Homework Assignment 4

600.435 Artificial Intelligence
Spring 2017
Due: May 4th

Machine Learning

This assignment will focus on implementing three of these techniques. In order to assess the effectiveness of these methods, you will be training and testing your code on real datasets. We will provide some code scaffolding that you should download and modify to complete the problems in the assignment. You can find links to the code on the course webpage.

Datasets

The provided datasets all come from the UCI machine learning repository under classification datasets. We have selected three datasets that we believe will give a diversity of results from our selected methods.

Congressional Voting Records Dataset

This dataset classifies congressmen into democrats and republicans based on their voting record. This is a somewhat easy problem to solve, and you should achieve fairly high accuracy with even simple methods of classification. This is a good dataset to use for debugging purposes.

MONKS Problems Dataset

This dataset has arbitrary attributes and uses 0 and 1 as labels. This dataset was generated in order to be a difficult problem to solve, and was used for a learning algorithm competition. There are three problems included in this dataset, please be sure to test your algorithms on all three! We don't expect great performance on this dataset because by its nature it is a difficult problem to solve. Note that some attributes here have different ranges of values than others.

Iris Dataset

This dataset classifies flowers into different classes of iris according to the size of different features. This is a classic classification problem because it is a mix of linearly separable and inseparable classes.

Methods

This section outlines the methods you will need to implement and test against the provided datasets. Please be sure to not only report your accuracy for each dataset, but also precision and recall.

Decision Tree

For this method you will implement a decision tree that split based on the attribute that offers the maximum information gain as outlined in section 18.3 of the textbook. Please be sure to add an option for using pure information gain or an information gain ratio to split. The information gain ratio should be information gain divided by the range of values provided by the examined attribute. Please report the accuracy, precision, and recall for each dataset using IG and IG ratio. Also discuss the differences in these metrics between using pure information gain and an information gain ratio.

In addition to the above, please implement pruning as outlined in section 18.3.5 of the textbook. Please also report the differences in the above metrics for your datasets between using pruning and not using pruning.

Naive Bayes

For this method you will implement a naive bayes classifier as outlined in section 20.2 of the textbook. Please be sure to report the accuracy, precision, and recall of the method on all datasets.

Neural Network

For this method you will implement a neural network as outlined in section 18.7 of the textbook. We will give you a bit of flexibility in how you implement your neural network, but please be sure to use at least one hidden layer (otherwise this is approach is trivial to implement). Also implement an alternate weight initialization scheme as shown in the class lecture on neural networks. Report the accuracy, precision, and recall of this method on all datasets with and without this alternate weight initialization and discuss the differences in these metrics between using the default weight initialization and one of your own.

Once you have implemented all three methods and collected data for each, discuss how the performances of the algorithms compares on each dataset. Which method would you consider the "best" for each dataset? Which metric(s) do you use to determine this, and how does the nature of the dataset itself affect your decision?

Getting Started

A good first step would be to examine the provided code scaffolding and start working with numpy. Think about what structures you need to solve these problems, and examine what data you would need to store for each method. If you don't have much experience programming data structures in python or just need help understanding how these methods work, please be sure to contact the TA (rmarvin3@jhu.edu) to arrange a meeting.

We will train your models by calling this command:

```
python classify.py --data [train_file] --mode train --model-file
name.model --algorithm decision_tree
```

We will test your models by calling this command:

```
python classify.py --data [test_file] --mode test --model-file
name.model --algorithm [algorithm_type]
```

Make sure that if you change any code, we can still call these commands on your code. Do not change the final output of the program, or your result will be wrong.

Submission Requirements

Please submit to Gradescope the following:

- A .zip file containing:
  - README that includes a brief description of each class. If there are any known bugs at the time of submission, please be sure to include these and a breakdown of debugging steps taken to ensure we can give you as much partial credit as possible (major bugs that aren't explained will make you lose credit for that particular method).
  - All code needed to train and test your methods on the provided datasets (you do not need to include the actual datasets in the .zip file, but please include all other code).
- A PDF that includes the breakdown of work between partners, an explanation of any lingering errors, and all answers to discussion prompts in the assignment. These are found in the methods section.