CSCI 446 Artificial Intelligence Project 2 Final Report

Roy Smart Nevin Leh Brian Marsh

November 20, 2016

Abstract

1 Introduction

Define MLA acronym here!!!!

2 Datasets

In this project we are asked to train our MLAs on five different datasets: the Wisconsin Breast Cancer Database, the Glass Identification Database, the Iris Plants Database, the Small Soybean Database, and the 1984 United States Congressional Voting Records Database. Each dataset has a different set of challenges to our MLAs, for example, the soybean dataset is really small, the cancer database is reasonably large, and the attributes in the glass dataset are not very well correlated (as we will soon see).

2.1 Discretization

Two datasets, the iris dataset and the glass dataset have continuous values. We are asked to discretize these dataset to more accurately compare the different MLAs. For our project, we adopted *binning* as our discretization scheme. Binning is a simple method for discretization that defines a certain number of adjacent intervals, known as bins, and then replaces points that fall within each bin by a value representative of that bin.

For the data in this report the number of bins was set to 10. In principle, the ideal number of bins in terms of MLA precision could be any integer larger than one, but through tests we found that the ideal number of bins is proportional to the number of classes in the dataset.

2.2 Cross-Validation

Cross-validation is statistical bootstrapping techinque used to estimate the generality of a statistical model using a dataset independent of the dataset used to create the model. In the context fo machine learning, this means that we will partition the full datasets into training and validation datasets. For this project we are asked to use the so-called k-fold cross validation technique, where the full dataset is partitioned into training and testing datasets through the use of *folds*. The folds are found by splitting the dataset into k equal sized pieces, and assigning each to be a fold. One fold is selected as the validation set, and the rest of the folds are assigned to the training set. This process is then repeated k times, where each fold is taken to be the validation set once.

To further increase the accuracy of our cross-validation process, we will *stratify* the folds described by k-fold cross-validation technique. Stratification in this context means that each fold will have an equal distribution of the classes in each dataset.. This process makes cross-validation more accurate, because it gives each MLA adequate information on each class for every training dataset. Without stratification, some training datasets could give a MLA zero examples of a particular class, hurting its ability to generalize.

In k-fold cross-validation, k is a free parameter that can be tuned by the user. For this project, we have selected $k = 10$, because it will allow us to perform the convergence tests discussed in Section 7.1 over a larger range.

2.3 Missing Values

Some of the datasets have missing values. This is problematic since most of the MLA's will yield vastly different answers if some data is missing. However, we note that the appearance of missing values in the voting dataset is merely an illusion, missing values in this dataset can signify a stance on a particular issue and this information should be used by our algorithms.

Since the voting set contains no missing values, the only dataset with missing values is the cancer dataset. Since there are so few missing values and since the cancer dataset is so large, we have decided to simply delete any data with missing values from the cancer dataset. Therefore, we have found that data imputation is not necessary for this project.

3 k-Nearest Neighbors

3.1 Training

3.1.1 Constructing Probability Table

3.2 Validation

3.2.1 Value Distance Metric

3.2.2 Determining k and p

4 Naive Bayes

4.1 Training

4.1.1 Constructing Probability Table

4.2 Validation

4.2.1 Determining Class Probability Distribution

5 TAN

5.1 Training

5.1.1 Constructing Probability Table

5.1.2 Constructing Augmented Tree

5.2 Validation

5.2.1 Determining Class Probability Distribution

6 ID3

6.1 Training

6.2 Validation

7 Results

7.1 Algorithm Convergence

7.2 Algorithm Precision

8 Conclusion