Solution Sketches

Midterm2 Exam

COSC 4335 *Data Mining*

April 7, 2016

Your Name:

Your student id:

Problem 1 --- Computing Entropy using R [11]

Problem 2 --- Tree Models and Classification in General [16]

Problem 3 --- K-NN and SVM [9]

Problem 4 --- DBSCAN [6]

Problem 5 --- Analyzing DBSCAN Clustering Results using R [10]

:

**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.

1. **Computing Entropy using R [11]**

Write a function *H* in R[[1]](#footnote-1), whose input is a vector of class proportions of arbitrary length[[2]](#footnote-2) called v (v contains O and positive numbers whose sum is exactly one) and returns the entropy of for v; e.g.

v<-c(0.5, 0.25, 0.25, 0)

H(v)

*would return:* 0.5\*log2(2) + 2\*1/4\*log2(4) + 0=1.5

Remark: Values of 0 in the input vector do not make any contributions to the overall entropy—their contribution is 0; therefore, make sure when you write the code of the H function that you do not compute 0\*log2(0) as this will return NA[[3]](#footnote-3).

H <- function(v){

H <- 0

for(i in 1:length(v)){

if(v[[i]] != 0){

 H<- v[i]\*log2(1/v[i])+ H}

 }

 return(H)

}

**2) Tree Models and Classification in General [16]**

1. Compute the Gini-gain[[4]](#footnote-4) for the following decision tree split[[5]](#footnote-5)[5] (compute the exact value; just giving the formula will only obtain partial credit)

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

(2, 1, 0)

Gini-Before=1-3\*(1/3)\*\*2=2/3

Gini-After= 2\*0.5\*(1-(1/3)\*\*2-(2/3)\*\*2)=4/9

Gini-gain=2/3-4/9=2/9

b) Compare decision trees ~~support vector machines~~[[6]](#footnote-6), k-nearest neighbor and support vector machine with respect to the shape and number of decision boundaries they employ in their respective classification models? [4]

SVM uses a single decision boundary, Decision trees and k-NN use multiple decision boundaries.[1.5]

SVM uses a hyperplane [0.5], decision trees uses axis-parallel decision boundaries[1; rectangular is also okay), for kNN the decision boundaries are composed of edges of the convex polygons that are part of the Voronoi Tessalation[1].

c) Why is pruning important when using decision trees? What is the difference between pre-pruning and post pruning? [4]

**To come up with a decision tree that uses the correct amount of model complexity to avoid under and overfitting. [1.5] to avoid overfitting is also correct answer!**

**Prepruning: directly prevents a tree from growing too much by using stricter termination conditions in the decision tree induction algorithm [1]**

**Postpruning; Grows a large tree and then reduces it in size by replacing subtrees by leaf nodes[0.5] based on an estimate of the generalization error[1]; e.g. by using a validation set.**

d) How does 2-fold cross validation work? [3]

solution template only:

Subdivide the data into two sets of equal size[1], subdivide …. into training set/test set pairs as follows:… [1] Accuracy is determined…[1]

**3) kNN and SVMs [9]**

a) What are the characteristics of hyperplanes that support vector machines learn from a training set? [3]

The hyperplane separate the examples of the 2 classes, such that the examples of one class are on one side of the hyperplane and the examples of the other class are on the other side of the hyperplane[1.5].

The obtained hyperplane has the widest margin[1]---the empty space that separates the examples of the two classes is maximized! [0.5]

b) Give a sketch how a 3-nearest neighbor classifier determines the class of an example [4].

No solution given!

c) kNN is a lazy classification approach; what are the disadvantages of kNN’s lazy classification approach? [2]

time consuming [1]

as no true model exists, it will be difficult to explain/demo/understand how the model works to a domain expert [1]

4) DBSCAN [6]

a) What is a border point when using DBSCAN? [2]

A point that is not a corepoint [1], but which is within the radius of a corepoint [1]

b) You run DBSCAN with eps=0.5 and minpoints=5 for a dataset and DBSCAN creates 3 clusters and 10% of the objects in the dataset are outliers. Now you run DBSCAN for eps=0.7 and minpoints=5. How, do you expect the clustering result to change? [4]

Usually, there will be more core points as the radius is increased and therefore less outliers/noise points [2]

New clusters might appear and neighboring clusters might be merged into a single cluster[2]; consequently, it is not clear if the number of clusters will increase, decrease, or remain the same.

**5) Analyzing DBSCAN Clustering Results using R [10]**

Write an R-function odbscan(ds,eps,minp) that clusters dataset ds using dbscan with parameters eps and minp and returns the percentage of outliers of the clustering result. Reminder: cluster 0 contains the outliers of a dbscan-clustering; however, if the obtained clustering does not contain any outliers cluster 0 is not created by dbscan, and the function odbscan should return 0 in this case! For example, let us assume you call

odbscan(iris[1:4],0.15,3) and the function returns

[1] 0.8266667

This would indicate the DBSCAN- clustering of the iris flower dataset for eps=0.15 and minp=3 contains 82.7% outliers; that is, 82.7% of the Iris-flower objects belong to cluster 0.

odbscan<-function(ds, eps, minp) {

 res <- dbscan(ds, eps, minp)

 cluster0 <- 0

for (i in 1:length(res$cluster))

 { if (res$cluster[i]==0)

 cluster0 <- cluster0+1

 }

return (cluster0/nrow(ds))}

Or more elegantly without the loop using R vector operations:

sum(res$cluster==0)

1. You will need to write your own code; calling a function in an R-package which computes entropy will not get much credit! [↑](#footnote-ref-1)
2. You can use the length function to determine how many numbers v contains. [↑](#footnote-ref-2)
3. Moreover, in R, log (8,2) computes log2(8). [↑](#footnote-ref-3)
4. Gini before (computed using the Gini-function) and after the split. [↑](#footnote-ref-4)
5. There are 3 classes and 2 examples of each class are in the root node, and then after the split, the left node contains 1 example of class 2 and two examples of class 3, and the right node contains 2 example of class1 and one example of class2. [↑](#footnote-ref-5)
6. This was a typo (it was intended to put decision trees here); consequently, only support vector machines and kNN had to be compared! [↑](#footnote-ref-6)