ENGR-UH 4560
Selected Topics in Information and Computational Systems

Mini Project - K-NN and Decision Tree

Due Date: Refer to NYU Class

Introduction

-KNN

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

-Decision Tree

In computer science, decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). This page deals with decision trees in data mining.

Dataset

Classification Trees with Numerical Features (two data sets)

- **Iris**: has three classes and the task is to accurately predict one of the three subtypes of the Iris flower given four different physical features. These features include the length and width of the sepals and the petals. There are a total of 150 instances with each class having 50 instances.
- **Spambase**: is a binary classification task and the objective is to classify email messages as being spam or not. To this end the dataset uses for seven text-based features to represent each email message. There are about 4600 instances.

Since both datasets have continuous features you will implement decision trees that have binary splits. For determining the optimal threshold for splitting you will need to search over all possible thresholds for a given feature (refer to class notes and discussion for an efficient search strategy). Use information gain to measure node impurity in your implementation.

Requirements

-Task1: KNN

1. In your algorithm, you will only use two of these features: *sepal length and petal width.* (We are only using these two features since this is an assignment for you to experiment with different values of k and different distance measurements. With all 4 features, it is easy to classify the Iris perfectly; try this when you have completed your assignment.) Implement the k-Nearest Neighbor algorithm to classify the test examples, using a Euclidean distance function. Run your algorithm with k = 1 , k = 3 and with k = 5. For k > 1, sort the training examples by distance from the test example, smallest to largest, to find the k nearest training examples. Answer the following questions:

2. For k = 1, which examples were not correctly classified?

3. Report the accuracy on the test set for k = 1.

4. For k = 3, which examples were not correctly classified?

5. Report the accuracy on the test set for k = 3.

6. For k = 5, which examples were not correctly classified?

7. Report the accuracy on the test set for k = 5.

8. That is, we classify all examples in the test set as belonging to the class that is most common in the training set. What is the resulting accuracy?

9. Choose a new distance function for your k-NN algorithm. You can choose whatever distance function you like but should choose something that you think might yield higher accuracy than the Euclidean distance function.

10. Run k-NN with your distance function, using the same training and test sets, to classify the examples in the test set. Did your distance function achieve higher accuracy (for k = 1, k = 3, and k = 5) than the first distance function? If it didn't, what is a possible reason that it didn't?

-Task2: Decision Tree

1. **Growing Decision Trees**

   Instead of growing full trees, you will use an early stopping strategy. To this end, we will impose a limit on the minimum number of instances at a leaf node, let this threshold be denoted as n_{min}, where n_{min} is described as a percentage relative to the size of the training dataset. For example, if the size of the training dataset is 150 and n_{min}= 5, then a node will only be split further if it has more than eight instances.

- For the Iris dataset use n_{min} E {5, 10, 15, 20}, and calculate the accuracy using 10 fold cross-validation for each value of min.
- For the Spambase dataset use n_{min} E {5, 10, 15, 20, 25}, and calculate the accuracy using 10 fold cross-validation for each value of n_{min}.

You can summarize your results in two separate tables, one for each dataset (report the average accuracy and standard deviation across the folds).

Deliverables

A .ipynb file containing the following:

1. source code
2. detailed description of the project
3. answers to the programming questions.

Before submitting your project, please make sure to test your program on the given dataset.

Notes

You may discuss the general concepts in this project with other students, but you must implement the program on your own. ***No sharing of code or report is allowed.*** *Violation of this policy can result in a grade penalty.*

Late submission is acceptable with the following penalty policy:
10 points deduction for every day after the deadline