FIT2086 Assignment 3

Due Date: 11:59PM, Sunday, 22/10/2017

Introduction

There are total of two questions worth $14 + 14 = 28$ marks in this assignment.

This assignment is worth a total of 20% of your final mark, subject to hurdles and any other matters (e.g., late penalties, special consideration, etc.) as specified in the FIT2086 Unit Guide or elsewhere in the FIT2086 Moodle site (including Faculty of I.T. and Monash University policies).

Students are reminded of the Academic Integrity Awareness Training Tutorial Activity and, in particular, of Monash University's policies on academic integrity. In submitting this assignment, you acknowledge your awareness of Monash University's policies on academic integrity and that work is done and submitted in accordance with these policies.

Submission: No files are to be submitted via e-mail. Correct files are to be submitted to Moodle, as given above. You must submit your files in a single ZIP archive. Your ZIP file should contain the following:

1. One PDF file containing non-code answers to all the questions that require written answers. This file should also include all your plots.

2. The required R script files containing R code answers.

Please read these submission instructions carefully and take care to submit the correct files in the correct places.

Question 1 (14 marks)

In this question you will use R to analyse a dataset. The data is contained in the file `heart.csv`. In this dataset, each observation represents a patient at a hospital that reported showing signs of possible heart disease. The outcome is presence of heart disease (`HD`), or not, so this is a classification problem. The predictors are summarised in Table 1 (overleaf). We are interested in learning a model that can predict heart disease from these measurements. To answer this question you must:

- Provide an R script containing all the code you used to answer the questions. Please use comments to ensure that the code used to identify each question is clearly identifiable. Call this `fn.sn.Q1.R`, where "`fn.sn`" is your first name followed by your family name.
- Provide appropriate written answers to the questions, along with any graphs, in a non-handwritten report document.

When answering this question, you must use the `tree` package that we used in Studio 9. The wrapper function for learning a tree using cross-validation that we used in Studio 9 is contained in the file `tree.wrappers.R`. Don't forget to source this file to get access to the function.

1. Using the techniques you learned in Studio 9, fit a decision tree to the data using the `tree` package. Use cross-validation with 10 folds and 1000 repetitions to select an appropriate size tree. What variables have been used in the best tree? How many leaves (terminal nodes) does the best tree have? **[2 marks]**
2. Plot the tree found by CV, and discuss what it tells you the relationship between the predictors and heart disease. *(hint: use the* `text(cv$best.tree,pretty=12)` *function to add appropriate labels to the tree).* **[4 marks]**
3. For classification problems, the `tree` package only labels the leaves with the most likely class. However, if you examine the tree structure in its textural representation on the console, you can determine the probabilities of having heart disease (see Question 2.3 from Studio 8 as a guide) in each leaf (terminal node). Take a screen-capture of the plot of the tree (don't forget to use the "zoom" button to get a larger image) or save it as an image using the "Export" button in R Studio.

   Then, use the information from the textual representation of the tree available at the console and annotate the tree in your favourite image editing software; next to all the leaves in the tree, add text giving the probability of contracting heart disease. Include this annotated image in your report file. **[2 marks]**
4. According to your tree, which predictor combination results in the highest probability of having heart-disease? **[1 mark]**
5. We will also fit a logistic regression model to the data. Use the `glm()` function to fit a logistic regression model to the heart data, and use stepwise selection with the BIC score to prune the model. What variables does the final model include, and how do they compare with the variables used by the tree estimated by CV? **[2 marks]**
6. Write down the regression equation for the logistic regression model you found using step-wise selection. **[1 mark]**
7. Calculate the *odds* of having heart disease given for a patient with characteristics listed in Table 2. The odds should be calculated for both (i) the tree model found using cross-validation and (ii) the step-wise logistic regression model. How do the predicted odds for the two models compare? **[2 marks]**

Variable name	**Description**	**Values**
AGE	Age of patient in years	$29 - 77$
SEX	Sex of patient	M = Male F = Female
CP	Chest pain type	Typical = Typical angina Atypical = Atypical angina NonAnginal = Non anginal pain Asymptomatic = Asymptomatic pain
TRESTBPS	Resting blood pressure (in $mmHg$)	$94 - 200$
CHOL	Serum cholesterol in mg/dl	$126 - 564$
FBS	Fasting blood sugar $> 120mg/dl$?	<120 = No >120 = Yes
RESTECG	Resting electrocardiographic results	Normal = Normal ST.T.Wave = ST wave abnormality Hypertrophy = showing probable hypertrophy
THALACH	Maximum heart rate achieved	$71 - 202$
EXANG	Exercise induced angina?	N = No Y = Yes
OLDPEAK	Exercise induced ST depression relative to rest	$0 - 6.2$
SLOPE	Slope of the peak exercise ST segment	Up = Up-sloping Flat = Flat Down = Down-sloping
CA	Number of major vessels colored by flourosopy	$0 - 3$
THAL	Thallium scanning results	Normal = Normal Fixed.Defect = Fixed fluid transfer defect Reversible.Defect = Reversible fluid transfer defect
HD	Presence of heart disease	N = No Y = Yes

Table 1: Heart Disease Data Dictionary. ST depression refers to a particular type of feature in an electrocardiograph (ECG) signal during periods of exercise. Thallium scanning refers to the use of radioactive Thallium to check the fluid transfer capability of the heart.

AGE	SEX	CP	TRESTBPS	CHOL	FBS	RESTECG	THALACH	EXANG	OLDPEAK	SLOPE	CA	THAL
55	M	Asymptomatic	140	217	<120	Normal	111	Y	5.6	Down	0	Reversible.Defect

Table 2: Characteristics of a patient attending hospital with heart pain.

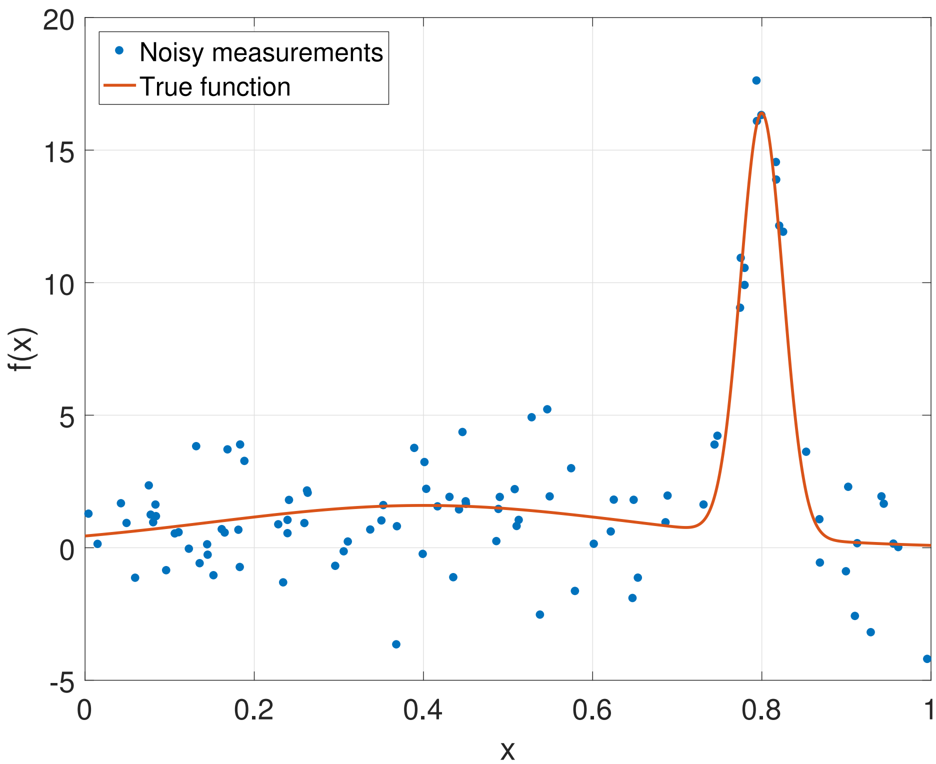

Figure 1: Noisy measurements (`q2.train$y`) and the true (in real life, unknown) function (`q2.test$y`).

Question 2 (14 marks)

Introduction

Data "smoothing" and interpolation is a very common problem in data science and statistics. We are often interested in examining the unknown relationship between a dependent variable (y) and an independent variable (x), under the assumption that the dependent variable has been imperfectly measured and has been contaminated by measurement noise. The model of reality that we use is

$$y = f(x) + \varepsilon$$

where $f(x)$ is some unknown, "true", potentially non-linear function of x, and ε is a random disturbance or error. This is called the problem of function estimation, and the process of estimating $f(x)$ from the noisy measurements y is sometimes called "smoothing the data" (even if the resulting curve is not "smooth" in a traditional sense). In this question you will use several tools to try and estimate the underlying function $f(x)$ from some provided noisy measurements.

The file `q2.train.csv` contains 100 pairs of x and y values, stored in `q2.train$x` and `q2.train$y` respectively. The `q2.train$y` values have been corrupted by adding normally distributed noise to the value of the true function at each `q2.train$x` value. The file `q2.test.csv` contains 1000 different x and y pairs. The values of `q2.test$y` are the values of the true function, without noise, for each of the corresponding `q2.test$x` values. These true vales would be unknown in a real-life situation, but can be used here to see how close your estimated function is to the truth. The samples `q2.train$y` and the value of the true function `q2.test$y` are plotted in Figure 1 against their respective x values.

To answer this question you must:

- Provide an R script containing all the code you used to answer the questions. Please use comments to ensure that the code used to identify each question is clearly identifiable. Call this file `fn.sn.Q2.R`, where "`fn.sn`" is your first name followed by your family name.
- Provide appropriate written answers to the questions, along with any graphs, in a non-handwritten report document.

When answering this question, you must use the `kknn`, `randomForest` and `glmnet` packages that we used in Studio 8 and 9. The wrapper functions for `glmnet` that we used in Studios 8 and 9 are contained in the file `wrappers.R`. You will need to source this before you can use the functions.

Questions

1. Use the k-nearest neighbours method (k-NN) to estimate the underlying function $f(x)$ from the training data. Use the `kknn` package we examined in Studio 9 to provide predictions for the x values in `q2.test` using `q2.train` as the training data. You should use the `kernel = "rectangular"` option when calling the `kknn()` function. This means that the predictions are formed by a simple unweighted average of the k points nearest to the point we are trying to predict.

   (a) Produce four graphs, each one showing: (i) the training data points (`q2.train$y`), (ii) the true function (`q2.test$y`) and (iii) the estimated function (predicted y values for the x values in `q2.test.csv`) produced by the k-NN method for four different values of k; do this for $k = 1$, $k = 5$, $k = 10$ and $k = 25$. Make sure the graphs have clearly labelled axis' and a clear legend. Use a different colour for your estimated curve. [**2 marks**]

   (b) Discuss the four different estimates of the function $f(x)$. How do the curves change as k increases. Which ones do better or worse at capturing the behaviour of the true function (`q2.test$y`), and why? [**2 marks**]

2. Use the cross-validation functionality in the `kknn` package to select the best value of k (make sure you still use the `rectangular` kernel). What value of k does the method select? [**1 mark**]

3. Random forests can also be used to perform function estimation. Use the random forest package to make predictions for `q2.test` using the training data in `q2.train`. [**1 marks**]

4. We can also use linear regression in conjunction with non-linear transformations to try and estimate an underlying function. Fit a 20-th order polynomial (all terms from x up to x^{20}) to the training data, `q2.train`, using the lasso regression method we examined in Studio 8 *(hint: don't forget our target* `q2.train$y` *is continuous, not binary)*. Make sure you use the cross-validation functionality in `glmnet` to select the appropriate lasso regression model. You should then use this model to produce predictions for `q2.test`. Write down the final regression equation found by the lasso method. [**2 marks**]

5. Produce a graph that shows: (i) the training data points, (ii) the true function, (iii) the estimate of the function found by k-NN using the k selected by CV, (iv) the estimate of the function found using random forests, and (v) the estimate of the function found using lasso polynomial regression. How do the three curves compare, and how well do they capture the underlying true function? [**2 marks**].

6. Implement your own version of the leave-one-out cross-validation approach to select the number of neighbours for the k-NN method. Call this function `knn.loo(x.tr, y.tr, k.max)`. This function must take x and y values for training data (`x.tr` and `y.tr`, respectively), and a maximum value of neighbourhood size to try (`k.max`). It should then use the leave-one-out (LOO) cross-validation technique to estimate the cross-validation error for each neighbourhood size from $k = 1$ to $k =$ `k.max`.

   Leave-one-out cross-validation works by leaving out one of the x-y pairs from your training data, using the remaining data and the `kknn()` function (with `kernel="rectangular"`) to make a prediction for the x value of data point that was left out, and then calculating the squared error between the prediction and the y value that was left out. For each candidate value of k, each data point in your training set is left-out one time in this manner, and the prediction errors are added together to get the total cross-validation error for that particular value of k. This is then repeated for each candidate value of k you are trying.

   Your function should return the estimated LOO cross-validation error for each value of k that was tried. To obtain marks for this question you must provide appropriately commented code for the `knn.loo()` function, and your code must work correctly. [**3 marks**]

7. Run your `knn.loo()` function on the `q2.train.csv` data using `k.max` $= 20$. Produce a graph showing the cross-validation error plotted against the values of k that were tested. [**1 marks**].