DAA Assignment Report

Implementing and investigating the least segmented squares algorithm.

R Monith Sourya (2016A7PS0006H),
Nikhil Sreekumar Nair (2016A7PS0006H)

Testing Criteria

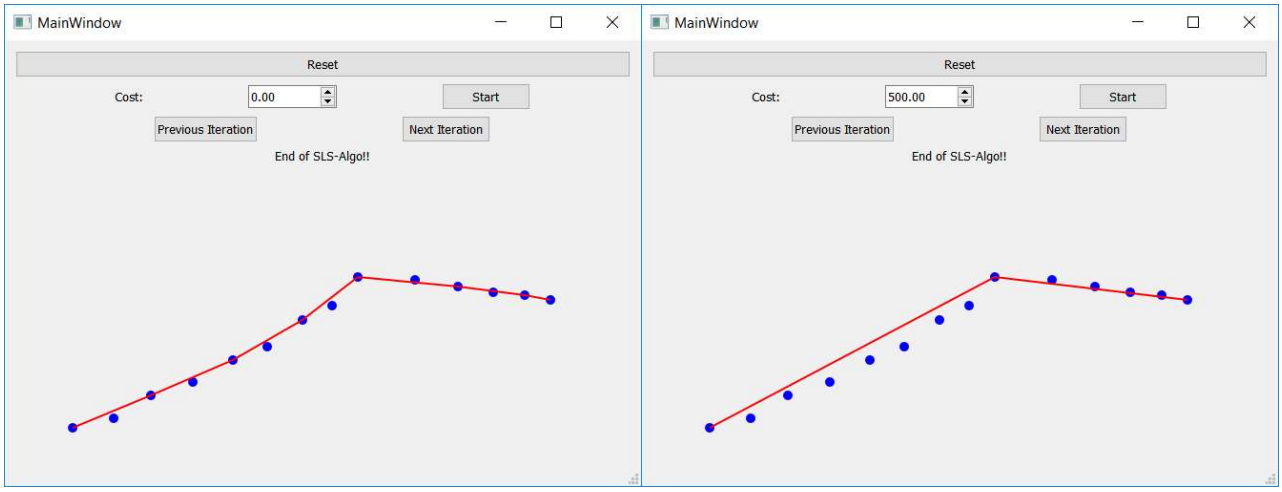
The least segmented squares algorithm is an overall $O(n^3)$ complexity algorithm, but that is not the major point of investigation in this report. This report tries to investigate the working conditions for this algorithm and looks into its behaviour with differing cost values. The algorithm is built off the least squares regression approximation and decides to fit the input points into segments, whose total count is decided by (i) the least squares errors cumulated for all points around a segment, and (ii) the cost of creating a new segmented (quantified by the input cost value).

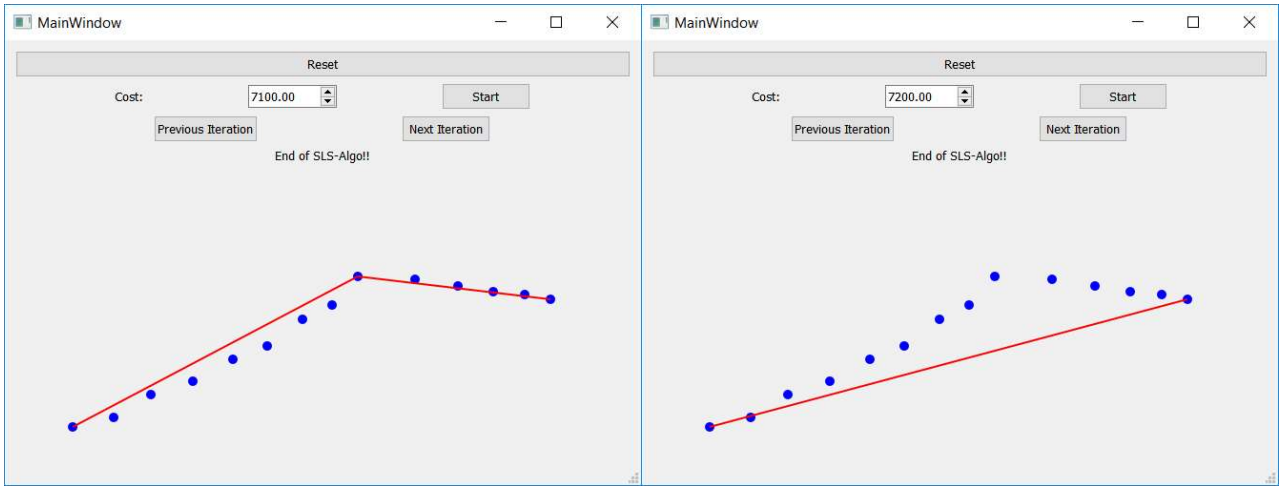
Observations

Before delving into the observations that result by changing the cost functions, it is important to understand how segments are depicted in the GUI this report uses for reference.

The GUI uses a representational coordinate axes system that has increasing *x-values* in a left-to-right direction and increasing *y-values* in a top-to-bottom direction. Also, the error values might seem very large from a plain magnitude point of view; however, this is necessary again because of how the GUI has been designed to depict coordinate points. For reference, the leftmost and rightmost points in the adjoining screenshots are (60, 271) and (561, 137) respectively.

Having established the GUI's specifications, we can now delve into the direct effect of changing the cost values on the decision of line segments. The first two screenshots simply depict the change in line segments on change in cost from 0 to 500. It might be tempting to believe that the algorithm is misbehaving in the first case (cost = 0), however, on inspecting the individual cost values of each segment, we clearly notice that the orders of those errors are negligible and practically in line with 0. On changing the cost to 500, the algorithm takes a sharp turn to 2 segments from 7 (when cost = 0) since the cost of creating a new segment becomes too much.





For costs ranging from about 500 to 7100, we see no change in segments, simply because the cost of creating a new a segment is still smaller than the cost of squares cumulative. However, 7100 happens to be just the threshold; at 7200, the algorithm switches to a single segment connecting the first and last segments, choosing to take the cumulative squares errors over the overhead of creating a new segment. Clearly, since we're looking at only a single segment connecting the extremes, for cost values above this (obviously for this particular test case of input points), the result will be the same, graphically. The only thing that will change is the total error value, simply due to increasing cost values.

```
Cost of the optimal solution : 0.000000
Optimal segments:
y = (-0.232558)x + 284.953 from (60,271) to (142,237); squared error: 2.34958e-028
y = (-0.318182)x + 282.182 from (142,237) to (228,200); squared error: 7.19096e-029
y = (-0.388889)x + 288.667 from (228,200) to (301,158); squared error: 6.07669e-028
y = (-0.483871)x + 303.645 from (301,158) to (359,113); squared error: 1.04623e-027
y = (0.05)x + 95.05 from (359,113) to (464,123); squared error: 6.22653e-030
y = (0.162162)x + 47.7568 from (464,123) to (534,132); squared error: 5.23853e-030
y = (0.185185)x + 33.1111 from (534,132) to (561,137); squared error: 1.22428e-029
60 271 142 237
142 237 228 200
228 200 301 158
301 158 359 113
359 113 464 123
464 123 534 132
534 132 561 137

Cost of the optimal solution : 1187.654545
Optimal segments:
y = (-0.478844)x + 306.602 from (60,271) to (359,113); squared error: 175.265
y = (0.12276)x + 66.9343 from (359,113) to (561,137); squared error: 12.3896
60 271 359 113
359 113 561 137

Cost of the optimal solution : 14387.654545
Optimal segments:
y = (-0.478844)x + 306.602 from (60,271) to (359,113); squared error: 175.265
y = (0.12276)x + 66.9343 from (359,113) to (561,137); squared error: 12.3896
```

```
Cost of the optimal solution : 14387.654545
Optimal segments:
y = (-0.478844)x + 306.602 from (60,271) to (359,113); squared error: 175.265
y = (0.12276)x + 66.9343 from (359,113) to (561,137); squared error: 12.3896
60 271 359 113
359 113 561 137

Cost of the optimal solution : 14533.121033
Optimal segments:
y = (-0.308736)x + 271.722 from (60,271) to (561,137); squared error: 7333.12
60 271 561 137
```

Takeaways and Inferences

This report investigates the functioning of the segmented least squares algorithm on changing cost values for a particular test case of input points. Throughout the four situations of different costs that we considered, the results were pretty much the way we had expected them to be. However, the least segmented squares algorithm has one major limitation, one that appears due to the fact that the algorithm uses the least squares regression analysis as its building block.

The least squares formulation only takes into consideration observational errors in the dependent variable, i.e. lines must be depicted in the form $y = ax + b$ so that errors can be computed as $(y_i – ax_i - b)^2$ for any points (x_i, y_i) wrt the line $y = ax + b$. Moreover, the best fit line is also determined wrt this representation: $a = n\sum x_i y_i – (\sum x_i)(\sum y_i) / n\sum x_i^2 – (\sum x_i)^2$ and $b = \sum y_i - a\sum x_i / n$. Now, on the off chance that we are to hit a position where our line of best-fit must be a vertical line, i.e. of the form $x = b_k$, then we realise that the same cannot be represented in the earlier form. In more formal terms, this happens to be a case where there is an observational error in the independent variable, a situation that the least squares method is simply not equipped for. As a consequence of this, the segmented least squares algorithm also cannot take care of such test cases and will give very erratic outputs that make little or no semantic sense.

(Cases like these, where there might be observational errors in both the dependent as well as the independent variables are better dealt with using a different regression formulation, called the total squares analysis.)