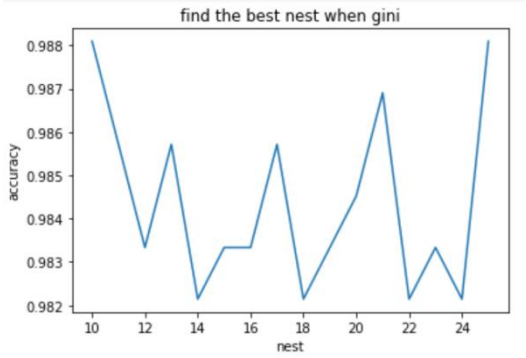
(a) Three functions :
get_subsample(Xarray,yarray)
random_forest(Xarray,yarray,Xnew,nest,criterion, maxDepth, minStamplesLeaf)
important_feature(Xarray,yarray,Xnew,nest,criterion, maxDepth, minStamplesLeaf)
Implement in the code hw4-2

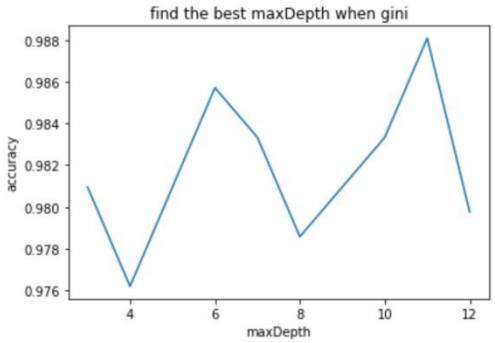
(b) For the nest=[10,25], the best parameters are: criterion=gini, nest=10, maxDepth=11, minStamplesLeaf=3.

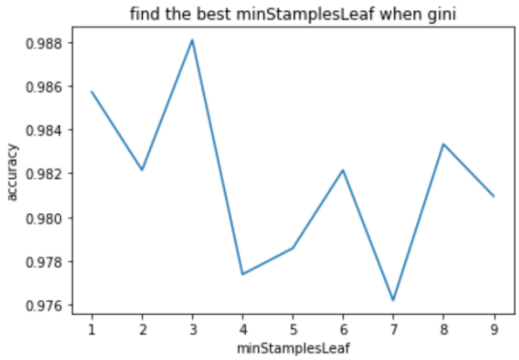
I find that the best parameters by three levels' for loop to both gini and entropy criterion.

As for the gini model, parameters are: nest=10, maxDepth=11, minStamplesLeaf=3
Its accuracy is 0.98809523809523814

The figure is as follow.

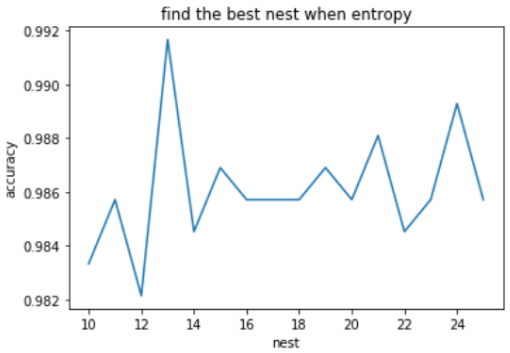


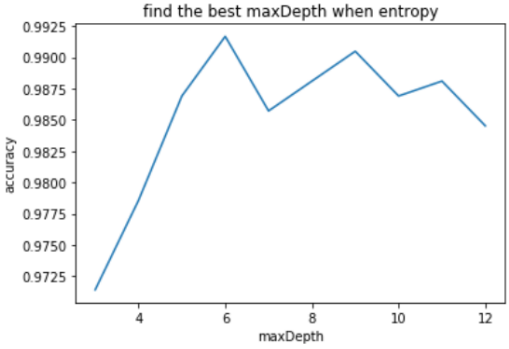


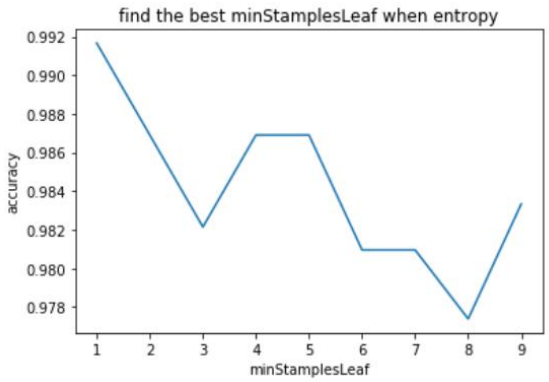


As for the entropy model, parameters are: nest=13, maxDepth=6, minStamplesLeaf=1
Its accuracy is 0.98095238095238091

The figure is as follow.







Therefore, gini performs a little better.

(c) Using the best parameters, when the random forest performs on the test data, the accuracy is 0.985714285714 The OOB sample accuracy is 0.985204081633
The two values are almost the same, which means OOB sample accuracy can validate the model while training.

(d) The most important feature is 'TSH'.
Other features sorted by importance are 'TSH', 'lithium', 'T3Ind', 'TT4Ind', 'T4UInd', 'onThyroxine', 'onAntithyroidMed', 'T3', 'I131', 'TSHInd', 'hypopituitary', 'goitre', 'TT4', 'FTIInd', 'queryHyperthyroid', 'queryOnThyroxine', 'sex', 'T4U', 'queryHypothyroid', 'thySurg', 'preg', 'FTI', 'tumor', 'sick', 'psych', 'age', 'refSource'

Problem 3

(a) At first I partition the data into 70%-30% train-test split. I use the training set to train the classification vector w by two algorithms, and use the test set to assess their performance. After I have chosen the better algorithm as well as the optimal parameter. I use this algorithm and exact parameter to train the whole data set by cross validation (5-fold).

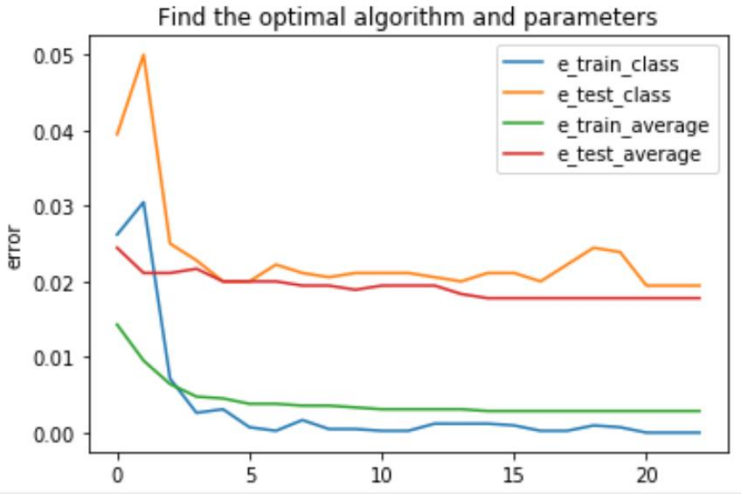
(b) First, I build a vocabulary list using the training set. To avoid overfitting, if the total number of one word is smaller than 30, I delete it. Then I also delete the keys of 1 and 0. At last I delete the words that appear fewer than 30 emails. Then I transform both train data and test data to a 1/0 table.

(c) Two functions
classification_vector(Xarray,yarray,epoch)
test_error(vector,Xarray,yarray)

(d) The number of mistakes made before algorithms terminate is 423, the epochs number is 21. The estimated predictive error is 0.01944444444(when run on the test set)

(e) average_vector(Xarray,yarray,epoch)
When using this algorithm, the test error (0.0177777778) is lower than the formal one, which means preventing the overfitting.

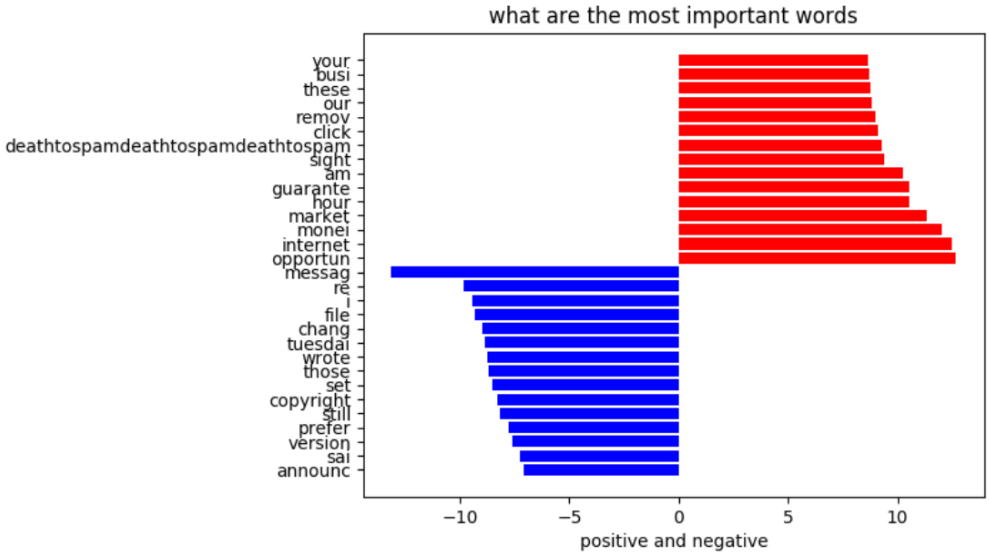
(f) As we can see from the graph, although the class perceptron (first algorithm) has better performance on train set, its performance on test set is not as good as the averaged algorithm. So the optimal algorithm is the averaged perceptron algorithm, with the maximum number of epochs is 14.



(g) I use the cross validation (5-fold) on the whole data set(both train set and test set). The expected predictive error of this algorithm is 0.0045.

(h) **the most positive words** are: ['your', 'busi', 'these', 'our', 'remov', 'click', 'deathtospamdeathtospamdeathtospam', 'sight', 'am', 'guarante', 'hour', 'market', 'monei', 'internet', 'opportun']

the most negative words are: ['messag', 're', 'i', 'file', 'chang', 'tuesdai', 'wrote', 'those', 'set', 'copyright', 'still', 'prefer', 'version', 'sai', 'announc']



Problem 4

(a) As we know, precision is $P=\frac{TP}{TP+FP}$,recall is $R=\frac{TP}{TP+FN}$

$$F_1=\frac{2PR}{P+R}, \qquad F_2=\frac{5PR}{4P+R}$$

Actually, when the data set is unbalanced (the number of positive samples and negative samples differs a lot), we cannot judge the model only by precision. In this scenario, we can tolerate being judged as ill when actually health, rather than being judged as health when actually ill. So we prefer to find every disease, which means recall is more important than precision. Therefore, we potentially are interested in optimizing for F2 score compared to F1 score.

(b) At first I partition the data into 85%-15% train-test split. Since the samples are unbalanced, I preprocess the data by standardizing. Let the mean of these 6 features: 'age','TSH','T3','TT4','T4U','FTI' be zero, and the variance be unit.

(c) **Linear SVM.** First I try larger range of the c to see if the F2 score goes up and down in the results. Then narrow down to a relatively small range. I do cross validation (5-fold) to the train set, try the c in range logspace(1, 2.6, 20), and use F2 score to find the best c, which is 102.453385939.

(d) **SVM with Polynomial kernel.** I firstly try some point values of c and d, and grasp a general idea of where they may be. Then I do cross validation (5-fold) to the train set, try the c in range of logspace(2, 3, 30), d in the range of 1-7, and use F2 score to find the best c(853.167852417) and d(2)

(e) **SVM with REF kernel.** I firstly try some point values of c and gamma, and grasp a general idea of where they may be. I do cross validation (5-fold) to the train set, try the c in range of logspace(1, 3, 30), and r in the range of logspace(-3, -0.1, 10)and use F2 score to find the best c(621.016941892) and gamma(0.00210001415571)

(f)

evaluation report:

	evaluation	Linear SVM train	Linera SVM test	SVM polynomial train	SVM polynomial test	SVM RBF train	SVM RBF train
0	misclassification	0.005042	0.021429	0.003361	0.019048	0.005882	0.019048
1	F1 score	0.888889	0.526316	0.923077	0.600000	0.860000	0.555556
2	F2 score	0.898876	0.543478	0.912548	0.638298	0.830116	0.555556

As we can see from the evaluation table, three models do quite well on the train set but not as good on the test set. Generally speaking, SVM polynomial does the best on the test set, and the second good model is SVM RBF.

When we see the misclassification rate, it seems not too bad. However, when we see the F1 and F2 score, we can see that the models could not recall well. Not only do we need to focus on the precision, but also the recall value.