Assignment 2 CS170. Introduction to Artificial Intelligence

Christopher Hyun Instructor: CS Eamonn Keogh

ID 861162836

chyun001@ucr.edu

17-December-2018

In completing this assignment, I consulted…

- PowerPoint "Project_Two_Briefing." I used the search algorithm on the slides as the base for my search when implementing Forward Selection, Backward Elimination, and the original algorithm.
- The machine learning slides 1-4 in order to understand cross validation and nearest neighbor algorithm.
- Watched this video (https://www.youtube.com/watch?v=pHOs7WomFSs) to further reinforce my understanding of leave-one-out cross validation.

All the important code is original. Unimportant subroutines that are not completely original are…

- The search algorithm on the PowerPoint. (Used it as a base)
- Matlab functions such as
    - load() To load the data
    - input() To get user input
    - disp() To display text
    - num2str() Convert string to int
    - size() Used to get size of the data
    - isempty() To see if array/matrix is empty
    - intersect() Used to return common data
    - norm() Used to get the distance between two points (euclidean distance)

Report

Overview

The feature selection algorithm is a program where you want to test how accurate a set of features are. This can be achieved by using a technique called leave-one-out cross validation and nearest neighbor. Cross validation tries to predict and classify a data point according to another nearby data point. In other words, for each data point, we want to find the shortest distance between all other data points using nearest neighbor and classify it according to its closest neighbor. The algorithms that utilize these techniques are

- Forward Selection
- Backward Elimination
- Christopher's Special Algorithm

Below, I will discuss more in depth how the most accurate set of features are extracted in each different algorithm.

Forward Selection

Forward Selection starts off with no features and checks each feature for the highest accuracy. It uses leave-one-out cross validation and nearest neighbor to calculate the accuracy. It will search through the data and keep adding features and will only keep the combination of features that yield the highest accuracy.

For example, lets say we have 3 features. You start off with an empty set.

Current_Set_Of_Features = []

At the first iteration, you might get something like this

Using feature {1} accuracy is 90%

Using feature {2} accuracy is 85%

Using feature {3} accuracy is 80%

As we can see, feature 1 yields the highest accuracy (90%) so we add that feature to our Current_Set_Of_Features. On the second iteration, you might get something like this

Current_Set_Of_Features = [1]

Using feature {1, 2} accuracy is 96%

Using feature {1, 3} accuracy is 82%

Since features 1 and 2 yields a higher accuracy then the previous accuracy, we add feature 2 to our Current_Set_Of_Features and etc.

Backward Elimination

Backward elimination is the opposite of forward selection. Instead of starting off with no features and adding them one by one, you start with all features and leave one feature out. The accuracy is also calculated using leave-one-out cross validation and nearest neighbor.

To give an example, lets say we have 3 features. You start off with a full set.

Current_Set_Of_Features = [1, 2, 3]

At the first iteration, you might get something like this

Leaving out feature {1} accuracy is 60%

Leaving out feature {2} accuracy is 70%

Leaving out feature {3} accuracy is 65%

As we can see, leaving out feature 2 yields the highest accuracy (70%) so we take out that feature in our Current_Set_Of_Features. On the second iteration, you might get something like this

Current_Set_Of_Features = [1, 3]

Leaving out feature {1} accuracy is 68%

Leaving out feature {3} accuracy is 86%

Since leaving out feature 3 results in a higher accuracy than the previous iteration, we can remove feature 3 in our Current_Set_Of_Features and etc.

Christopher's Special Algorithm

Now, this algorithm is a special algorithm which builds upon Forward Selection. It's the same as Forward Selection except to speed up the process, I have added a 'pruning' type of algorithm which helps speed up the process. This is how it works. Let's say you have data set with 200 instances and 100 features. Let's say on the first iteration, the highest accuracy you get is 85%. Now, for the second iteration, when adding a feature, constantly check if the accuracy is higher

than the previous highest accuracy. If the accuracy starts to decay under 85% simply return zero and check the second feature in the second iteration and so on.

Comparison

Okay, now that the algorithms have been explained, let's see how they do against each other first on the small data set. We will be comparing them according to

1. The highest accuracy.
2. The feature set that resulted in the highest accuracy
3. The time it took to get the accuracy/feature set.

Small Dataset #97 (10 features)

	Forward Selection	Backward Elimination	Christopher's Special Algo
Highest Accuracy	95	95	95
Feature Set	{1, 6, 9}	{1, 6, 9}	{1, 6, 9}
Time	4 Seconds	4 Seconds	3 Seconds

So on the small data set, every algorithm performed very similarly. All three algorithms achieved an accuracy of 96% with the same feature set {1, 6, 9}. Something to note was that Christopher's Special Algorithm achieved a second faster than the other two algorithms. However, performance wise, they are all very similar. Let's test the algorithm on a data set with much more features.

Large Dataset #21 (100 features)

	Forward Selection	Backward Elimination	Christopher's Special Algo
Highest Accuracy	96	89	90
Feature Set	{5, 96, 24}	{5, 82}	{5, 96, 100, 85}
Time	8 Minutes	8 Minutes	2Min 44Sec

After running the data set with 100 features there was a big difference in run time. Forward Selection and Backward Elimination had a similar run time but Backward Elimination had a worse accuracy. Christopher's Special Algorithm on the other hand was much faster than both of the algorithms and was more accurate than Backward Elimination but less accurate then forward Selection.

Conclusion

So going off of the test results from above, we can see that for small features, there isn't a difference in accuracy and feature set. All the algorithms run pretty fast and seem to agree with each other. However, when it comes to a large amount of features, there was a lot of discrepancy. Forward Selection was the most accurate but was very slow. Christopher's Special Algorithm was the fastest but it was the least accurate. Backward Elimination was the least accurate and as slow as Forward Selection. In theory, Christopher's Special Algorithm should have been as accurate as Forward Selection since Christopher's Special Algorithm is just Forward Selection pruned but that was not the case.

In conclusion, when it comes to accuracy

Forward Selection > Christopher's Special Algorithm > Backward Elimination

When it comes to run time

Christopher's Special Algorithm > Forward Selection > Backward Elimination