DAMI 2019

Homework Assignment 3

Supervised Learning – Classification

These instructions should be used together with the provided assignment R code ("*HW3_Rcodes.R*"). To complete this assignment, you can choose either to write code on your own or to use the provided R code and only fill in the missing parts. Download and use the synthetic datasets "*entropy_data.RData*" for task 1 and "*Perceptron_data.RData*" for task 2. **ONLY** submit the complete R code (.R file). Screenshots, plots, word or pdf documents **WILL NOT** be graded.

Part 1: Simulating a node splitting process in a Decision Tree.

A: Entropy calculation

Entropy is a measure of impurity. It is defined as

$$I(D) = -\sum_{i=1}^{m} p_i * log_2(p_i)$$

Where *m* is the number of classes, *p* is the probability of each class and *D* is the distribution of class labels. For example, if the class label is binary and 5 of the training examples belong to class "YES" while 7 of them belong to class "NO", then the entropy of this class distribution is

$$I(5,7) = -p_1 log_2 p_1 - p_2 log_2 p_2 = -\frac{5}{5+7} log_2\left(\frac{5}{5+7}\right) - \frac{7}{5+7} log_2\left(\frac{7}{5+7}\right)$$

[YOUR TASK] *Write a function that can calculate entropy for a given class frequency distribution. Use the provided dataset ("entropy_data.RData") for this task to evaluate your function. Notice that the class label is "buys" in this dataset. This function will be used in later part of the assignment to calculate entropy of a node in a decision tree while making a split decision. Use the class frequency distribution of these observations. The correct answer should be 0.940286. [5 Points]*

B: Information gain

Information gain of a node split in a decision tree on a certain attribute A is a difference in entropy before split and after split. Entropy after split can be obtained as

$$I(D, A) = \sum_{j=1}^{v} \frac{n_j}{n} * I(D_j)$$

Where v is the number of categories of attribute *A*, n_j is the number of observations with category *j* in attribute *A*, *n* is total number of overall observations and D_j is the distribution of class label among observations with category *j* in attribute *A*.

Information gain obtained when a tree node is split using attribute *A* is then

$$Gain(D, A) = I(D) - I(D, A)$$

[YOUR TASK] *Fill in the R code to write a function that compute information gain while splitting a tree node on certain attribute. Use the provided dataset ("entropy_data.RData") as in the first task. Find information gain by splitting all the observations (i.e. root node) on attributes "income", "student" and "credit". Correct answers would be 0.02922257, 0.1518355 and 0.04812703 respectively. [5 Points]*

C: Information gain for numeric attributes

As you have already noticed above, attribute A used to split a node have categories or it is a categorical attribute. What if A is numeric? Here is one solution for computing information gain for numeric attribute.

Algorithm

1. Sort the values in numeric attribute *A*, remove duplicates and store it as *A1*.
2. If *A1* has *K* elements, for *k* from *1* to *K-1*, find midpoints m_k between k^{th} and $(k+1)^{th}$ element .
3. Discretize original numeric attribute *A* using midpoint m_k. If an observation has a value for *A* less than m_k, assign the value to *A "YES"* else *"NO"*. So eventually, numeric attribute *A* is changed to categorical with categories "*YES"* and *"NO"*. Now, compute information gain for discretized attribute *A*.
4. Iterate the discretization process using all the midpoints m_k and compute corresponding information gain.
5. Select the midpoint used for discretization that delivers maximum information gain among all midpoints.
6. The selected midpoint is then saved as a cutoff point for the numeric attribute A and information gain obtained as delivered by the attribute.

[YOUR TASK] *Fill in the R code to write a function where numeric attribute is discretized to make it ready for computing information gain. This function also iterates over all attributes, both numeric and categorical and computes information gain obtained for each one of them. Again use the "entropy_data.RData" to test the function. Information gain for numeric attribute "age" should be 0.10224356 at best cutoff point 31. Based on the result, which attribute will be used to split the node. [5 Points]*

Part 2: Perceptron Learning Algorithm

Perceptron is a simple supervised learning algorithm which makes its predictions based on a linear predictor function combining a set of weights with the feature vector. Let's say we have input $X = (x_1, x_2, \dots, x_d)$, where d is the number of features. Given the weight vector $w = (w_0, w_1, \dots, w_d)$, the hypothesis is

$$h(X) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + w_0\right)$$

In order to take into account w_0, we introduce an artificial coordinate $x_0 = 1$ for all observations.

$$h(X) = sign\left(\sum_{i=1}^{d} w_i x_i\right)$$

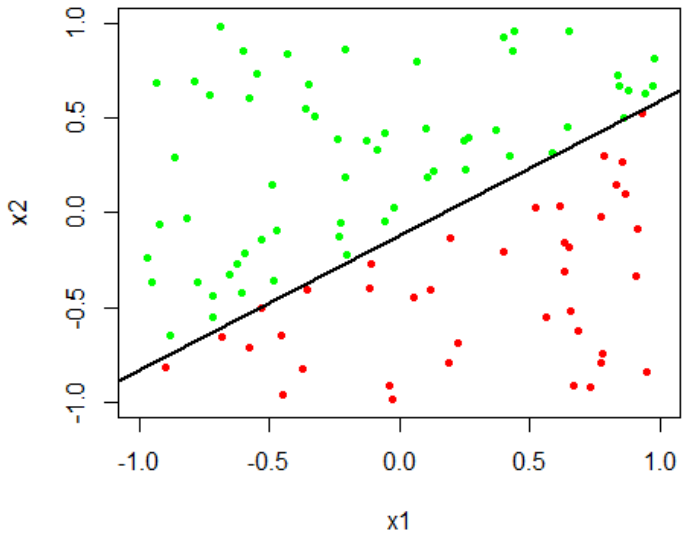
In vector form, the perceptron implements $h(X) = sign(w^T X)$. Given the training data $(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)$, where N is the number of examples or points in a space, pick a misclassified point: $sign(w^T X_n) \neq y_n$ and update the weight vector by $w \leftarrow w + yX$, where (X, y) is a misclassified training point. Iterate this implementation until no point is misclassified.

[YOUR TASK] *Write a function that implements perceptron learning algorithm. Use the synthetic data ("Perceptron_data.RData") for this task.*

A: Calculate the hypothesis, given the training data X and weight vector w by using matrix product (vector product). Hint: For the given synthetic data, hypothesis should a matrix of dimension 1X100. [6 Points]

B: Update weight vector by adding yX to the current weight vector where (X,y) is a misclassified training point. [6 Points]

After the algorithm has converged, the final weight vector should be [1, -6.319699, 9.025192] and the plot for the splitting hyperplane should look similar to the figure below.



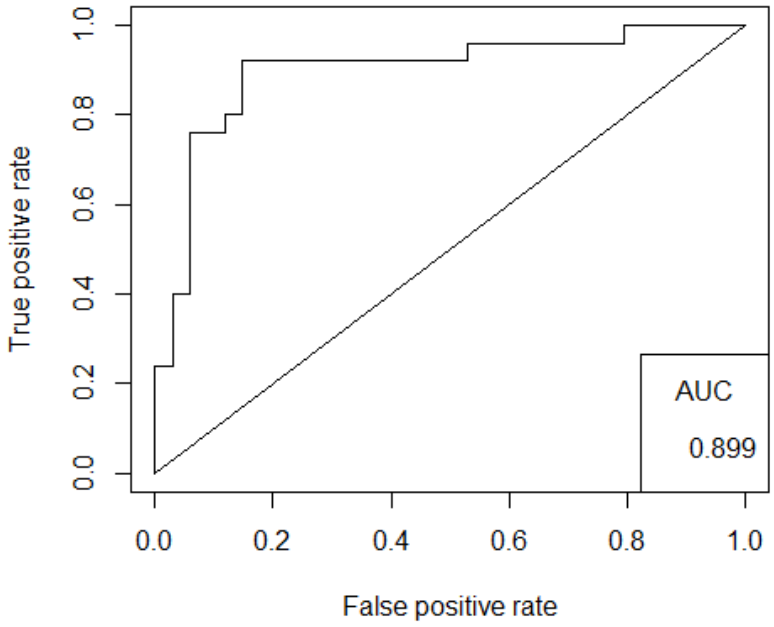
Part 3: Cross Validation, SVM and parameter tuning

Download "Heart Disease" dataset from UCI Machine Learning Repository and load in your R environment.

URL https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data

A [YOUR TASK] *Train a SVM model on the dataset. Use a radial basis kernel. Find the best value for cost parameter from among 0.1, 1, 10 and 100. Perform 10 fold cross validation for each cost to compute average CV accuracy across ten folds. Compare average CV accuracy for different cost values and determine the value for the cost that gives best accuracy. [8 Points]*

B [YOUR TASK] *Using the best value for cost parameter discovered in task 3A, train a SVM model with radial basis kernel on 80 percent of the data randomly fetched for training purpose. Predict probabilities on test examples and plot a ROC curve for a positive class (i.e. 1 in this case) and compute AUC. The plot should look similar to the one shown below. [5 Points]*


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC
0.899
False positive rate