Final Project Report:

Application of various learning methods on some benchmark data sets

STAT 542: Statistical Learning

Professor Feng Liang

Group members:

Javad Ghaderi, jghaderi@illinois.edu, ECE Department
Siva Kumar Gorantla, sivaiitm@gmail.com, ECE Department
Bo Tan, botan2@uiuc.edu, ECE Department

Fall 2010

Introduction

The goal of this project is to apply different learning schemes on 4 data sets, namely Spam, Housing, and two NIPS challenge data sets Gisette (confusable handwritten digits) and Dexter (corporate acquisitions). We will apply different classification (or regression) techniques such as KNN, naive Bayes, Full regression model, AIC, BIC, Ridge, Lasso, Logistic Regression, LDA, QDA, RDA, SVM with different kernels, classification (or regression) tree, and Boosting Algorithms to the data sets. We aim to extract the important features of each data set and compare the performance of the different classification (or regression) methods. Finally, we will compare our performance results to those reported in the literature.

Next, we give a short description of each data set.

Spam

In this data set of 4601 instances, the collection of spam e-mails came from some postmaster and individuals who had filed spam, and the collection of non-spam e-mails came from filed work and personal e-mails. These are useful when constructing a personalized spam filter. Note that "false positives" (marking good mail as spam) are very undesirable in spam filtering.

Features: There are 58 attributes. 57 of them are continuous feature values, and the last one column represents the nominal class label, indicating whether the e-mail was considered spam (1) or not (0). Most of the features indicate whether a particular word or character was frequently occurring in the e-mail. The run-length features (Col. 55-57) measure the length of sequences of consecutive capital letters.

Housing

The Boston housing data consist of 506 instances and concerns housing values in suburbs of Boston. The goal is to find out how different attributes such as crime rate, nitric oxide concentration, pupil-teacher ratio, and etc affect house prices. There are 12 continuous features and one binary-valued feature. Since most of the variables exhibit an asymmetry with a higher density on the left side, some transformations have been proposed such as

taking the logarithm or raising the variables to the power of something smaller than one to reduce the asymmetry. This is due to the fact that lower values move further away from each other, whereas the distance between greater values is reduced by these transformations.

Dexter

The data represents frequency of appearance of words (word stems) formatted in a "bag-of-words" representation that are collected from Reuters articles. The task is to learn which Reuters articles are about "corporate acquisitions".

Features: There are 9947 features representing frequency of word stems in data. An additional 10053 (dummy) features are added. These features are drawn independently using Zipf law. The Zipf law states that the frequency of occurrence of words is determined by power-law function $P_k \sim 1/k^a$ QUOTE , where k is the rank depending on freq of occurrence, and a is the exponent (which is chosen to be 0.9 for generating the dummy features). The number of non-zero values in the data is 0.5% and hence, the data is "sparse".

Gisette

The goal is to discriminate between two confusable handwritten digits: the four and the nine. This is a two-class classification problem with sparse continuous input variables. The data set was constructed from the MNIST data set of images of dimension 28*28. The feature set consists of the original variables (normalized pixels) plus a randomly selected subset of products of pairs of variables to attain a number of 2500 features. Another 2500 pairs have been used to construct "probes": the values of the features were individually permuted across patterns (column randomization) to obtain probes that are similarly distributed to the other features. This produces the Gisette data set of size 6000*5000.

Procedure

The data sets Spam and Housing have small number of features so no preprocessing on the data is necessary. The data sets Dexter and Gisette have large number of features, respectively, 20000 and 5000, so we need some forms of preprocessing on these data sets to make computations tractable by the CPUs. Dexter has only 300 samples and therefore it has only 300 nonzero singular values. So, for the Dexter, in general, we project the data on

the 300 right eigenvectors to get a 300*300 training data. Such a transformation is not feasible for the Gisette as singular values decay very slowly. For example, the top 1000 PC directions contain less than 50% of the variations in the Gisette data. This can be due to the permutation scheme used in the construction of the Gisette. Instead, to extract the important features, we have used the marginal significance test for each feature, i.e., using the student t-test for two groups of data in each column and ignoring the columns that have high p-values. In fact, we kept only the features corresponding to the top 1000 p-values since the reduced data matrix seems to be tractable by our CPUs and memories.

Next, we briefly explain the methods that we have used and their corresponding parameters.

K Nearest Neighbours (KNN)

We test k = [1,3,5,7,11,21,31,45,69,101,151] and pick the k with the smallest test error.

Linear Regression

The application of linear regression methods on the data sets was fairly easy. To perform further model selection based on AIC/BIC, we have used exhaustive search when the dimension was less than 50 and stepwise algorithms when the dimension was greater than 50.

Naïve Bayes

Naïve Bayes method is vulnerable to zero-variance features, so before predicting the test data with this approach, we need to eliminate those columns with zero variances.

Tree

For Housing data we use regression tree, and for the other three we use classification tree. Our approach is as follows: 1. generate a full tree and observe the test error; 2. use 5-fold CV to pick the best size; 3. prune the tree to the best size according to two impurity measures, including deviation and misclassification rate (if it is a regression tree, only "deviation" can be used), and see which one has a lower test error.

In R, for a classification tree, "predict" will generate a two-column matrix, where the second column is the probability that the test sample is in class 1. We need to use 0.5 as a

threshold to quantify the prediction to 1 or 0.

Boosting Algorithms

We have used the AdaBoost exponential loss for 0/1 classification problems and the squared error loss (called gaussian in the gbm package in R) for the Housing data. For a fixed shrinkage, we used a 5-fold CV to choose the optimal number of iterations. It seems that smaller values of shrinkage always give improved predictive performance However, there are computational costs, both storage and CPU time, associated with setting shrinkage to be low. The shrinkage value that we have used in our model is 0.01. The value of the bag.fraction, to determine the fraction of the training set observations randomly selected to propose the next tree in the expansion, is chosen to be 1 or 0.5 depending on the data set. The exact value of bag.fraction will be clear from the tables in the empirical results section.

Ridge and Lasso Regression

We use the Ridge and Lasso regularization techniques in addition to full regression. For ridge regression, we look for the optimal parameter lambda, by iterating through various values of lambda from 0 to 50. We choose the lambda which minimizes the GCV value. For Lasso regression, we choose the model which gives the least RSS value. Its not surprising from Spam and Housing results that the model chooses all parameters in the final model because all parameters have a reasonable weightage in the full regression.

Discriminant Analysis

We perform Linear, Quadratic and Regularized Discriminant Analysis modeling the problems as a Classification problem. We use the standard LDA,QDA,RDA packages in the MASS library to evaluate results. For RDA, the optimal values of gamma and lambda are picked by the rda() command and we reported them in the empirical results section below. For the regression problem (Housing), its not possible to apply Discriminant analysis.

Support Vector Machines

SVMs can be used for both classification and Regression problems (Support Vector Regression). We have used SVM or SVR with different kernels to all the data sets. The parameter cost (C) is varied through 1 to 10 to find out the optimal cost. For the housing data (regression problem), we use epsilon based SVR (eps-svr) model with

default epsilon value (0.1). We tried "Linear, Quadratic and Gaussian" kernels for all the data sets. For the benchmark datasets, we have also looked at various Polynomial (k=2,3,4,5) kernels and chose the one with the best performance.

Empirical Results

3.1- Spam

Method	Test Error Rate (%)	Parameters
KNN	19.50	K = 1
Linear Regression Full	10.91	
Linear Regression AIC	11.35	Stepwise, optimal step-size = 43
Linear Regression BIC	12.05	Stepwise, optimal step-size = 28
Naïve Bayes	29.33	
Classification Tree	9.29	Size = 26 (full tree; pruning has no improvement)
Adaboosting	5.5	# of iterations=2995, bag.fraction=1
Ridge	11.02	Lambda = 6.5
Lasso	11.13	Variables in Final model = 57 (all)
LDA	10.91	
QDA	17.11	
RDA	32.53	gamma = 0.9973356, lambda = 0.9999842
Logistic Regression	8.36	
SVM-Linear	6.73	cost C = 8.5
SVM-Quadratic	9.07	Cost C = 1.0
SVM-Gaussian	7.22	Cost C = 1.0

3.2- Housing

Method	Mean Square Error	Parameters
Linear Regression Full	0.03646	
Linear Regression AIC	0.03601	Exhaustive
Linear Regression BIC	0.03789	Exhaustive
Regression Tree	0.03767	Best size = 12 (based on 5-fold CV)
Boosting	0.01116	# of iterations=12579, bag.fraction=1
Lasso Regression	0.03635	Lambda = 7.5
Ridge Regression	0.03674	Number of variables added = 15
Support Vector Regression – Linear	0.04128	Cost C = 2.5, Num of support vectors = 233
Support Vector Regression – Quadratic	0.03059	Cost C = 1, Num of support vectors = 228
Support Vector Regression – Gaussian	0.02576	Cost C = 1, Num of support vectors = 213

3.3- Dexter

Method	Test Error Rate (%)	Parametrs
KNN	9.33	K = 11
Linear Regression Full	45.3	
Linear Regression AIC	45.3	Exhaustive
Linear Regression BIC	45.3	Exhaustive
Naïve Bayes	25.33	
Classification Tree	50.33	Best size = 2 (based on 5-fold CV; two impurity measures have the same results)

Adaboosting	11.3	iterations=2562, bag.fraction=0.5
LDA	43	
QDA	50	Predicts +1 for all.
RDA	50	Gamma = 0.7641465 lambda = 0.5846650 Predicts +1 for all.
SVM – Linear with L2 normalization of data	7	Cost C = 9.5
SVM – Poly(k) , k=2,3,4,5 with L2 normalization of data	7	K=2 (Quadratic polynomial) shows the best performance Cost C = 9.5
SVM-Rad With L2 normalization of data	13.33	Cost C = 1

3.4- Gisette

Method	Test Error Rate %	Parameters
KNN	4.7	K = 5 or 7
Linear Regression (Full Model)	31.8	
Naive Bayes	14.1	
Classification Tree	7.7	Best size = 14 (based on 5-fold CV and "misclassification rate" impurity measure)
Adaboosting	4.9	iterations=5243, bag.fraction=0.5

LDA	51.6	Used only top 250 features selected through p-value test. LDA,QDA,RDA gave errors otherwise.
QDA	48	Used only top 250 features selected through p-value test.
RDA	44	Gamma= 0.6834070, lambda = 0.7396465 Used only top 250 features selected through p-value test.
Logistic	49.4	
SVM- Linear	49.8	Cost C = 1
SVM – Quadratic	49.4	Cost C = 1
SVM- Gaussian	49.8	Cost C = 1

Conclusions:

Adaboosting techniques, Classification or Regression trees and Support Vector Machines (SVM or SVR), seem to perform better consistently over all datasets.

For the Dexter dataset, SVM after doing L2 regularization (data pre-processing) over the data performs the best (7% error rate) while the best benchmark performance is 3.15%

For the Gisette dataset, Adaboosting works the best after picking top 1000 features (with low p-value) with a performance of 4.9% error rate while the best benchmark performance is 0.7%.