MIS 382N: Advanced Predictive Modelling

Assignment #5

Due: Sun, Nov 23 11.59pm, 2014; Total points: 40

Note the atypical due date and time.

Upload instructions: Your homework should be a single pdf file containing all the answers, plots/pictures and code. You may however insert equations by hand if you wish. Name the pdf file as <your eid>.pdf The included code should be straightforward to run by just copying it into the interpretrator or the command line without any changes. So add any include statements for libraries etc. in the code as well. If the choice of a programming language is specified in the problem, please don't use any other language.

1. **(4 pts) Multi-layer Perceptron for Supervised Learning**

   (a) In this question, you will use a Multilayer Perceptron (MLP) network to classify hand-written digits. You can use the R-package '*monmlp*' to train the MLP.

   Load training data '*pendigits.tra*' uploaded on Canvas. The last column is the ground-truth label y.

   Use only one hidden layer and assume the hidden layer nodes use the *logistic* activation function while the output node uses *linear* activation. Train your MLP for the following settings:

No. of hidden layer nodes	Maximum Epochs		
5	500	1000	2000
10	100	1000	2000
15	100	1000	2000

   Analyze performance changes (by observing the confusion matrix and accuracy on test data '*pendigits.tes*') for different parameter settings. More specifically, comment on performance when the network has fixed number of hidden units but trained for more epochs. Also comment for the case when number of hidden nodes are increased but trained for less number of epochs. Report the best parameter setting as measured by accuracy on the test set.

   Note: The last column of '*pendigits.tes*' is the ground-truth for test samples.

2. **(6+6+6=18 pts) Imbalanced data** In this question, you will explore three ways of doing classification on imbalanced data - up/down sampling, cost sensitive regression trees, and cost sensitive SVMs. This question should be done in R.

   The data used in this question is adapted from `Adult` dataset from the UCI repository. After removing `NA` values, data has been divided into training and test data. It is available as saved R workspace `imbalanced.RData` in canvas which has variables `trdata` and `tedata` for training and test respectively. The task is to predict `income` as `large` or `small` using all other variables.

   Important: For consistency, run `set.seed(10)` to initialize the random seed at the beginning of your script before running any experiments. This will generate the same sequence when generating random numbers across different runs of the script.

   (a) **Sampling Methods.** Fit a logistic regression (package `glm`) on the given training data, and report AUC for the test data. You can use the package `ROCR` in R. Now, use commands `upSample` and `downSample` in the package `caret` to create two new training datasets. Note how the dependent variable's distribution changes in the three training datasets. Run two logistic regressions on the two newly generated training sets, and report the AUC on the test set. Syntax for `upSample` (its similar for `downSample`; fill in the dots):

```
upSample(x = ..., y = ..., yname = "income")
```

(b) **Regression trees.** You practised using `rpart` in a previous homework. It can handle optimizing for different costs of misclassification. First, build a regression tree for the given training data, prune appropriately and report the AUC on the testset. Look at the option `parms=list(loss=costMatrix)`, where `costMatrix` will be a 2X2 matrix with 0s on the diagonal that encodes the different costs of misclassification. Set the matrix to represent costs so that it is twice as costly to misclassify a `small` income as `large` than otherwise. Re-build the tree on the training data, prune appropriately and report the AUC on the testset.

(c) **SVMs** Use `ksvm` function in the package `kernlab` to fit an SVM. Use `vanilladot` as the kernel (this is the linear kernel). While predicting, use `type='r'` to get predicted labels directly. This is important because predicted class probabilities do not correspond to the predictions directly if the costs associated with labels are different. Fit two models - one without the option `class.weights`, and another with `class.weights` set so that the label `large` has twice as much cost as `small`. Report AUC values for both the models. You might have to convert the predicted labels to numbers 0,1 before using `ROCR`'s commands for AUC.

3. **(4+4+2=10 pts) Support Vector Regression.** In this question, you will explore using SVR in python's `sklearn.svm`. You are provided with `data.train` and `data.test` (available on canvas). The data is based on `auto-mpg` dataset from UCI repository. Some outliers have been artificially added in the training set.

   (a) Use `sklearn.svm.SVR` to fit a support vector regression model on the training set, use the default options. Report the training and test RMSE.

   (b) Fit a least squares regression. Again, report the train and test RMSE. Using the RMSE numbers of the two models, comment on the respective model's ability to handle outliers.

   (c) Can you suggest another model for regression to handle outliers?

4. **(2+6=8 pts) Ensemble methods** In this question, you will apply a simple ensemble of logistic regressions for spam classification. Use the data `spam.csv` (the same dataset was used in HW4-Q4, refer to it for details). Use python for this question. Load and standardize the data.

   (a) Fit a logistic regression using `sklearn.linear_model.LogisticRegression` on the training set, and report the prediction accuracy on the testset. A prediction is correct if predicted_class=true_class, and is incorrect otherwise.

   (b) Build an ensemble of logistic regressions as follows. Say n is the number of member logistic regressions in the ensemble. For each member, randomly sample with replacement 1000 points from the training data, use the sampled points as the training data and predict on the test data. After all n members have made their predictions on the test data, select final prediction as the one made by majority of the members. Use this final prediction to calculate accuracy. Report accuracy numbers for $n = 1, 11, 21, 31$.