Midterm2 Exam

Solution Sketches

COSC 4335 *Data Mining*

April 5, 2018

Your Name:

Your student id:

Problem 1 --- R-Functions [6]

Problem 2 --- Basic R-code for Data Frames [6]

Problem 3 --- Visualizing outliers in DBSCAN [11]

Problem 4 --- Tree Models and Classification in General [9]

Problem 5 --- Neural Networks [6]

Problem 6 --- Support Vector Machines [5]

Problem 7 --- DBSCAN [6]

:

**Grade:**



The exam is “open books” and use of computers (but not e-mail) is allowed and you have 75 minutes to complete the exam. The exam will count approx. 14-19% towards the course grade.

**Functions in R [6]**

Write a function tss in R, whose input is vector v of one or more numerical observations, for example:

v <- c(0.3, 0.78, 0.12, 0.35, 0.70)

tss(v)

If the function is called for just one observation, it should return 0; otherwise it should return the value of the following formula:



You are not allowed to call any inbuilt function in R except the mean value function.

tss(v){

if(length(v)<=1)

return(0)

else{

sum<-0;

for(i in v){

sum=sum+(i-mean(v))^2

}

return(sum)

}

}

1 error: -2

More than 1 error: 1 point

**2) Simple Computations with Data Frames [6]**

Suppose you are given a dataframe, df, as follows:

A B

5 2

8 8

3 6

Write R-function called *B-meanA* calculates the mean of column B considering only those rows whose corresponding column A values are larger than 1 but smaller than 8. For df given above B-meanA(df) would return 3=(6+2)/2, as the second observation would be excluded, because it’s A-value is not smaller than 8.

BmeanA(df) {

sum<-0

count<-0

for(i in 1:nrow(df))

{

if((df$A[i]>1) & (df$A[i]<8))

{

sum = sum +df$B[i]

count=count+1

}

}

return(sum/count)

}

1 error: -2

More than 1 error: 1 point

**3) Outlier Visualization [11]**

Write an R-function visoutdbscan(eps, minp) that runs dbscan for the variation of the Complex9 dataset we used in Assignment2, and visualizes the outliers/noisepoints[[1]](#footnote-1) in one color and all other points in the dataset—that is, the points that belong to clusters in a different color—in a different color!

visoutdbscan(eps,minp) {

r<-dbscan(Complex9\_dataset[,1:2], eps, minp)

plot(x = Complex9\_dataset[,1], y = Complex9\_dataset[,2], col = ifelse(r$cluster==0,"red","blue"))

}

1 error: -3

More than 1 error: at most 5 points

**4) Tree Models and Classification in General [9]**

1. Compute the information-gain[[2]](#footnote-2) for the following decision tree split[5] (compute the exact value; just giving the formula will only obtain partial credit)

(1,1,2) (0,0.2)

(1,1,0)

Entropy-before = = 1.5 [2]

Entropy-after = [2]

Information Gain = 1.5-0.5 = 1 [1]

b) What are the characteristics of overfitting? What can be done to deal with overfitting when learning decision tree models? [4]

**Overfitting:** when model is too complex and test errors are **non-optimal** although training errors are small. [1]

To reduce overfitting:

* Pre-Pruning (Early Stopping Rule)-Stop the algorithm before it becomes a fully-grown tree [1]
* Post-pruning-Grow decision tree to its entirety and Trim the nodes of the decision tree in a bottom-up fashion [1]
* Enhance number of training examples [1]

**5) Neural networks [6]**

a) How do neural networks compute the value/activation of a node? [2]

The value of a node is computed by applying the activation function to the weighted sum of its input values [2]

1. How do multi-layer neural networks learn a model for a training set? Limit you answer to at most 5 sentences! [4]

Neural network learning tries to find weights that minimize the error in the neural network prediction for a training set [1]. Neural network learning adjust weights example by example [0.5]; adjusting weights in the direction of the steepest negative gradient of the error function---weights are updated accordingly moving in the direction that reduces the error the most [2]. The step width of the weight update in the direction of the steepest gradient depends on the learning rate and other factors [0.5]. The error in the intermediate layer, as it not initially given, is computed using the back-propagation algorithm [1].

At most 4 points.

**6) SVM [5]**

The soft margin support vector machine solves the following optimization problem:

svn-equation

What does the first term minimize (be precise)? What does ξI  measure? How many examples have non-zero ξi in the figure below! What purpose/role does C play? How many decision boundaries do support vector machines use [5]



The first term minimizes the inverse size of margin between the hyperplanes [1(have not mentioned “inverse” -0.5)]

𝜉 measures “error” [1].

6 examples have non-zero 𝜉 [1].

C helps in determining the importance of minimizing the error in relationship to maximizing the width of margin [1]

One decision boundary [1]

7) DBSCAN [6]

a) What are the characteristics of a border point in DBSCAN [2]?

A border point has fewer than MinPts within Eps but is in the neighborhood of a core point [2].

**b)** Assume you run DBSCAN with MinPoints=6 and epsilon=0.1 for a dataset and I obtain 4 clusters and 2% of the objects in the dataset are classified as outliers/noise points. Now you run DBSCAN with MinPoints=8 and epsilon=0.1. How do expect the clustering results to change? [4]

Some clusters might disappear [1]

Some bigger cluster might be broken up into several smaller clusters [1]

Some clusters might shrink in size

There will be a decrease in the number of core points. [1]

There will be an increase in the number of outliers. [1]

1. Cluster 0 contains all the outliers/noise points. [↑](#footnote-ref-1)
2. Entropy before (computed using the H-function) and after the split. [↑](#footnote-ref-2)