EECS 738 lab 2 Zaikun Xu

Part I

experiments

First, fix c = 0.1, p = 2, vary n from 10 to 1000 and calculate the MSE

Second, fix n = 10, p = 2, vary c = 0.01 to 1 and calcualte MSE

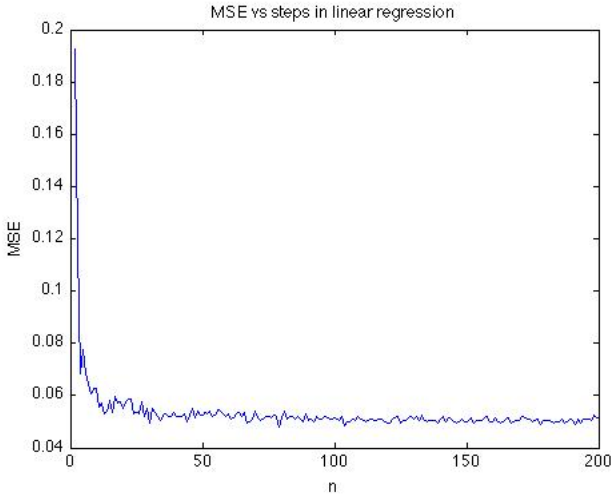
Third, fix n = 1000 > p, c= 0.1, vary p from 2 to 1000, calculate MSE

source code

```
function [m, b] = meanse(n,c, p)

x = 0;

for i= 1:30

x1 = rand(n,p-1)';

x2 = ones(1,n);

X = [x1;x2]' ;

beta = ones(p,1);

elson = rand(n,1);

Y = X * beta + c * elson;

beta_hat = inv(X'*X) * X' * Y ;

x = x + sqrt(sum((beta - beta_hat).^2));

end

m = x/30;

b = beta_hat;

end
```

result analysis

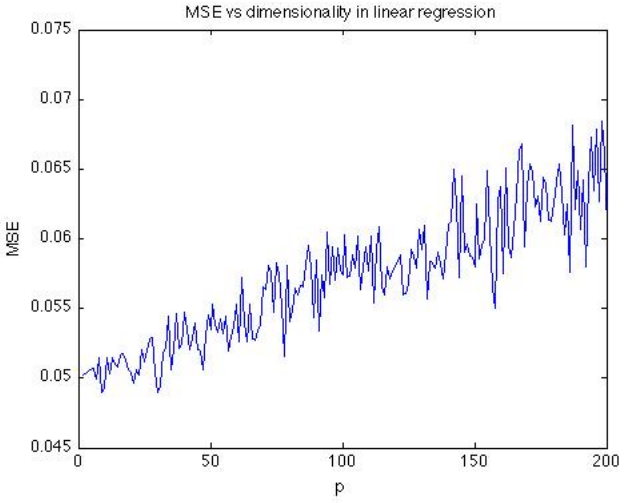
In table1, we can tell that as n increses, the MSE decreases dramatically initially and then after at about n = 20, decreases very slowly. this make sense, because as you have more data, the variance of points will be lower and thus our model can represent the ground truth better.



For table2, as c increases, the MSE increases linearly with c. This also makes sense that c contributes the variance term. as c increses, we have more noise, then the MSE will increase and should scale linearly with c.

For table3, as P increases, the MSE increases according, in a linear pattern. It also makes sense. As you incerase the dimensonality of data, the effect of

curse of dimensionality will be more obvious



Part II (weka)

Introduction

dataset

The dataset is a 1001 by 1559 matrix, where each column is data point and each row is one feature. the data include two classes: advertisement image and non-advertisement image. Features include image url, anchor text, but there are lots of missing values.

model

I choose three models, the Random Forest, Navie Bayes and the RBF neural netwrok.

The idea of Random Forest is combine the idea of "bagging ", where you sample the data multiple times and the ideas of random selection of features. It works by combining many decision trees to improve the prediction accuracy.

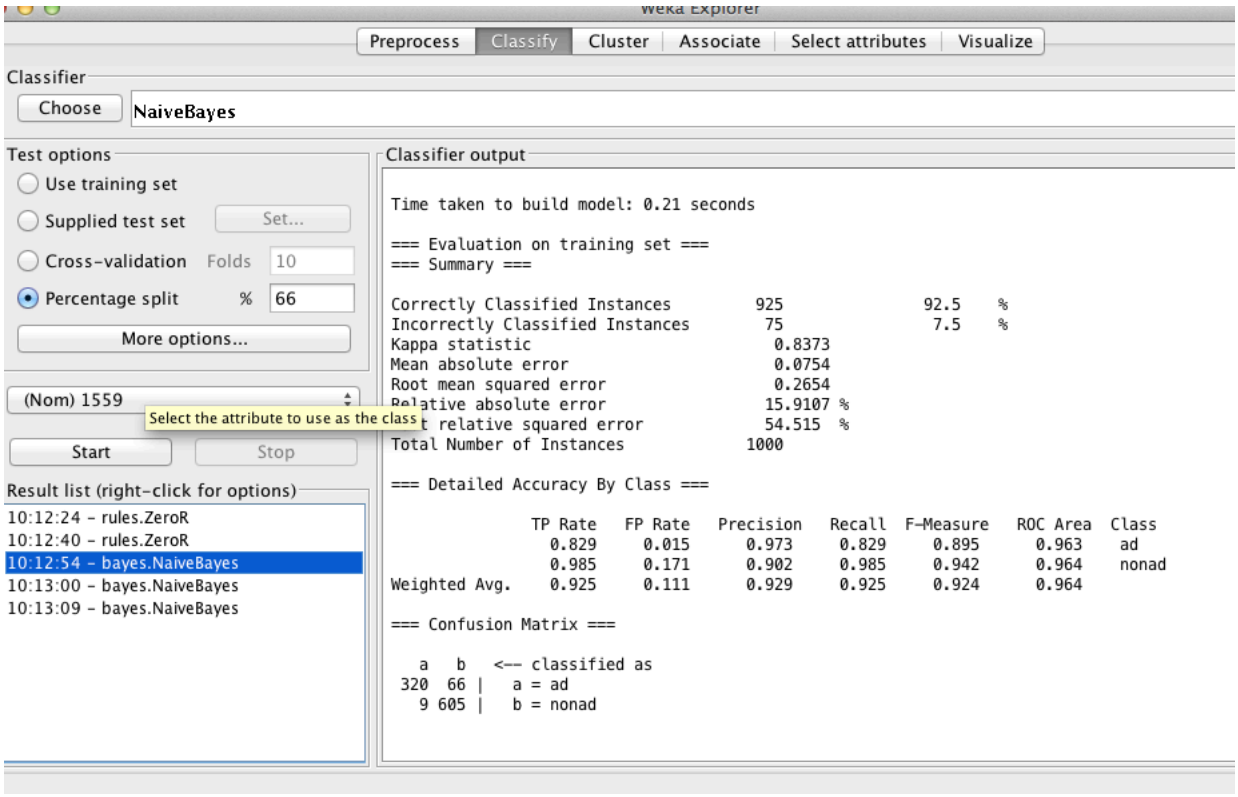
The idea of Naive Bayes is based on the bayes rule and have the assumption (which might be wrong, but in reality it works well) that one feature is independent of another give a class variable. It is a probabilistic model.

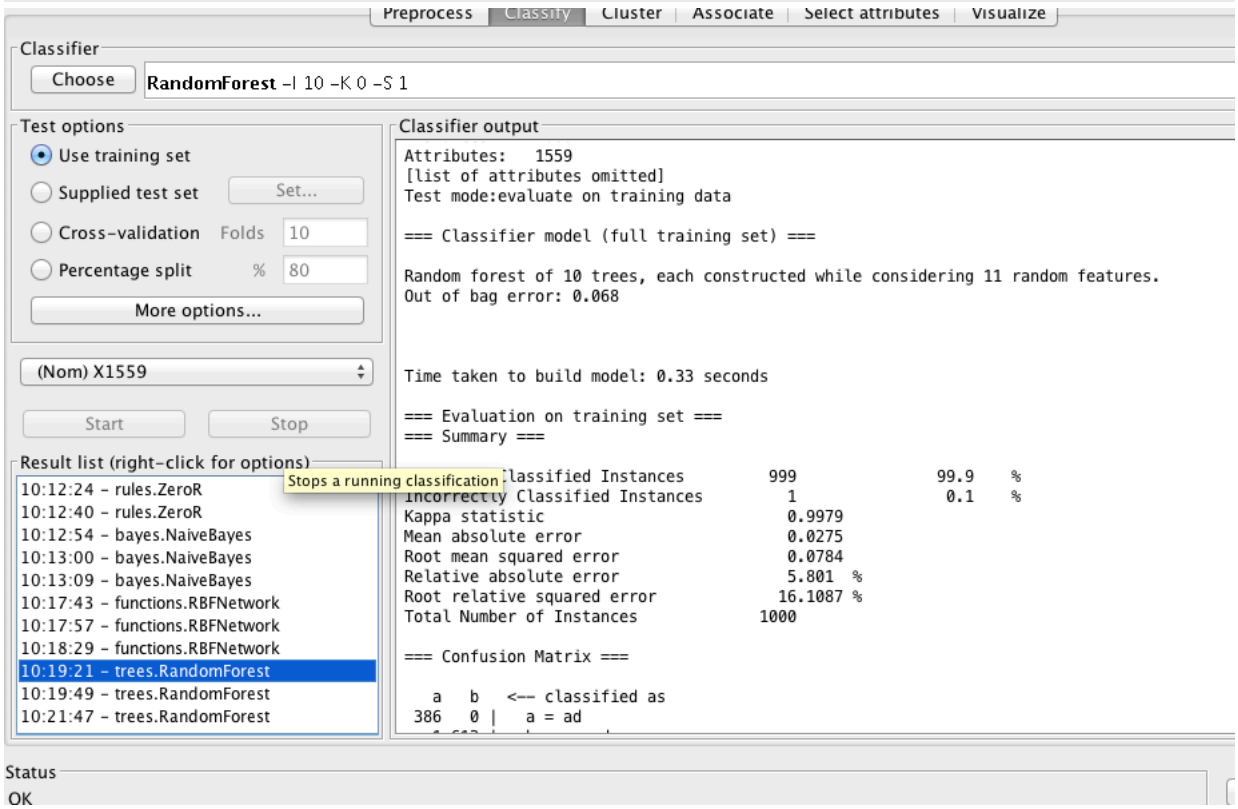
The idea of RBF neural network is as follows: the unknow function f(x) can be approximated by the weighted sum of neurons. From the input layer to the hidden layer, you have a activition funciton , which is a radial basis function. By propagating and udate weights of each neurals, the trained outputs will approximated their real values.

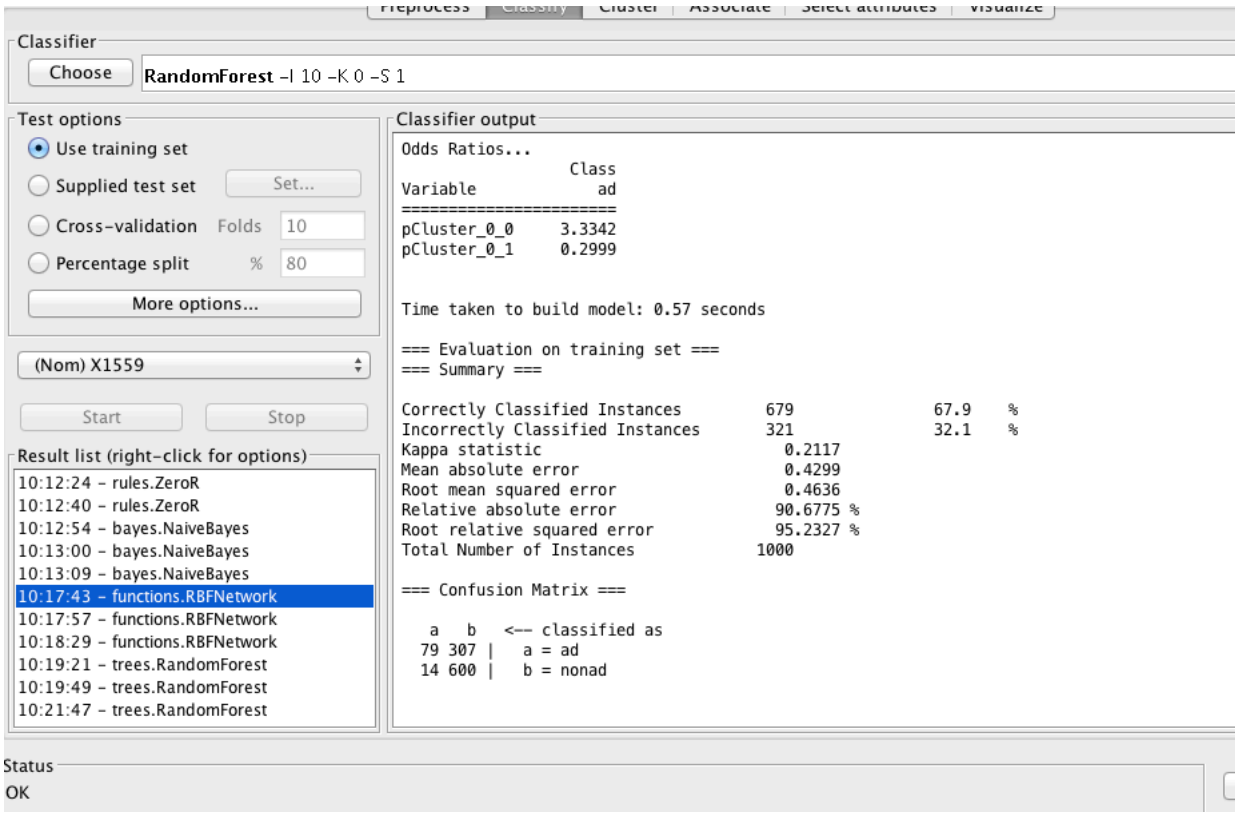
Experiments

For each model, use there different options, namesly, Using training set, cross-validation and percentage split to see how these choices influence the performance of each model.

screen prints for each classifier


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) 1559
Select the attribute to use as the class
Start
Stop
Result list (right-click for options)
10:12:24 - rules.ZeroR
10:12:40 - rules.ZeroR
10:12:54 - bayes.NaiveBayes
10:13:00 - bayes.NaiveBayes
10:13:09 - bayes.NaiveBayes
Classifier output
Time taken to build model: 0.21 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 925 92.5 %
Incorrectly Classified Instances 75 7.5 %
Kappa statistic 0.8373
Mean absolute error 0.0754
Root mean squared error 0.2654
Relative absolute error 15.9107 %
Root relative squared error 54.515 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
0.829 0.015 0.973 0.829 0.895 0.963 ad
0.985 0.171 0.902 0.985 0.942 0.964 nonad
Weighted Avg. 0.925 0.111 0.929 0.925 0.924 0.964
=== Confusion Matrix ===
a b <-- classified as
320 66 | a = ad
9 605 | b = nonad


Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
RandomForest -I 10 -K 0 -S 1
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) X1559
Start
Stop
Result list (right-click for options)
Stops a running classification
10:12:24 - rules.ZeroR
10:12:40 - rules.ZeroR
10:12:54 - bayes.NaiveBayes
10:13:00 - bayes.NaiveBayes
10:13:09 - bayes.NaiveBayes
10:17:43 - functions.RBFNetwork
10:17:57 - functions.RBFNetwork
10:18:29 - functions.RBFNetwork
10:19:21 - trees.RandomForest
10:19:49 - trees.RandomForest
10:21:47 - trees.RandomForest
Classifier output
Attributes: 1559
[list of attributes omitted]
Test mode:evaluate on training data
=== Classifier model (full training set) ===
Random forest of 10 trees, each constructed while considering 11 random features.
Out of bag error: 0.068
Time taken to build model: 0.33 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 999 99.9 %
Incorrectly Classified Instances 1 0.1 %
Kappa statistic 0.9979
Mean absolute error 0.0275
Root mean squared error 0.0784
Relative absolute error 5.801 %
Root relative squared error 16.1087 %
Total Number of Instances 1000
=== Confusion Matrix ===
a b <-- classified as
386 0 | a = ad
Status
OK



results and analysis

	Random Forest			Navie Bayes			RBF neural network		
1000instance	Training set	Cross-validation	80% to 20 % split	Training set	Cross-validation	80% to 20 % split	Training set	Cross-validation	80% to 20 % split
Training error	0.0275	0.075	0.068	0.0754	0.0782	0.0568	0.4299	0.4542	0.4322
SD	0.0381	0.176	0.1581	0.2654	0.27	0.2195	0.4636	0.475	0.4556

Compare) between models

The RBF works not good, the training erorr is around 0.4. Both the Navie Baye and Random Forest work better and Random Forest works best in both training error and standard derivation.

Compare between training choices

In both the navie bayes and RBF,the 80% split works better with low training errors and low standard derivation. In the random forest case, the case is different, where you have low training error with the training data only.