Assignment 2 Report

Nikhil Prasad z5049360

# Introduction

For my project, I decided to work on an implementation of the k-Nearest-Neighbor algorithm. Since the algorithm is rather straightforward, most of my project focused on the implementation of the algorithm and the visualization. I also made some quick charts that show how the error in the dataset changes as the value is changed.

The code is included in this submission, but is also listed online at <https://github.com/globmont/KNearestNeighbor>. Going to the GitHub link may make it easier to read the README file for the code since it’s written in markdown.

# Implementation

The implementation for this project was rather simple. I wrote the project from scratch in Java, and I tried to make it as versatile as possible given my time constraints. With only slight modification to the program, it should be able to read in any dataset in the form of an ARFF file and run the kNN or wNN algorithms on it.

To start, the program reads in the specified ARFF file. Then, it creates a dataset consisting of all the observations in the ARFF file. It then calculates the distance between every pair of points in the dataset for use in the kNN or wNN algorithms later. Then, for each observation in the dataset, it sorts the other points by their distance from the point in concern. Finally, the program takes the first points from the sorted list and depending on whether the data is categorical or numerical, calculates the output or class value, and computes some value that will estimate the error for the whole dataset. In the case of numerical data, the dataset error value is the average error for the whole set. That is, for a set of observations, and for observation there is a set of input parameters , the actual value and the predicted value , the error for the whole dataset is calculated by the following formula: . For a categorical dataset, the dataset error value is simply the number of misclassified instances in the dataset.

When the program is run, the console shows the values of , and for each observation since the program uses leave-one-out cross-validation. The last line of the console output shows the overall dataset error. However, this information is also displayed in the visualization that is launched when the program is run.

# Experimentation

The program was tested on three datasets: autos.arff, ionosphere.arff and iris.arff. The autos dataset is a numerical dataset, and both the ionosphere and iris datasets are categorical. When comparing the difference between the wNN and kNN algorithms for the autos dataset, we see that there really is not too much difference between the two algorithms’ outputs. However, when looking at the ionosphere and iris datasets, we see that even after accounting for the difference in size in the datasets (ionosphere has more than double the number of observations that iris has), the iris dataset still has a much lower error value. Especially when you bring the value up to around 19, the iris dataset only has 3 errors in the whole dataset, giving an error rate of while the ionosphere dataset has 57 errors giving an error rate of around . This may be due to the fact that the ionosphere dataset has so many more input parameters than the iris dataset does. The ionosphere dataset has 34 inputs while the iris dataset only has 4. Thus, each parameter is able to make more of a change in the iris dataset, and that may account for the lower error bounds in the iris dataset. Of course, more testing would need to be done on various datasets with various k values to actually verify that hypothesis. The charts are shown in the appendix.

# References

1. http://www.math.le.ac.uk/people/ag153/homepage/KNN/OliverKNN\_Talk.pdf

# Appendix