Homework 2

Jarod Klion

February 16th, 2022

**Chapter 3:**

1. **Identify at least two advantages and two disadvantages of using color to visually represent information.**
   1. *Advantages:* Color can improve readability of information by making elements more visually distinguishable. Also, colors are nicer to look at.

*Disadvantages:* Colorblind people can have problems reading a color figure. Additionally, colors can be difficult to use properly as a poor color scheme can cause more readability problems than a grayscale image would have.

1. **What are the arrangement issues that arise with respect to three­-dimensional plots?**
   1. We must arrange the three-dimensional plots to show as much information as possible while still portraying all the necessary information.
2. **Describe one advantage and one disadvantage of a stem and leaf plot with respect to a standard histogram.**
   1. A stem and leaf plot shows all the actual distributions of values. However, this becomes problematic when there are a large number of values.
3. **How might you address the problem that a histogram depends on the number and location of the bins?**
   1. Since a histogram shows the distribution of values of a single variable anyway, the best approach to address this problem would be to approximate the actual distribution function of the data.
4. **Comment on the use of a box plot to explore a data set with four attributes: age, weight, height, and income.**
   1. A box plot of each attribute can show the distributions of values or how one attribute varies based on different classes of objects, such as how income increases with age if we look at box plots of different age categories.
5. **Describe the types of situations that produce sparse or dense data cubes. Illustrate with examples other than those used in the book.**
   1. A sparse data cube would happen with data where many combinations of values are unlikely to occur. For example, discrete attributes where many combinations of values don’t occur would be in this group
   2. A dense data cube, on the other hand, would occur when almost all combinations of values occur or for high aggregation that makes all combinations likely to have values. For example, consider a data set of flower species in addition to its precise native location and color. This would be a somewhat sparse data cube, but if we aggregate it to have categories of primary color and the country of origin, then we would obtain a dense data cube.

**Chapter 4:**

1. **Table

   Description automatically generatedConsider the following data set for a binary class problem.**
   1. **Calculate the information gain when splitting on A and B. Which attribute would the decision tree induction algorithm choose?**

|  |  |  |
| --- | --- | --- |
|  | A = T | A = F |
| + | 4 | 0 |
| - | 3 | 3 |

|  |  |  |
| --- | --- | --- |
|  | B = T | B = F |
| + | 3 | 1 |
| - | 1 | 5 |

* Entropy(p) =
* Information gain after splitting on A:
* Information gain after splitting on B:
* Attribute A would be chosen since it maximizes GAIN
  1. **Calculate the gain in the Gini index when splitting on A and B. Which attribute would the decision tree induction algorithm choose?**
* Gain in GINI index after splitting on A:
* Gain in GINI index after splitting on B:
* Attribute B would be chosen since it maximizes gain in GINI index
  1. **Figure 4.13 shows that entropy and the Gini index are both monotonously increasing on the range [0, 0.5] and they are both monotonously decreasing on the range [0.5, 1]. Is it possible that information gain and the gain in the Gini index favor different attributes? Explain.**
* Yes, it is possible that information gain and the gain in Gini index favor different attributes regardless of their similar range and behaviors. This can be seen by information gain favoring splitting on A while the gain in Gini index favors splitting on B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Number of Instances | |
| A | B | C | + | - |
| T | T | T | 5 | 0 |
| F | T | T | 0 | 20 |
| T | F | T | 20 | 0 |
| F | F | T | 0 | 5 |
| T | T | F | 0 | 0 |
| F | T | F | 25 | 0 |
| T | F | F | 0 | 0 |
| F | F | F | 0 | 25 |

1. **The following table summarizes a data set with three attributes A, B, C and two class labels +, −. Build a two-­level decision tree.**
   1. **According to the classification error rate, which attribute would be chosen as the first splitting attribute? For each attribute, show the contingency table and the gains in classification error rate.**

      * Gain in error rate after splitting on A:

|  |  |  |
| --- | --- | --- |
|  | A = T | A = F |
| + | 25 | 25 |
| - | 0 | 50 |

|  |  |  |
| --- | --- | --- |
|  | B = T | B = F |
| + | 30 | 20 |
| - | 20 | 30 |

* + - Gain in error rate after splitting on B:
* Gain in error rate after splitting on C:

|  |  |  |
| --- | --- | --- |
|  | C = T | C = F |
| + | 25 | 25 |
| - | 25 | 25 |

* Attribute A has the highest classification error rate gain, so it would be chosen as the first splitting attribute
  1. **Repeat for the two children of the root node.**
     + A = T node is already pure, so only need to split A = F node

|  |  |  |  |
| --- | --- | --- | --- |
| B | C | + | - |
| T | T | 0 | 20 |
| F | T | 0 | 5 |
| T | F | 25 | 0 |
| F | F | 0 | 25 |

* Gain in error rate after splitting on B:

|  |  |  |
| --- | --- | --- |
|  | B = T | B = F |
| + | 25 | 0 |
| - | 20 | 30 |

* Gain in error rate after splitting on C:

|  |  |  |
| --- | --- | --- |
|  | C = T | C = F |
| + | 0 | 25 |
| - | 25 | 25 |

* Attribute B has the highest classification error rate gain, so it would be chosen as the first splitting attribute
  1. **How many instances are misclassified by the resulting decision tree?**
     + 20 instances
  2. **Repeat parts (a), (b), and (c) using C as the splitting attribute.**

**(d.b)**

* + - For the C = T node:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | + | - |
| T | T | 5 | 0 |
| F | T | 0 | 20 |
| T | F | 20 | 0 |
| F | F | 0 | 5 |

* + - * Gain in error rate after splitting on A:

|  |  |  |
| --- | --- | --- |
|  | A = T | A = F |
| + | 25 | 0 |
| - | 0 | 25 |

* + - * Gain in error rate after splitting on B:

|  |  |  |
| --- | --- | --- |
|  | B = T | B = F |
| + | 5 | 20 |
| - | 20 | 5 |

* + Attribute A has the highest classification error rate gain, so it would be chosen as the splitting attribute.
* For the C = F node:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | + | - |
| T | T | 0 | 0 |
| F | T | 25 | 0 |
| T | F | 0 | 0 |
| F | F | 0 | 25 |

* + - * Gain in error rate after splitting on A:

|  |  |  |
| --- | --- | --- |
|  | A = T | A = F |
| + | 0 | 25 |
| - | 0 | 25 |

* + - * Gain in error rate after splitting on B:

|  |  |  |
| --- | --- | --- |
|  | B = T | B = F |
| + | 25 | 0 |
| - | 0 | 25 |

* Attribute B has the highest classification error rate gain, so it would be chosen as the splitting attribute.

**(d.c)**

* 0 instances are misclassified by the resulting decision tree
  1. **Use the results in parts (c) and (d) to conclude about the greedy nature of the decision tree induction algorithm.** 
     + The greedy nature of the decision tree induction algorithm doesn’t necessarily pick the best tree.

1. Diagram

   Description automatically generated
2. Consider the decision trees shown in Figure 4.3. Assume they are generated from a data set that contains 16 binary attributes and 3 classes, C1, C2, and C3. Compute the total description length of each decision tree according to the minimum description length principle.

* The total description length of a tree is given by:

*Cost*(*tree, data*) = *Cost*(*tree*) *+ Cost*(*data|tree*)

* Each internal node of the tree is encoded by the ID of the splitting attribute. If there are m attributes, the cost of encoding each attribute is *log*2(*m*) bits.
* Each leaf is encoded using the ID of the class it is associated with. If there are k classes, the cost of encoding a class is *log*2(*k*) bits.
* *Cost*(*tree*) is the cost of encoding all the nodes in the tree. To simplify the computation, you can assume that the total cost of the tree is obtained by adding up the costs of encoding each internal node and each leaf node.
* *Cost*(*data|tree*) is encoded using the classification errors the tree commits on the training set. Each error is encoded by *log*2(*n*) bits, where *n* is the total number of training instances.
  + - **Which decision tree is better, according to the MDL principle?**
      * Cost for each internal node in the tree:
      * Cost of encoding a class:

1. Cost(tree) = (2 nodes) \* 4 + (3 leaves) \* 1.585 = 12.755

Cost(data|tree) = (7 errors) \* log2(n) = 7log2(n)

Cost(tree, data) = 12.755 + 7log2(n)

1. Cost(tree) = (4 nodes) \* 4 + (5 leaves) \* 1.585 = 23.925

Cost(data|tree) = (4 errors) \* log2(n) = 4log2(n)

Cost(tree, data) = 23.925 + 4log2(n)

* + - * Decision tree (a) is better than (b) according to the MDL principle for n < 14, but (b) is better than (a) for n > 14.