CSCI-B565: Data Mining

Homework Assignment # 2

Assigned: 02/08/2016 *Due: 02/25/2016, 11:59pm, through Oncourse*

Six questions, 130 points in total. Good luck!
Prof. Predrag Radivojac, Indiana University, Bloomington

All data sets relevant for this assignment can be found in the UCI Machine Learning Repository. It is available at

`http://archive.ics.uci.edu/ml/`

Problem 1. (10 points) Download data set Iris and answer the following questions:

a) (2 points) Calculate the average value and standard deviation for each of the four features.

b) (3 points) Repeat the previous step but separately for each type of flower.

c) (5 points) Draw four box plots, one for each feature, such that each figure shows three boxes, one for each type of flower. Properly label your figures and axes in all box plots. Make sure that the box plots look professional and appear in high resolution. Experiment with thickness of lines, font styles/sizes, etc. and describe what you tried and what looked the most professional.

The data set Iris can be downloaded from `http://archive.ics.uci.edu/ml/datasets/iris`. You may use libraries or built-in functions from any software package you choose.

Problem 2. (15 points) Download data set Wine and answer the following questions:

a) (5 points) Provide pairwise scatter plots for four most correlated and four least correlated pairs of features, using Pearson's correlation coefficient. Label all axes in all your plots and select fonts of appropriate style and size. Experiment with different ways to plot these scatter plots and choose the one most visually appealing and most professionally looking.

b) (5 points) Use Euclidean distance to find the closest example to every example available in the data set (exclude the class variable). Calculate the percentage of points whose closest neighbors have the same class label (for data set as a whole and also for each class).

c) (5 points) Repeat the previous step but after the data set is normalized using first 0-1 normalization and then z-score normalization. Investigate the reasons for discrepancy and provide evidence to support every one of your claims. Provide the code you used for normalizing and visualizing the data.

Data set Wine can be downloaded from `http://archive.ics.uci.edu/ml/datasets/wine`.

Problem 3. (25 points) Data exploration is often the first step in many data analysis tasks. Visualizing relationships between features as well as between features and the target variable(s), for example, can be

exploited to design a good model or to understand why a particular model works. There are many software packages developed to make this step easier. In this question you will experiment with Tableau.[1] Tableau can be downloaded from

`http://www.tableau.com/academic/students`.

a) (10 points) Download and study the Auto MPG data set from the UCI Machine Learning Repository. Import the data set into Tableau. Create a new feature make (Honda, Toyota, ...) that contains the make of the automobile (extract this feature automatically from other features through Tableau) and in a single figure generate box plots of mpg for 10 makes of your choice. Then, for 5 makes of your choice create scatter plots of weight versus mpg. Include all figures in your submission and comment on what you observe.

b) (10 points) Pick 3 data sets of your choice from UCI Machine Learning Repository. Visualize each the data set in meaningful ways that show hidden patterns. Experiment with colors, size, shapes, filters, groups and sets. Feel free to experiment with other advanced functionalities of Tableau.

c) (5 points) Tell us about your experience with Tableau. What did you learn? What did you like/dislike about Tableau?

The data set Auto MPG can be found at `https://archive.ics.uci.edu/ml/datasets/Auto+MPG`.

Problem 4. (35 points) Implementing classification trees and evaluating their accuracy.

a) (15 points) Implement the greedy algorithm that learns a classification tree given a data set. Assume that all features are numerical and properly find the best threshold for each split. Use Gini and information gain, as specified by user, to decide on the best attribute to split in every step. Stop growing the tree when all examples in a node belong to the same class or the remaining examples contain identical features.

b) (15 points) Implement 10-fold cross-validation to evaluate the accuracy of your algorithm on 10 different data sets from the UCI Machine Learning Repository. Select only those data sets where all features are numerical. In certain cases you can convert categorical features into numerical by encoding them using sparse binary representation. That is, if feature values belong to a set $\{\mathsf{blue}, \mathsf{yellow}, \mathsf{red}, \mathsf{green}\}$, encode this feature using 4-dimensional binary vectors such that if the feature value is blue, the encoding is $(1, 0, 0, 0)$, if the feature value is yellow, the encoding is $(0, 1, 0, 0)$, etc. You can also transform regression problems from the repository into classification problems by using the mean of the target variable to dichotomize the continuous target into binary class labels.

c) (5 points) Compare Gini and information gain as splitting criteria and discuss any observation on the quality of splitting.

You can implement a classification tree using standard recursive partitioning of the data or you can choose to implement it without recursion.

Problem 5. (20 points) Modify your decision tree from the previous question and evaluate the overfitting prevention methods listed below. All evaluation should be carried out using 10-fold cross-validation as implemented in the previous Problem, but the performance should be evaluated using multiple measures.

a) (5 points) Use 'pessimistic' estimates of the generalization error by adding a penalty factor 0.5 for each node in the tree (see Textbook page 181).

[1] Tableau is free for students.

b) (5 points) Use a validation set that consists of 25% of the training partition.

c) (10 points) Use the minimum description length principle, as explained in Question #8, page 201 of your Textbook.

Evaluate all three overfitting prevention techniques using at least 10 different data sets from the UCI Machine Learning Repository. To evaluate performance, calculate simple accuracy, balanced sample accuracy, and the F_1-measure. For each of the data sets, plot the ROC curves as well as the precision-recall curves. Label all axes and make all figures look as professional as you can.

Problem 6. (25 points) Beyond greedy: combining beam search and classification tree construction. Extend the greedy algorithm for tree learning developed in the previous two questions to find a better classification tree as follows: At each step of node splitting, instead of picking the single best attribute to find the successor tree, keep top m successor trees based on the splits according to the top m attributes. For each of the trees in the list, generate the successor tree by splitting according to the next attribute. Then, keep top m successor trees (from all generated trees) for the next step and drop all the others. Continue the process until the stoppage criteria are fulfilled and select the best remaining tree as your model. Compare this algorithm with the greedy classification tree learning algorithm ($m = 1$) and comment on what you observe. Feel free to adjust this algorithm and experiment with different m's and other parameters to see whether there is improvement in accuracy.

Homework Directions and Policies

Submit a single package containing all answers, results and code. You submission package should be compressed and have extension .zip. In your package there should be a single pdf file named main.pdf that will contain answers to all questions, all figures, and all relevant results. Your solutions and answers must be typed[2] and make sure that you type your name and IU username (email) at the beginning of the file. The rest of the package should contain all code that you used. The code should be properly organized in folders and subfolders, one for each question or problem. All code, if applicable, should be turned in when you submit your assignment as it may be necessary to demo your programs to the associate instructors. Use Matlab, Python, R, Java, or C/C++.

Unless there are legitimate circumstances, late assignments will be accepted up to 5 days after the due date and graded using the following rules:

on time: your score $\times$ 1

1 day late: your score $\times$ 0.9

2 days late: your score $\times$ 0.7

3 days late: your score $\times$ 0.5

4 days late: your score $\times$ 0.3

5 days late: your score $\times$ 0.1

For example, this means that if you submit 3 days late and get 80 points for your answers, your total number of points will be $80 \times 0.5 = 40$ points.

All assignments are individual, except when collaboration is explicitly allowed. All the sources used for problem solution must be acknowledged; e.g., web sites, books, research papers, personal communication with people, etc. Academic honesty is taken seriously! For detailed information see Indiana University Code of Student Rights, Responsibilities, and Conduct.

[2]We recommend Latex; in particular, TexShop-MacTeX combination for a Mac and TeXnicCenter-MiKTex combination on Windows. An easy way to start with Latex is to use the freely available Lyx. You can also use Microsoft Word or other programs that can display formulas professionally.