ENGINEERING DIVISION | NYU ABU DHABI

ENGR-UH 4560

Selected Topics in Information and Computational Systems

Machine Learning

Project 04 - Decision tree

Introduction - 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.

Requirements

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.

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 ten 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 ten 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).

Interpreting the results

Select the best value of min for the Iris dataset, and create a class confusion matrix using ten-fold cross-validation (use only the test set for populating the confusion matrix). How do you interpret the confusion matrix, and why?

Regression Trees

In this problem you will implement regression trees using a new dataset:

- **Housing**: This is a regression dataset where the task is to predict the value of houses in the suburbs of Boston based on thirteen features that describe different aspects that are relevant to determining the value of a house, such as the number of rooms, levels of pollution in the area, etc.
    - As this dataset has only numerical features we will be growing decision trees using only binary splits. Use **mean squared error (MSE)** to define the splits. Use an early stopping strategy similar to the previous decision tree problems and use n_{min} E {5, 10, 15, 20}. Calculate the MSE using ten fold cross-validation for each value of n_{min} and report the average and standard deviation across the folds (summarize your results in a table).
    - Does n_{min} impact the results significantly? Explain your answer.

Deliverables

A zip file containing the following:

1. a working project (source code, makefiles if needed, etc)
2. a report for the detailed description of the project
    a. explain the main aspects of your code
    b. how to run your project
    c. plots and diagrams

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***