Project 2: Feature Selection with Nearest Neighbor

Student Name: Arman Essaian SID: 862072094

Solution:

Dataset	Best Feature Set	Accuracy
Small Number: 121	Forward Selection = {5, 3}	0.96
	Backward Elimination = {3, 4, 5, 6, 7, 9}	0.91
Large Number: 121	Forward Selection = {39, 23}	0.966
	Backward Elimination = {31}	0.82

----------<Begin Report>----------

In completing this project, I consulted following resources:

Standard C++ Library Reference

Lecture slides

Project briefing

I. Introduction

It is important to consider the features of a dataset when building a classifier for datasets. The features of a dataset are a measurable piece of data commonly represented as a column of a set. A good feature provides high accuracy in terms of correct classifications. Not all features of a dataset are required for use. We can increase the accuracy using a set of the highest accuracy features. We first use a method of feature search, Forward Selection or Backwards Elimination. Both searches will add/remove a feature one-by-one in order to find what features give us our highest accuracy. We use K-cross validation to check our classifier, k-nearest neighbors. kNN is sensitive to features and using an irrelevant feature affects the accuracy. The larger the number of correct classifications over the total number of instances in a dataset the higher the accuracy. In this project, our primary goal is to point out what subset of features can give us the best accuracy. This subset represents a set of features that are most desirable and won't give us irrelevant information.

II. Challenges

The biggest challenge, personally, was making the I/O method to read the files. This is a subject in programming that I always struggle with and need to read up on in order to fix errors. One main error I had to fix lied in the conversion from ascii to floating point when retrieving the data values. Another challenge was simplicity. As I started working on the project, I had to scrap multiple attempts due to overcomplicating my code. I think my current version is fairly simple, but I could definitely improve certain aspects of the code given more time including the backward elimination algorithm. I will touch on this later in the report. The final issue I had was attempting to normalize the data. I did not fully understand what was required and so I think my data may not be properly normalized for Experiment 2.

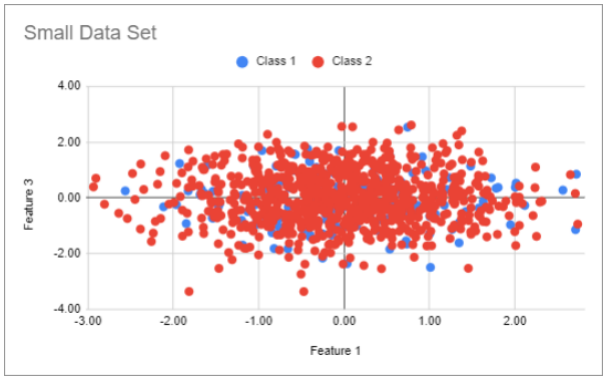
III. Code Design

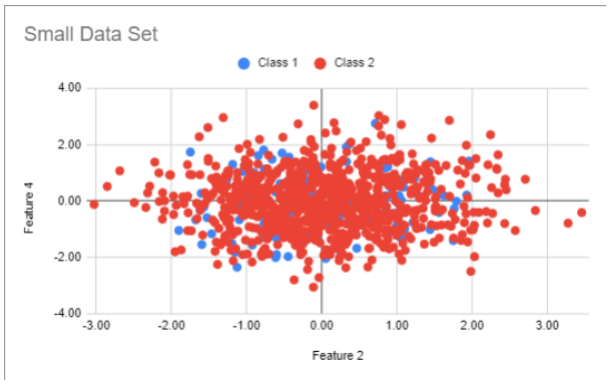
The main methods that make this code function are the I/O method to read from the files, forward and backward elimination search algorithms, and accuracy and distance calculation methods. I have a few helper functions that check if data is contained in certain vectors or sets as well.

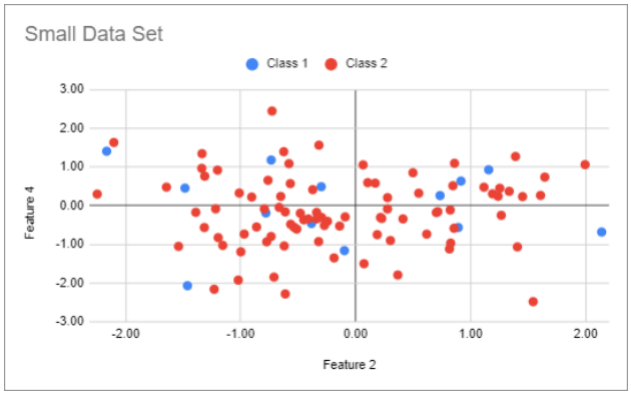
IV. Dataset details

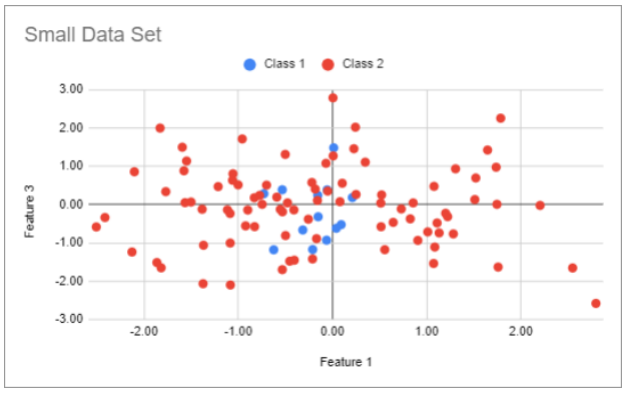
Your Small Dataset:10 features, 100 instances

Your Large Dataset: 40 features, 1000 instances









V. Algorithms

1. Forward Selection

2. Backward Elimination

VI. Analysis

Experiment 1: Comparing Forward Selection vs Backward Elimination.

In the Small Data Set, the Forward Selection(FS) achieves an accuracy of 96% selecting features 5 and 3 whereas Backward Elimination(BE) achieves an accuracy of 91% selecting features 3, 4, 5, 6, 7, and 9. Not only are the feature selections different, but BE has a drop in accuracy of 5%. The Large Data Set using FS achieves an accuracy of 96.6% selecting features 39 and 23 whereas BE only scores 82% accuracy selection feature 31 alone. In both datasets, FS has higher accuracy and, coincidentally, these accuracies are fairly similar with a difference of 0.6%. BE achieves lower accuracies in both datasets, but the amount of features differs dramatically between the two datasets. In the small set, BE selects 6 features which could potentially be the cause of the lower accuracy compared to FS since k-nearest neighbor is sensitive to irrelevant features.

VII. Conclusion

Subset {5, 3} provides us the highest accuracy of 96% with forward feature selections on the small dataset. The backwards elimination method provided us with a subset containing 5 and 3, but contained several additional false features. Features 5 and 3 are the best choices for relevant and meaningful features.

Overall, the subset {39,23} boasts the highest accuracy of 96.6% with Forward Selection on the large dataset. This could mean that features 39 and 23 are the best choices for relevant and meaningful features, however backward elimination never selected either of these values for the large dataset. There may be a better feature set with higher accuracy.

Forwards and backwards provided us with different results for the both datasets. The use of backwards vs forwards should depend on the size of a dataset. Backwards performed nearly as well as forward but faster on the small dataset but performed poorly on a large dataset. Each feature search method is different and should be used for different scenarios.

VIII. Trace of your small dataset

Welcome to Arman Essaian's Feature Selection Algorithm Program!

Type in the name of the file to test: CS170_Spring_2022_Small_data__121.txt

Type the number of the algorithm you want to run:

(1) Forward selection

(2) Backward Elimination

Processing data ...

This dataset has 10 features (not including the class attribute), and has 100 instances

Starting forward selection...

Using features(s) {1} accuracy is 77%

Using features(s) {2} accuracy is 74%

Using features(s) {3} accuracy is 79%

Using features(s) {4} accuracy is 76%

Using features(s) {5} accuracy is 83%

Using features(s) {6} accuracy is 78%

Using features(s) {7} accuracy is 80%

Using features(s) {8} accuracy is 83%

Using features(s) {9} accuracy is 73%

Using features(s) {10} accuracy is 76%

Feature set {5} was best at an accuracy of 83%

Using features(s) {1} accuracy is 88%

Using features(s) {2} accuracy is 79%

Using features(s) {3} accuracy is 96%

Using features(s) {4} accuracy is 72%

Using features(s) {6} accuracy is 84%

Using features(s) {7} accuracy is 86%

Using features(s) {8} accuracy is 84%

Using features(s) {9} accuracy is 78%

Using features(s) {10} accuracy is 88%

Feature set {3,5} was best at an accuracy of 96%

Removed for space

Using features(s) {8} accuracy is 83%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)
Feature set {8,5,3} was best at an accuracy of 83%

Using features(s) {2} accuracy is 80%
(Warning, Accuracy has decreased! Continuing search in case of local maxima)
Feature set {2,5,3} was best at an accuracy of 80%

Search completed.
The best feature subset is {5 3 } with an accuracy of 96%!