One of the issues with predicting a job salary is choosing what key words we should use to predict them. The amount of words that we can choose and the combinations of how they interact are countless, even in small datasets. Fortunately, there are ways of automatically choosing what words we should choose in predicting the salary for the ads. One such method is the Lasso method.

The lasso method attempts to solve the following equation:

Which is minimizing the least square solution added to it the absolute values of the value of the words. By making lambda large you are making **x**, the number of words you predict salaries on, small. In doing this the number of values that are zero increases thus making the numbers of words that we are using less. More importantly by solving this optimization problem we are removing words that are occurring less in the dataset thus removing the sparsity in the matrix (ie removing mostly zero word occurrences in lots of examples).

So how do we write the algorithm for the Lasso method? There is no close form solution to the solving the lasso method. However there are good approximates that give the solution to it very quickly.

Here is the code for it

Initialize Xsolution and XnewSolution to all zeros

get SVD of A by [T,S,V] = svd(A)

Initialize alpha to 1/(largest svd value of A) ( S(1,1) )

Set delta to large value

Set iteration to 0

While the delta is larger than the tolerance && Iter is less than iter max

Xsolution = XnewSolution

Y = Xsolution + alpha \* A’ \* (b – A\* Xsolution)

XnewSolution  = sign(y) \* max(abs(y) – alpha \*lambda, 0)

Iter++

delta = || XnewSolution  - Xsolution ||

end

and return Xsolution  as the predictor xhat.

Try writing up the Lasso method and try it with lambda = 10. The resulting xhat value should be all zeroes except for 2 values.

xhat =

1.0e+06 \*

0

0

0

0

0

0

0

0

0

0

0.7166

0

0

0

0

0

0

0

2.3215

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

By checking what words the non-zero values correspond to we get the relevant words that we should be using for our predictor. In this case the important words that we should use as a predictor are “experience” and “client”. By using the current xhat values we can use it to predict the salary on the warmup test set description and see how it compares to the regulation and least squares methods. The predicted salary we obtained was 29554 pounds which is 15446 pounds off. This is an improvement over the general least squares method in this case and ( **ADD COMPARISON to Regularization HERE).**

Warmup on Lasso:

Let us try the Lasso method with different lambda values (lambda = 10, 5, 2, 1) and see how many words are chosen and what they are as the key words and how much does it affect our predicted salaries.

Solution:

|  |  |  |  |
| --- | --- | --- | --- |
| λ | # of important words | Important words | Error on predicted salary |
| 10 | 1 | experience, client | 15446 |
| 5 | 5 | experience  client  engineering  responsible  development | 16916 |
| 2 | 7 | experience  engineering  responsible  client  development  looking  providing | 22422 |
| 1 | 10 | experience  engineering  responsible  client  providing  looking  development  engineer  must  specialist | 26261 |

As you can clearly see, the higher the lambda value the less words that appear in the list. It is also useful to know is that the if we if we have a large lambda contains a subset of the words in the smaller lambda signifying that the word will always remain important in the smaller lambda cases.

Post Lab Activity:

Let us try the Lasso on the larger activity dataset and with different and larger lambda values (10000, 100, 500, 400, 300, 250, and 200) and see what the lasso comes with as the important words and the predicted error on the salaries.