Ariel Lira

SID: 862125470

Email: alira008@ucr.edu

March 17, 2021

Project 2 for CS 170 Winter 2021, with Dr. Eamonn Keogh

All code is original, except:

Libraries I used for file and terminal output:

1. Iostream
2. Fstream
3. Sstream
4. Iomanip

C++ containers to help store data:

1. Vector
2. Unordered_set

Misc:

1. String: to store strings
2. Limits: to get max value of double
3. Cmath: to use sqrt() function
4. Ctime: to time my program

References for help:

1. For help with containers, string manipulation, and line parsing help, I used www.cplusplus.com.
2. For help with timing my program I used www.geeksforgeeks.org

In this project, we were tasked to implement a classification algorithm on a data set. Our goal was to classify data from a file and determine the accuracy of our classifications. The classification algorithm that we used for this project was the k nearest-neighbors algorithm. For this project, we only needed to find the nearest neighbor so k would default to 1. To figure out how accurate our classifications are, we are given test files that already have the correct classification. We compare our classifications with the correct ones and determine our accuracy.

Our next task was to find the best combination of features in our data set that would give us the highest accuracy. We used two different search algorithms to figure this out. For both search algorithms, we used a form of greedy search. We calculate what the accuracy of a combination of features, use the combination that has the best accuracy, and move on with our search. The first algorithm we used was forward selection search. With forward selection search, we start with no features and continuously add features that could increase our accuracy until we tried all possible features. The second algorithm we used was backwards elimination search. With backwards elimination search, we start with all the features and continuously remove features that could increase our accuracy until we have tried all possible features.

In Figure 1, we see the result of running forward selection search on CS170_SMALLtestdata__20.txt, which is the small file that was assigned to me.

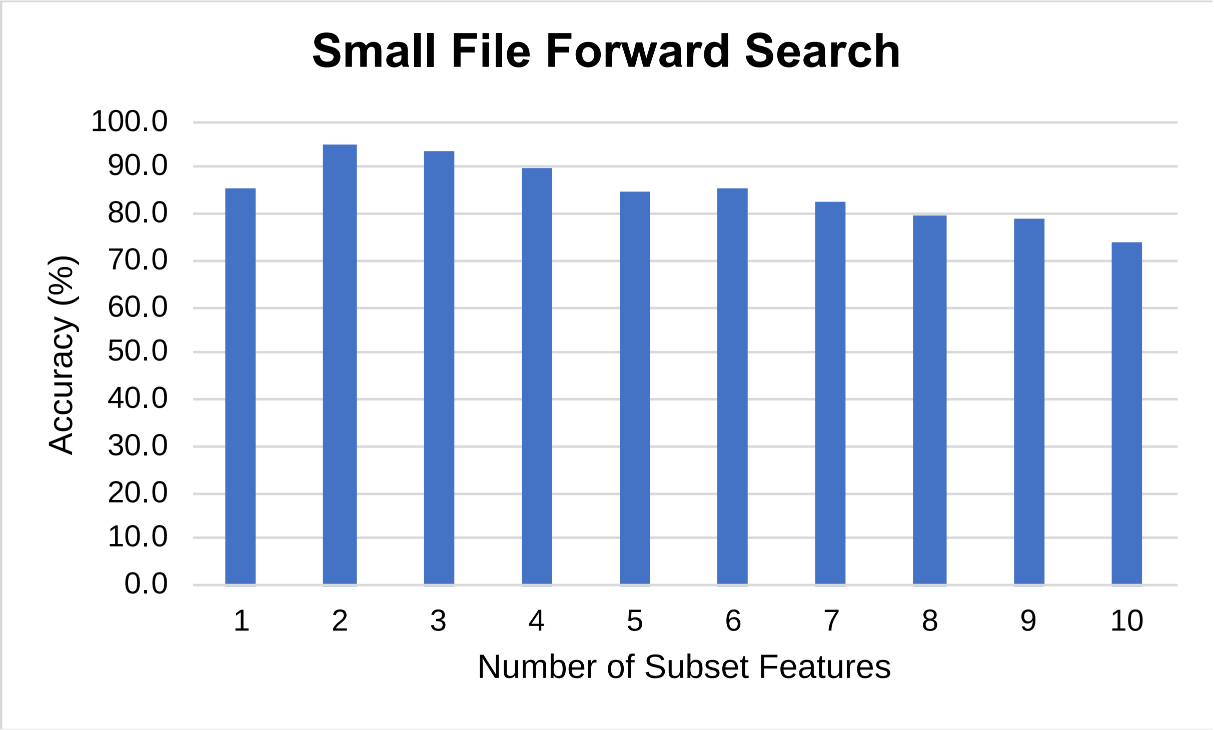

Figure 1: Accuracy of increasing search levels where subset of features increases using forward selection search.

At the beginning of the search, we have no features added. When we add the first best feature, {3}, we start with 85.667% accuracy. From the graph, we can see that the max accuracy of 95.333% accuracy with the feature subset {3,4}. After we add more than 2 features, the accuracy of the classification algorithm decreases. From looking at the graph, it looks like we could have a possible third feature, ‘8’, we can add even though the accuracy is 93.667%. The difference between the two subsets does not seem substantially big. At the end of the search, it seems that we have the lowest accuracy with all features included into our data. This accuracy was 74.000%.

In Figure 2, we see the result of running backwards elimination search on CS170_SMALLtestdata__20.txt, which is the small file that was assigned to me.

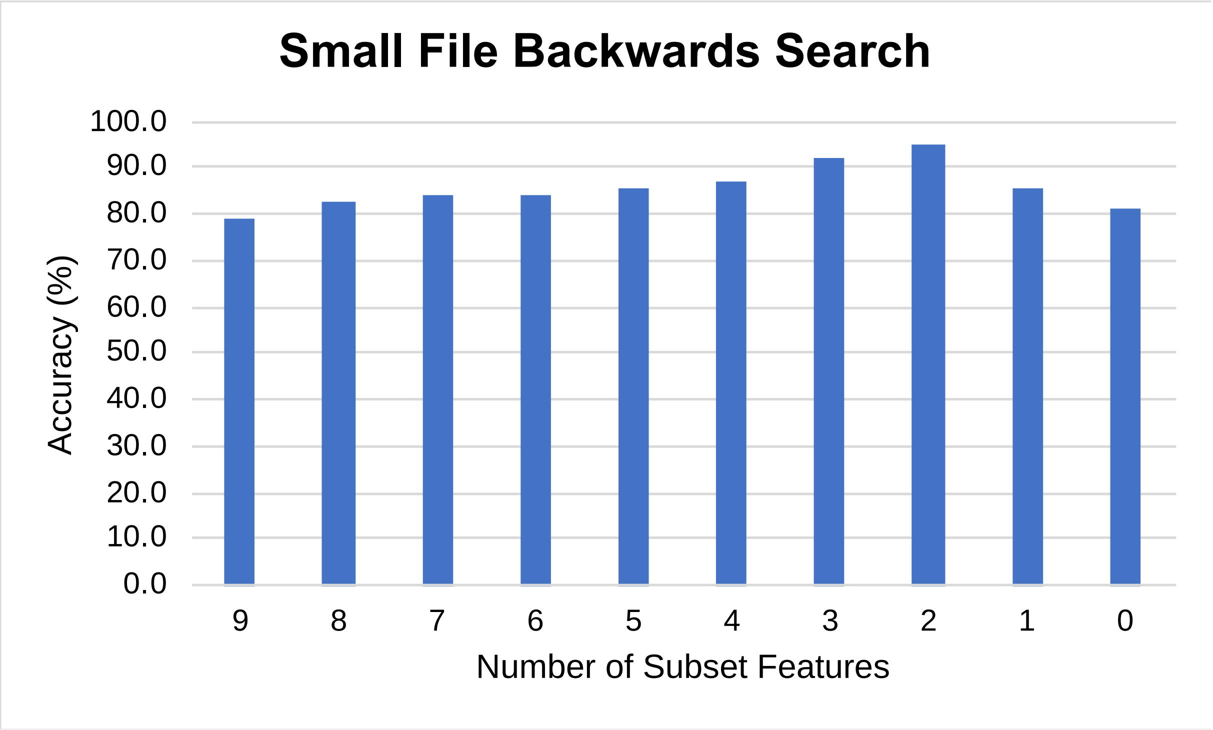

Figure 2: Accuracy of increasing search levels where subset of features increases using forward selection search.

At the beginning of the search, we have all features in our subset of features. The accuracy of removing the first feature seems to the 79.000%. As we remove features from the subset of features, we increase our accuracy. There seems to be a significant increase in accuracy when we have 2 features in our subset. When we have 2 features in our subset, {4,3}, we reach a maximum in our accuracy of subset features. The accuracy drastically decreases after removing one of these features.

Conclusion for Small Dataset

From looking at the results from the forward search and backward elimination search algorithms, I can determine that feature ‘3’ and feature ‘4’ are the best subset of features. This is some evidence that feature ‘8’ could also be a useful subset of features but there would need to be more research to see if that is true. The best accuracy for our best subset of features is 95.333%.

In Figure 3, we move onto a larger dataset. This will be a challenge because it has 100 features. This means there are 100 levels we have to search through to see the best combination of features. We can see here the result of running forward selection search on CS170_largetestdata__69.txt, which is the large file that was assigned to me.

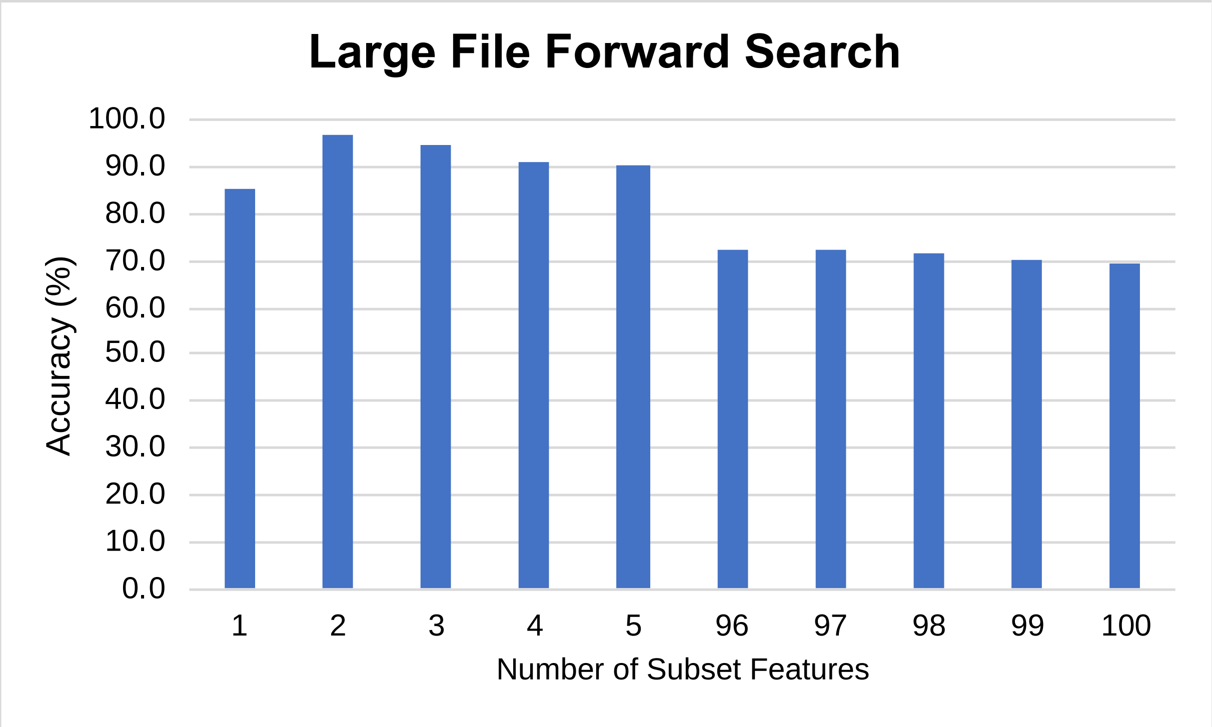

Figure 3: Accuracy of increasing search levels where subset of features increases using forward selection search.

I have omitted the middle 90 levels to save space for the large dataset files. The first feature that I added, ‘40’, turned out to have an 85.4% accuracy. When a second feature was added, our accuracy drastically increased. Our subset was {47, 40} when there were 2 features, with an accuracy of 97.0%. After the second feature was added, the accuracy of the search algorithm decreased slowly. When the third feature was added, for example, the accuracy was 95.0%. We reached the lowest accuracy of 69.4% when all features were in our subset.

In Figure 4, we see the result of running backwards elimination search on CS170_largetestdata__69.txt, which is the large file that was assigned to me.

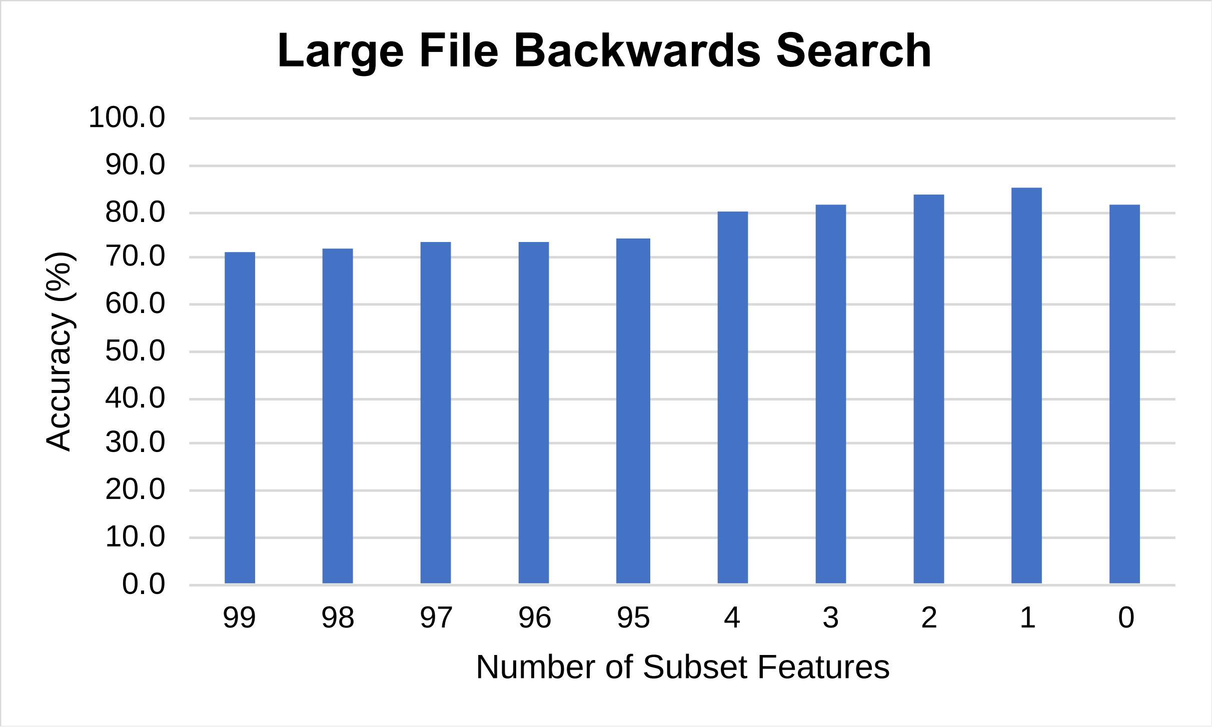

Figure 4: Accuracy of increasing search levels where subset of features increases using forward selection search.

For Figure 4, I omitted the middle 90 subsets of features for space again. From looking at the chart, the accuracies of the subset of featured increased the more features you removed. With the backwards elimination search algorithm, the max accuracy seemed to be at a subset where there was only one feature. This subset was {40} with an accuracy of 85.4. I believe that this could be a slight mistake. When the subset of features had 2 features, {40, 47}, the accuracy was at 83.4. This could be the true subset because it was the result from the forward selection search.

Conclusion of Large Dataset

From looking at the results from the forward search and backward elimination search algorithms, I can determine that feature ‘40’ are the best subset of features. This is some evidence that feature ‘47’ could also be a useful subset of features because this was the result subset of features from the forward search. More research would need to be done to determine if this were true. The best accuracy for our best subset of features is 85.4 %.

Computational Effort for Search

I implemented these search algorithms with the classification algorithm in Python initially but decided to switch to C++. The speed of computations was proving to be too slow in Python. C++ helped me complete the search in less time. I ran all the experiments on a MacBook Pro with a 2.6 GHz 6-Core Intel Core i7. In Table 1, I report the running time for the four times I run the program.

	Small Dataset (10 features, 300 instances)	Large Dataset (100 features, 500 instances)
Forward Selection	10.408 seconds	2.5019 hours
Backward Search	13.256 seconds	3.4899 hours

Table 1: Showing the running time of each experiment.

Below I show a single trace of my algorithm. I am only showing the forward selection search algorithm on the small dataset.

```
./prog
Which test data file would you like to try.
            1. Small test data file
            2. Large test data file

1
Which search algorithm would you like to use?
            1. Forward Selection
            2. Backward Elimination

2
Number of features: 10
Number of instances: 300
```

```
On level 1 of the search tree
        Considering removing feature 1 with 75.333% accuracy.
        Considering removing feature 2 with 76.000% accuracy.
        Considering removing feature 3 with 74.667% accuracy.
        Considering removing feature 4 with 69.000% accuracy.
        Considering removing feature 5 with 74.667% accuracy.
        Considering removing feature 6 with 74.000% accuracy.
        Considering removing feature 7 with 78.667% accuracy.
        Considering removing feature 8 with 75.333% accuracy.
        Considering removing feature 9 with 71.667% accuracy.
        Considering removing feature 10 with 79.000% accuracy.
        Removed feature 10
On level 1 , the best feature subset was { 9 8 7 6 5 4 3 2 1 } . Accuracy was 79.000%

On level 2 of the search tree
        Considering removing feature 1 with 81.000% accuracy.
        Considering removing feature 2 with 74.000% accuracy.
        Considering removing feature 3 with 74.333% accuracy.
        Considering removing feature 4 with 71.667% accuracy.
        Considering removing feature 5 with 82.667% accuracy.
        Considering removing feature 6 with 76.000% accuracy.
        Considering removing feature 7 with 81.667% accuracy.
        Considering removing feature 8 with 77.000% accuracy.
        Considering removing feature 9 with 77.333% accuracy.
        Removed feature 5
On level 2 , the best feature subset was { 9 8 7 6 4 3 2 1 } . Accuracy was 82.667%
```

{Here I delete half of the output to save space.}

```
On level 8 of the search tree
        Considering removing feature 2 with 95.333% accuracy.
        Considering removing feature 3 with 83.333% accuracy.
        Considering removing feature 4 with 70.333% accuracy.
        Removed feature 2
On level 8 , the best feature subset was { 4 3 } . Accuracy was 95.333%

On level 9 of the search tree
        Considering removing feature 3 with 85.667% accuracy.
        Considering removing feature 4 with 71.333% accuracy.
        Removed feature 3
**** Warning accuracy has decreased! Continuing search in case of local maxima. ****
On level 9 , the best feature subset was { 4 } . Accuracy was 85.667%

On level 10 of the search tree
        Considering removing feature 4 with 81.000% accuracy.
        Removed feature 4
**** Warning accuracy has decreased! Continuing search in case of local maxima. ****
On level 10 , the best feature subset was { } . Accuracy was 81.000%

Finished search! The best feature subset is { 3 4 } , which had an accuracy of 95.333%
Search took 13.283 seconds to find best combination of features.
```

Code: Below is my code for the project. If you wish to download it for yourself, you can go to my GitHub to download it.