

CS 559: Machine Learning

Homework Assignment 3

Due Date: Tuesday 6:30 PM, November 15, 2022

Total: 100 points

Problem 1. Decision Trees based on Entropy (20 points)

Consider the following dataset for a binary classification problem and answer the following questions.

Instance	Attribute 1 (a_1)	Attribute 2 (a_2)	Class
1	T	1.0	+
2	T	6.0	+
3	T	5.0	-
4	F	4.0	+
5	F	7.0	-
6	F	3.0	-
7	F	8.0	-
8	T	7.0	+
9	F	5.0	-

- (1) Among the two attributes, show which attribute will be chosen as the first splitting for decision tree using **information gain (gain in the entropy)**. Show all split points for all attributes. For the continuous attribute a_2 , please perform the binary decision by considering each distinct value as the splitting threshold and find the best one.

Please write down every step including the calculation of entropy of the parent, count matrix after splitting, entropy of the children after splitting, and the information gain of the attributes at each split.

- (2) What happens if we use “Instance” as another attribute? Do you think this attribute should be used for a decision in the tree?

Problem 2. Decision Trees based on GINI index (20 points)

The following table summarizes a dataset with two attributes A, B, and two class labels +, -. The original dataset had 5 attributes, but only 2 of the most informative attributes are selected for further analysis.

A	B	Class Label	
		+	-
T	T	0	20
T	F	20	10
F	T	15	0
F	F	0	35

Cost Matrix	Attribute Value		
Actual Class		T	F
	+	-1	100
	-	0	-10

- (1) According to **GINI index**, which attribute would be chosen as the first splitting attribute?
- (2) Use the given cost matrix and decide the first splitting attribute. The total cost is the metric for splitting. Note that this question is independent to question (1). You will need to use both tables provided above to calculate the cost of splitting A and B, and choose the attribute with smaller cost to split.

Please write down every step including the calculation of the count matrix after splitting, GINI index of the parent, GINI index of the children after splitting, and the cost of splitting based on each attribute.

Problem 3. AdaBoost (20 points)

Let us consider the AdaBoost algorithm in which all the data points are initially given the uniform weights. Given below is a dataset with feature X and response Y.

ID	1	2	3	4	5	6	7	8	9	10
X	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Y	1	1	1	-1	-1	-1	-1	-1	1	1

We will now consider three weak classifiers (Hypotheses) H1, H2, and H3.

$$H1: \text{if } X \leq 0.35 \rightarrow Y = 1, \text{ else } Y = -1$$

$$H2: \text{if } X \leq 0.75 \rightarrow Y = -1, \text{ else } Y = 1$$

$$H3: \text{if } X \leq 0.3 \text{ or } X \geq 0.95 \rightarrow Y = 1, \text{ else } Y = -1$$

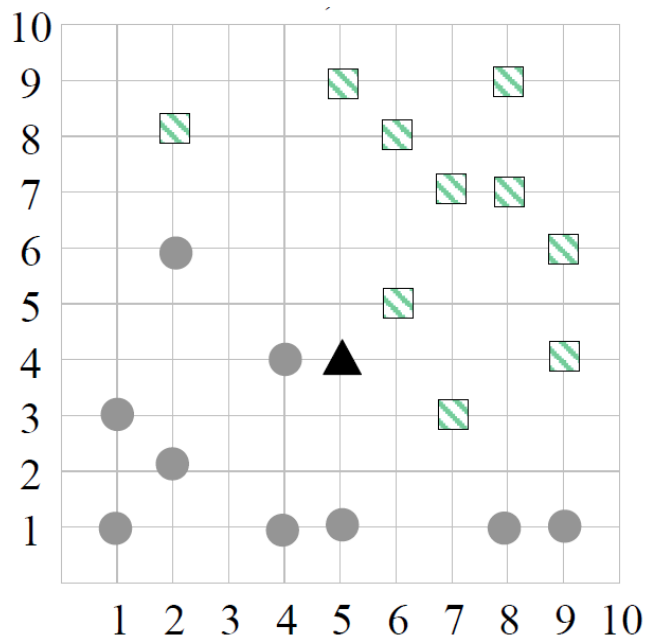
For each of the three weak classifiers, answer the following questions. Note that each weak classifier is independent and will start with the initial weight.

- (1) Compute the weights of all the instances after the first round of the AdaBoost algorithm. No need to perform normalization of the weights at the end of the iteration.
- (2) Clearly specify the data instances which will be reweighted after the first iteration. The answer can be 'None of them', 'All of them', or you can list the specific data instances.

Problem 4. K-nearest Neighbor Classification (40 points; 10+30)

Part I. Consider the following 2-dimensional dataset (Treat the squares as + and circles as -). Classify the test point (triangle) using the following two strategies:

- (1) 5- nearest neighbor
- (2) **Manhattan distance** weighted 3-nearest neighbor (the weight is $1/d^2$)



Part II. Consider the three-dimensional data set in **train.csv**.

- (1) Classify the data points in **test.csv** according to their 3-nearest neighbors. Also give the probability estimates for the final decision.
- (2) Do the same for the **Euclidean distance** weighted 3-nearest neighbors ($1/d^2$). Do the predicted label for each point remain the same as that in question (1)?
- (3) In the test.csv, the true class labels are also provided. Construct the confusion matrix and calculate Accuracy, Precision, F-measure for questions (1) and (2). From your results, which method gives better performance?

You can implement by yourself or use the sklearn packages to implement KNN. Please submit your source code, outputs, and the details of the calculation in (3) for full credit.