**April 4, 2017 Review for COSC 4355 Midtem2 Exam**

**1) Hierarchical Clustering**

17. Hierarchical Clustering algorithm creates dendrograms; what is a dendogram? How are the clustering results K-means creates differerent from those of hierarchichal clustering algorithms?

*A* ***dendrogram*** *(from* [*Greek*](https://en.wikipedia.org/wiki/Greek_language) *dendro "tree" and gramma "drawing A dendrogram (from Greek dendro "tree" and gramma "drawing") is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering) is a* [*tree*](https://en.wikipedia.org/wiki/Tree_(graph_theory)) *diagram frequently used to illustrate the arrangement of the clusters produced by* [*hierarchical clustering*](https://en.wikipedia.org/wiki/Hierarchical_clustering)*. Edges of the dendrogram represent split/merge relationships between the nodes of the tree which represent clusters.*

K-means creates a single clustering; hierarchical clustering creates multiple clusterings, namely a set (of nested) clusterings.

**2) Classification**

1. Compute the GINI-gain[[1]](#footnote-1) for the following decision tree split (just giving the formula is fine!)[3]:

(12,4,6) (3,3,0)

(9,1,0)

(0,0,6)

G(6/11,2/11,3/11) – (6/22\*G(0.5,0.5,0) + 10/22\* G(0.9,0,1,0) + 0)=

(1- (6/11)\*\*2-(3/11)\*\*2-(2/11\*\*2)- (6/22\*0.5)- 10/22\*(1-0.9\*\*2-0.1\*\*2)=

(121-36-9-4)/121 - …=

72/121-,,,=

0.595-=

1. Assume there are 5 classes; Compute the entropy of the following class distribution: (1/2,1/4.1/8,1/8,0), giving the exact number not only the formula! [2]

H(1/2,1/4,1/8, 1/8,0)= ½\*log2(2)+ \*1/4log2(4)+ 2\*1/8log2(8)+0=0.5+0.5+6/8=1.75

c) Compare Nearest Neighbor Classifiers and Support Vector Machines! What are the main differences between the two approaches? [6]

|  |  |
| --- | --- |
| Nearest Neighbor Classifiers | Support Vector Machines |
| Consider only the neighborhood to classify🡪local classifer | Usually, Maps objects into higher dimensional space to classify |
| Uses multiple decision boundaries that are edges pf convex polygons that can be computed using Vornoi tessalations | Uses a single, global decision boundary which is a usually high-dimensional |
| Lazy learner, so doesn’t build a model of the training data | Builds the model of the training data, and usually uses a kernel function to map the data in higher dimensions |
| No training cost | Very expensive to learn |
| Distance function is critical; scale of attributes may impact predictions | Kernel function is critical |
| Can handle multiple class output directly | Can handle only two class output. To handle multiple class outputs, several binary classifiers need to be learned to separate instance of class from rest of classes |
| Finds local maxima only | Finds global maxima |

**d)** The following dataset is given (depicted below) with A being a continuous attribute and GINI is used as the evaluation function. What root test would be generated by the decision tree induction algorithm? What is the gain (equation 4.6 page 160 textbook) of the root test you chose? Please justify your answer![6]

Root test: A >=

|  |  |
| --- | --- |
| A | Class |
| 0.22 | 0 |
| 0.22 | 0 |
| 0.31 | 0 |
| 0.33 | 1 |
| 0.33 | 1 |
| 0.41 | 0 |
| 0.41 | 1 |

**Possible slits**

**A≤0.22: (0,2); (3,2)**

**A≤0.31: (0,3); (3,1)**

**A≤0.33: (2,3); (1,1)**

as A≤0.31has a purity of 100%/75% which is much higher than the purity of the other splits, this split will be selected.

e)Most decision tree tools use gain ratio and not GINI or information gain in their decision tree induction algorithm. Why? [3]

Information gain does not consider model complexity in terms of how many additional nodes added to a tree, whereas gain ratio does!

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

Write a function *H* in R[[2]](#footnote-2), whose input is a vector of class proportions of arbitrary length[[3]](#footnote-3) 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[[4]](#footnote-4).

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)

}

**6) 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) 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]

1. (GINI before the split) minus (GINI after the split) [↑](#footnote-ref-1)
2. 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-2)
3. You can use the length function to determine how many numbers v contains. [↑](#footnote-ref-3)
4. Moreover, in R, log (8,2) computes log2(8). [↑](#footnote-ref-4)