CS578/STAT590: Introduction to Machine Learning **Fall 2014**

Problem Set 2

Gen Nishida *Handed In: September 30, 2014*

1 Questions

1. (1) Boolean function

   By using conjunctions for every combination of three variables, we can test if at least three variables are active. Therefore, the Boolean function for this is as follows:

   $$f = (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_4) \vee (x_1 \wedge x_2 \wedge x_5) \vee (x_1 \wedge x_3 \wedge x_4) \vee (x_1 \wedge x_3 \wedge x_5) \\ \vee (x_1 \wedge x_4 \wedge x_5) \vee (x_2 \wedge x_3 \wedge x_4) \vee (x_2 \wedge x_3 \wedge x_5) \vee (x_2 \wedge x_4 \wedge x_5) \vee (x_3 \wedge x_4 \wedge x_5)$$

   (2) Linear function

   Linear function that can achieve the same classification just needs to checke if the sum is more than two.

   $$f = \begin{cases} 1 & (x_1 + x_2 + x_3 + x_4 + x_5 \geq 3) \\ 0 & Otherwise \end{cases}$$

2. What is the size of CON_B?

   For every variable, there are two cases, used in the conjunctions or not used. Thus, the size of CON_B is 2^n.

3. What is the size of CON_L?

   For every pair of $f_b^{(i)}$ and $f_b^{(j)}$ $(i \neq j)$ in CON_B, $\exists x, f_b^{(i)} \neq f_b^{(j)}$. Thus, ths size of CON_L has to be larger than the size of CON_B to make it consistent with CON_B. However,

4. Mistake bound

   Mistake bound is the maximum possible number of mistakes made by the online learning algorithms, which is also used to evaluate the performance of the convergence of the algorithms.

5. mistake bound algorithm

   1) initialize : $w_i = 1$
   2) if no mistake do nothing
   3) else
   4) for all in-active variables, remove x_i from the conjunctions

   For every mistake, we remove at least one unnecessary variable from the conjunctions. Since we have at least one variable in the conjunctions, the total number of mistakes is at most $n - 1$, which is a polynomial in n. Thus, this is a mistake bound algorithm.

6. (1) Will both classifiers converge?

   (2) What will be the training error of each one of the classifiers?

7. kernel function $K(x, y)$

2 Programming Assignment

1. Define in one sentence: (1) training set, (2) test set, (3) validation set.

   - training set

     Training set is a set of data used to optimize a hypothesis function.
   - test set

     Test set is a set of data used to measure the accuracy of the hypothesis generated through training and validation phases.
   - validation set

     Validation set is a set of data used to estimate the performance of the hypothesis and to generalize the hypothesis preventing from overfitting.

2. Can you use the validation set as a test set?

   Technically yes, but practically no, because it is meaningless. Since validation set is used to estimate the accuracy of the hypothesis during the training phase, the resulting hypothesis is optimized for the validation set, and it is meaningless to use the validation set as a test set in order to measure the actual performance of the hypothesis.

3. Define in one sentence: overfitting

   A hypothesis is said to overfit the training data if it has smaller error on the training data but loses the generalization performance and has larger error on test data.

4. True or False (and why): A learned hypothesis f has a training error e_{tr} and a testing error e_{ts}, where $e_{tr} > e_{ts}$.

   - (1) can we say that f overfits to the training data?

     False. Since the hypothesis f is optimized for the training data while the test data is unknown during training phase, the training error e_{tr} is smaller than e_{ts} in general, even if f is well generalized. In this case, $e_{tr} > e_{ts}$, which indicates that e_{ts} is very small and f is generalized very well.
   - (2) Now, assume that $e_{tr} < e_{ts}$, does f overfit to the training data?

     False. Since the hypothesis f is optimized for the training data while the test data is unknown during training phase, the training error e_{tr} is smaller than e_{ts} in general, even if f is well generalized. The optimal hypothesis minimizes e_{ts}, but still it may be the case that $e_{tr} < e_{ts}$. Therefore, we cannot conclude that f overfits to the training data even if $e_{tr} < e_{ts}$, unless we find another hypothesis f' which has larger error on the training data but smaller error on the test data compared to f.

3 Decision Trees

1. The "Thrill and Romance" bookstore

   - What is the entropy of the target variable? (Buy)
     The number of examples labeled "Buy=Y" is 7, while the number of examples labeled "Buy=N" is 4. Thus,

     $$-\frac{7}{11}\log\frac{7}{11}-\frac{4}{11}\log\frac{4}{11}=0.94566$$

   - What are the attributes considered by the algorithm?
     All the attributes, "Pages", "Famous Author", "Category", and "Cover Color" should be considered by the algorithm. For "Pages", since it is a continuous attribute, we first have to sort examples according to the values of "Pages" and check the mid-point as a possible threshold in order to discretize. The sorted values are as follows:
     45(-), 50(+), 72(+), 100(-), 120(+), 142(+), 150(+), 200(-), 300(+), 350(+), 1000(-).
     Thus, the possible thresholds for "Pages" are 47.5, 86, 110, 175, 250, and 675. These thresholds are called cut points [1], and used to split the continuous space into two ranges so that the continuous values can be handled in the same manner with the discrete values. Note that the thresholds have to be calculated for every node, since the set of the continuous values for each node may differ from the root node.
   - What is the first attribute that the algorithm will split the data on? What is its information gain?
     The information gain by the split of each attribute is computed as follows.
     - Pages(threshold=47.5)

       $$0.94566-\left[0\times\frac{1}{11}-\left(-\frac{7}{10}\log\frac{7}{10}-\frac{3}{10}\log\frac{3}{10}\right)\times\frac{10}{11}\right]$$
       $$=0.94566-0.80117=0.14449$$

     - Pages(threshold=86)

       $$0.94566-\left[\left(-\frac{2}{3}\log\frac{2}{3}-\frac{1}{3}\log\frac{1}{3}\right)\times\frac{3}{11}+\left(-\frac{5}{8}\log\frac{5}{8}-\frac{3}{8}\log\frac{3}{8}\right)\times\frac{8}{11}\right]$$
       $$=0.94566-0.94458=0.00108$$

     - Pages(threshold=110)

       $$0.94566-\left[1.0\times\frac{4}{11}-\left(-\frac{5}{7}\log\frac{5}{7}-\frac{2}{7}\log\frac{2}{7}\right)\times\frac{7}{11}\right]$$
       $$=0.94566-0.91289=0.03277$$

- Pages(threshold=175)
  Same as threshold=110, which is 0.03277.
- Pages(threshold=250)
  Same as threshold=86, which is 0.00108.
- Pages(threshold=675)
  Same as threshold=47.5, which is 0.14449.
- Famous Author

$$0.94566 - \left[\left(-\frac{5}{7}\log\frac{5}{7} - \frac{2}{7}\log\frac{2}{7}\right) \times \frac{7}{11} + 1.0 \times \frac{4}{11}\right] = 0.94566 - 0.91289 = 0.03277$$

- Category

$$0.94566 - \left[\left(-\frac{4}{5}\log\frac{4}{5} - \frac{1}{5}\log\frac{1}{5}\right) \times \frac{5}{11} + 1.0 \times \frac{6}{11}\right] = 0.94566 - 0.8736 = 0.07206$$

- Cover Color

$$0.94566 - \left[\left(-\frac{6}{9}\log\frac{6}{9} - \frac{3}{9}\log\frac{3}{9}\right) \times \frac{9}{11} + 1.0 \times \frac{2}{11}\right] = 0.94566 - 0.93315 = 0.01251$$

The attribute which achieves the highest information gain is "Pages" with 47.5 or 675 as the threshold, so we should use "Pages" as the first attribute to split, and the information gain is 0.14449. Note that if we split all the values of "Pages", then we will be able to completely separate the labels with the weighted averaged of the entropy = 0.0. However, this will be overfitting, and we do not want to do that.

- Due to a computer error some of the training examples attributes were deleted! Revise the decision tree training algorithm to deal with missing values in the training data.

  The goal of training is to minimize the expected loss over the distribution of values of attributes. Intuitively, it is better to use this distribution to estimate the missing values, but, we do not know the distribution in advance. One option would be to choose the majority votes for the missing values. Suppose the first row in our training data does not have a value of "Famous Author". If we use "Famous Author" as the first attribute to split the tree, then we assign "Y" for the first row, becuase "Y" is the majority. Although this options is easy to implement, the minority values may be ignored during the decision tree building.

  Another option is to use the probability distribution according to the training data. If we have enough number of training data without missing values, the distribution of the values in the training data is likely to be similar to the one of testing data. Hence, it is reasonable to use the distribution of the values in the

training data to estimate the missing values. In the same example above, in which the first row of the training data has missing value for "Famous Author", we get "Y" with probability 0.6 and "N" with probability 0.4. Based on this, the first row contributes 0.6 to the sub-tree for "Y", while it contirbutes 0.4 to the other sub-tree for "N". In the decision tree implementation in the following section, I used the latter approach.

2. Decision Tree Implementation

I implemented the decision tree algorithm using the second option for the missing values. With regard to the continuous values, for every node, I computed the expected information gain for each cut point discussed above in addition to the discrete attributes, and choose the best one which achieves the highest information gain. Also, the validation data was used to find the best maxDepth, changing from 0 to 20. Finally, the performance of the resulting decision tree was measured on the test data. Please refer to the attached source code for details.

Table 1: Results

maxDepth	Validation data	Test data
0	0.58461	0.50746
1	0.96923	0.52238
2	0.96923	0.52238
3	0.96923	0.52238
4	0.90769	0.56716
5	0.92307	0.56716
6	0.92307	0.56716
7	0.92307	0.56716
8	0.92307	0.56716
9	0.92307	0.56716

The results of my decision tree on the validation and testing data are shown in Table 1. Since the testing data is not available beforehand, we have to choose the best maxDepth only based on the results on the validation data. Thus, we choose $maxDepth = 1$, and we get 0.52238 as the accuracy on the test data. Note that this result is better than the case of $maxDepth = 0$, which is a baseline, but there is significant difference in the accuracy between the results on validation and training data.

This poor performance is caused by the different distribution of values in the feature space between the training data and the test data. To investigate more details, I computed the distribution of labels in the child nodes of the root node as shown in Figure 1. The results indicate that the distribution of the values of 9th attribute against the labels is significantly different between the training data and the test data. In the training data, most data with the value "t" for the 9th attribute have positive labels, while most data with the value "f" for the 9th attribute have negative labels. This polarization leads to low entropy, i.e. high information gain. In the test data, on

the other hand, both subsets have both labels almost evenly, which results in very high entroy, i.e. low information gain. Hence, the decision tree trained from the training data could not get high accuracy on the test data.

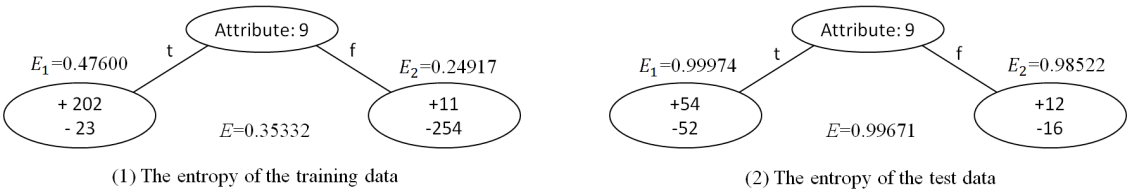

Figure 1: Comparison of the entropy between the training data and the test data: The splitting by the 9th attribute on the training data achieves very low entroy, whose weighted average is 0.35332. This is why the 9th attribute was selected as the root node for the decision treee. On the other hand, the splitting by the same attribute on the test data results in very high entroy, whose weighted average is 0.99671. This indicates that the distribution of the 9th values against the labels are significantly different between the training data and the test data, and this causes very poor performance of the decision tree on the test data.

References

[1] Fayyad, Usama M. and Irani, Keki B. 1992. On the Handling of Continuous-Valued Attributes in Decision Tree Generation. *Machine Learning*, 8, pp. 87 - 102.