c

cs 205 aRTIFICIAL INTELLIGENCE:   
FEATURE SELECTION WITH NEAREST NEIGHBOR

Sandesh Basrur Sid: 861244757

**CS 205: Introduction to Artificial Intelligence**

**Sandesh Basrur: 861244757**

Nearest neighbor is one of the most simple and effective search algorithms. Despite its simplicity, it has been successful in large number of classification and regression problems.

However if the dataset has irrelevant or correlated features the accuracy and efficiency suffers leading to high computational costs.

One of the weakness of nearest neighbor is that, it is sensitive to units of measurements. To mitigate this problem we normalize the dataset using Z-normalization where we use the relation X = (X – mean(X)) / standard deviation(X)

Since in the given dataset the units of features are unknown, the data is normalized before performing the search operation. The data is reduced to its standard/normal form and hence data manipulation becomes more easier.

In forward elimination, we find the best combination of variables by starting the search with single variable and then increasing number of variables used by adding a new variable at every level. We perform this until we find an optimum solution. In my implementation of forward elimination, for the smaller dataset it took 4.6 seconds and achieved an accuracy of 93% for the dataset [4, 8]. For the larger dataset, it performed far worse than the custom search and achieved an accuracy of 89% with [25, 41, 52, 20, 91, 62, 42, 22, 72, 46, 78, 31, 93, 42, 82, 13, 9]

Backward elimination starts with a set containing all the variables and removes one variable at a time at each level until the feature subset contains just one element. For the smaller dataset, it took more time of 13.1 seconds and achieved an accuracy of 93% for the dataset [4, 8]. For the larger dataset it performed worse than forward elimination and custom search and had far more features in the best feature subset [25, 41, 52, 20, 91, 62, 42, 22, 89, 23, 45, 28, 13, 73, 37, 57, 98, 83, 27, 65] at an accuracy of 87%

In the custom search, the features subsets are saved in a priority queue with a custom comparator where features are compared based on accuracy and stored in the decreasing order of accuracy. Thus at every level the features subset with the most accuracy is chosen and additional variable is added in order to verify if adding additional variable will improve the accuracy. It has to expand far lesser number of subsets than forward selection and backward elimination. It chooses to expand the subset based on accuracy level rather than size of subsets and thus if there are more accurate subsets which are smaller, they will be expanded instead of expanding a larger subset with less accuracy. It also avoids the cost of expanding these less accurate larger subset. Thus it executes faster compared to forward selection and backward elimination. For larger dataset, It achieved an accuracy of 85% for feature subset [25, 41, 52, 20, 91, 62, 42, 22]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number of features** | **Forward elimination**  **Time of execution** | | **Backward elimination**  **Time of execution** | | **Custom Search**  **Time of execution** | |
|  | Time in seconds | Accuracy | Time in seconds | Accuracy | Time in seconds | Accuracy |
| 10 | 4.6 seconds | 93% | 13.2 seconds | 93% | 3 seconds | 85% |
| 100 | 1080 seconds | 89% | 3520 seconds | 87% | 336  seconds | 85% |

References:

<http://scikit-learn.org/stable/modules/neighbors.html>

K-fold-cross-validation: https://www.youtube.com/watch?v=nZAM5OXrktY

http://machinelearningmastery.com/evaluate-performance-machine-learning-algorithms-python-using-resampling/

http://scikit-learn.org/stable/modules/neighbors.html

https://www.google.com/search?q=%22forward+selection%22&ie=utf-8&oe=utf-8#q=%22forward+selection%22&start=100

http://luisangeloshome.weebly.com/uploads/1/6/2/5/16253368/analysis.pdf

http://stackoverflow.com/questions/15389768/standard-deviation-of-a-list

http://www.dr-chuck.com/csev-blog/2010/01/fun-python-syntax-operator-itemgetter-sorted-and-lambda-functions-oh-my/

Please enter file name: cs\_205\_small5.txt

1. Forward Selection

2. Backward Selection

3. Sandesh's Custom Search

Enter your choice: 3

Normalizing dataset

Starting custom search..

Using feature [4] the accuracy is 84.0%

Using feature [4, 2] the accuracy is 84.0%

Using feature [4, 2, 8] the accuracy is 85.0%

Using feature [4, 2, 8, 6] the accuracy is 74.0%

Accuracy is decreasing. Searching for local maxima

Using feature [4, 2, 8, 6, 5] the accuracy is 72.0%

Accuracy is decreasing. Searching for local maxima

Using feature [4, 2, 8, 6, 5, 1] the accuracy is 65.0%

Accuracy is decreasing. Searching for local maxima

Using feature [4, 2, 8, 6, 5, 1, 3] the accuracy is 63.0%

Accuracy is decreasing. Searching for local maxima

Using feature [4, 2, 8, 6, 5, 1, 3, 7] the accuracy is 62.0%

Accuracy is decreasing. Searching for local maxima

Using feature [4, 2, 8, 6, 5, 1, 3, 7, 9] the accuracy is 64.0%

Accuracy is decreasing. Searching for local maxima

Using feature [4, 2, 8, 6, 5, 1, 3, 7, 9, 10] the accuracy is 63.0%

Accuracy is decreasing. Searching for local maxima

Feature [4, 2, 8] has the best accuracy of 85.0%

Time taken: 2916.99981689 milliseconds

**For larger dataset:**

Please enter file name: cs\_205\_large5.txt

1. Forward Selection

2. Backward Selection

3. Sandesh's Custom Search

Enter your choice: 3

Normalizing dataset

Starting custom search..

Using feature [25] the accuracy is 83.0%

Using feature [25, 41] the accuracy is 75.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52] the accuracy is 64.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20] the accuracy is 65.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91] the accuracy is 78.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62] the accuracy is 77.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42] the accuracy is 82.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22] the accuracy is 85.0%

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95] the accuracy is 80.0%

Accuracy is decreasing. Searching for local maxima

…..

*# skipped portion of output for terseness*

…….

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47] the accuracy is 77.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30] the accuracy is 77.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30, 9] the accuracy is 81.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30, 9, 100] the accuracy is 82.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30, 9, 100, 70] the accuracy is 82.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30, 9, 100, 70, 57] the accuracy is 81.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30, 9, 100, 70, 57, 51] the accuracy is 81.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30, 9, 100, 70, 57, 51, 50] the accuracy is 81.0%

Accuracy is decreasing. Searching for local maxima

Using feature [25, 41, 52, 20, 91, 62, 42, 22, 95, 87, 67, 81, 43, 38, 21, 11, 5, 3, 86, 82, 58, 32, 28, 24, 90, 68, 40, 26, 98, 74, 66, 61, 56, 55, 47, 30, 9, 100, 70, 57, 51, 50, 49] the accuracy is 82.0%

Custom Search is completed. Feature [25, 41, 52, 20, 91, 62, 42, 22] has the best accuracy of 85.0%

Time taken: 336530.999899 milliseconds