The original motivation: Overfitting & Analytic solution

Back to the method of primary linear regression, we can find two annoying shortcomings up there:

First, when m (the amount of training samples) > n (the dimension of training vector), you will get a great chance that X transposed \* X is noninvertible ,which makes it very hard to give an analytic solution on that optimization.



However, on the opposite, when m is closed to n, the overfitting may come up. We don’t want a

Complicated yet not general model. 3 methods are:

1) choose a subset of vector X with some principles defined.

2) decrease the dimension of X. PCR & PLS, make them uncorrelated by remapping them.

3) shrinkage methods, this is the topic here.

Shrinkage Method & Sparse Constraint

They introduced an additional term related with Θ (the parameters) , which is a constraint, or punishment for complication. The goal is trying to make Θ sparse.

3 popular terms:







If you choose l2 norm, then it is called ridge regression. The form is:



equals to the optimization objective:



There’s another concept: Sparseness. In fact, the training samples we got usually contain noises and redundancy. A sparse Θ(Some of Xi = 0) means you somehow find those meaningful components, which actually prevents an overfitting. So we desire a sparse Θ.

Is l2 norm a sparse constraint?



Apparently you can notice that all the  are together determined by the  alone, which means the sparseness cannot do any particular benefits for decreasing your , so when you eventually minimize your , the chance of capturing a sparse solution  is pretty small, though you may find some  are really close to 0, but they aren’t 0.

If you choose l0 norm, the description for that is to minimize the cost function while the amount of non-zero xi is less than given constant K, then of course, you will gain a sparseness.



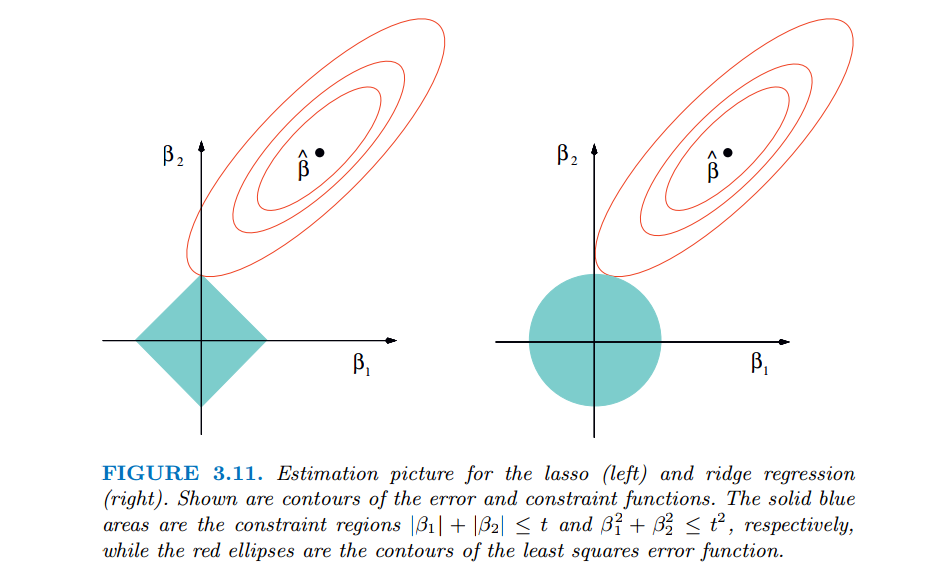
But It is a discrete combination optimization, it’s NP hard…

L1 norm is the optimal convex approximation (so called…) of l0 norm. It makes sparseness and easy to solve meanwhile, so it is widely used, named least absolute shrinkage and selection operator (LASSO)





Actually I don’t know how to prove that l1 norm gives sparseness (though there’s a blog about that which I don’t understand…). But here is a picture about it:



You can see that in l1 norm, the constraint and the best cost function will more likely meet at axes, that gives sparseness, while ridge regression doesn’t. Notice here that the sparseness we’re discussing is a property that means it may come in a great probability.

Forward Stepwise/Stagewise Regression & Least Angle Regression

I’m not sure about a method to give an analytic solution for LASSO, since there’s an absolute operator involved. But we have nice iteration algorithms to give an equal solution, e.g. FS & LAR.

So let’s talk about Forward Stage/Stepwise Regression. The idea is , since we know that some components of X is not (or less) related with Y and we want to distinguish them , then we choose none of them firstly at all (all of  is 0), so the estimated  = 0. The measurement we take to evaluate our model is residual error R = Y - .

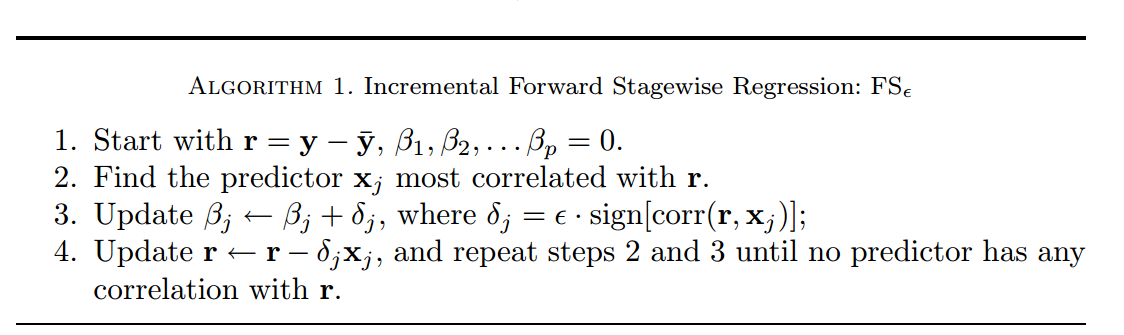
We try to minimize R by changing  , adding the weight of predictors we chose, adjusting s from 0s , that’s why it’s “forward”. In what principle do we choose to add the

weight of a Xi? We simply calculate the correlation coefficient between Xi and R corr(Xi, R), and we choose the Xi = argmax(corr(Xi, R)), and then we update estimated Y and R by adding the specific 

by  \* signal of (corr(Xi,R)) , that’s called a step, where  should be very small. Iterate here till

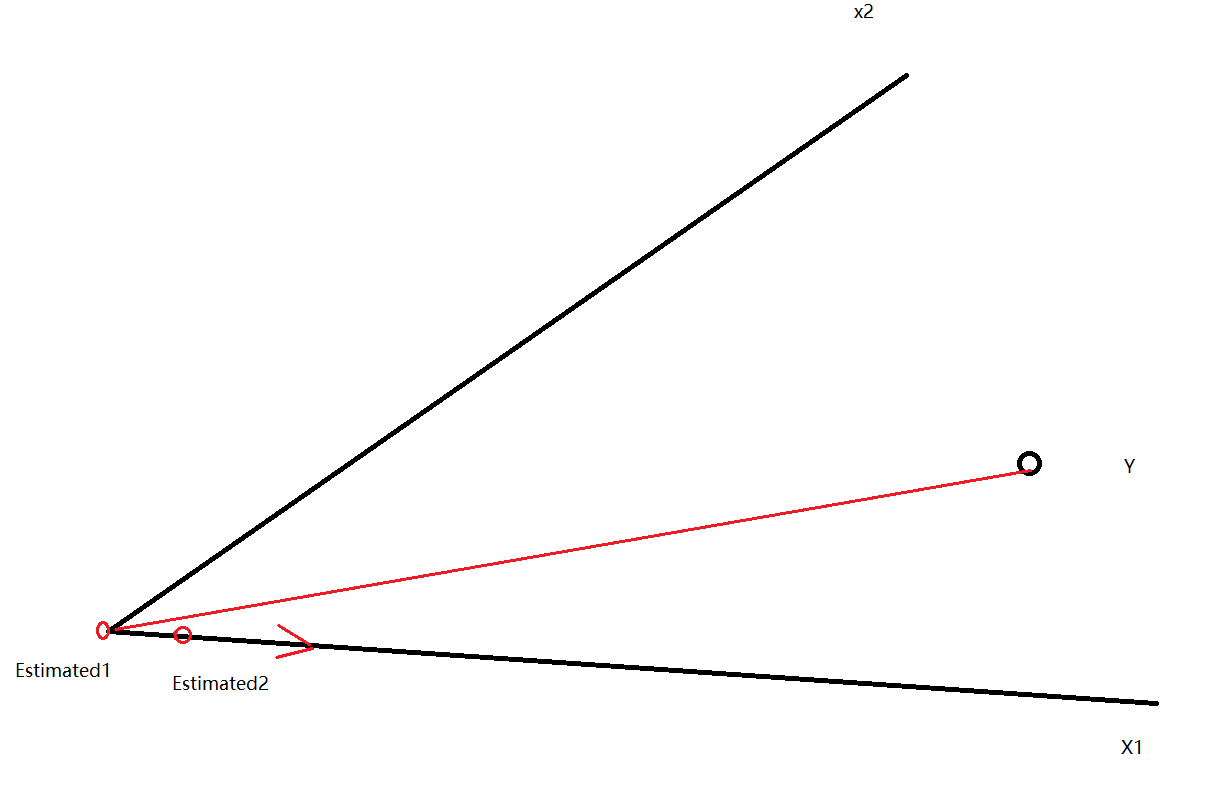
none of the Xi has correlation coefficient with R. Notice that this is an implied but nice constraint.

The procedure is:



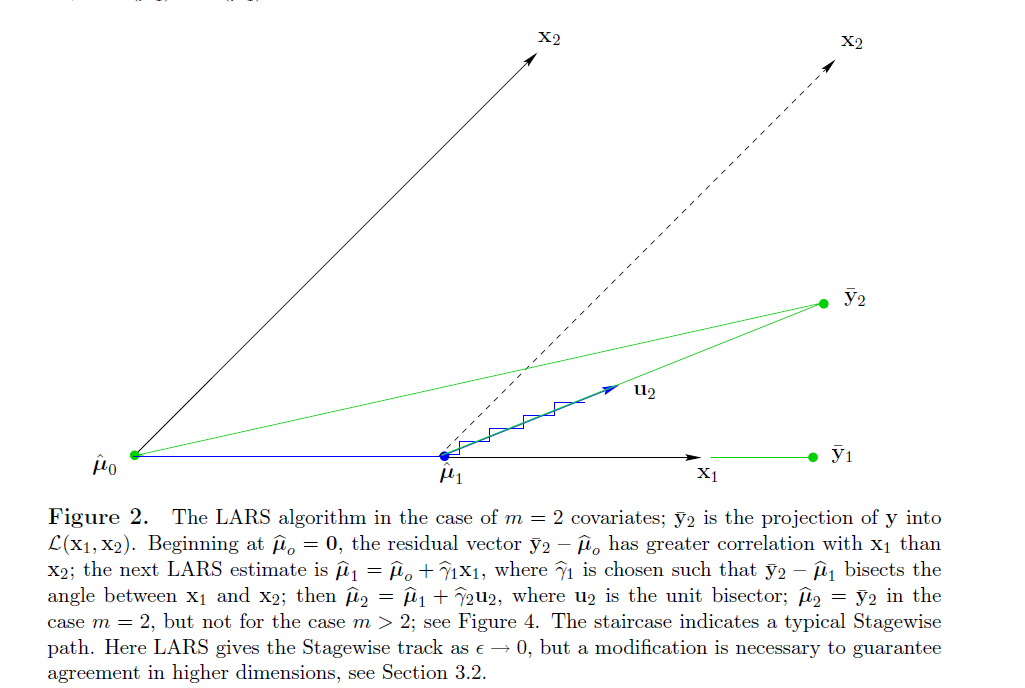
This is called stagewise, and if you claim  as corr(r, Xj) , and regress with the new set again, that’s called stepwise, which means you completely choose one predictor Xj into your set, it has higher speed indeed, however you may miss better solution because of taking a large “step”.

So you can see the point is the conversion of direction: We choose none at first, and approach the solution during the iteration constrained by the correlated coefficient we observe here.



More intuitively in the graph above, at first we find the vector Y - Estimated1 (= 0) is more correlated with x1 than x2, so we move forward at the direction of x1 in a tiny length to, say Estimated2, and then calculate the correlated coefficient again and iterate.

It seems we are doing repeat here: by moving a very small length, we will probably still choose the same Xi next time, in fact we can calculate the maximum length which we can move before changing into another Xi, easily proved to be the length which will make the vector Y - (next estimated) to bisect the angel <X1, X2>. That’s a very nice advanced algorithm: Least Angle Regression (LAR).



Appendices:

1) Forward Stagewise Regression code:

function result = stagewiseRegress(X, Y, epsilon, precision)

[m, n] = shape(X);

% normalization

X = (X - mean(X)) / std(X);

Y = Y - mean(Y);

Beta = zeros(n, 1);

R = Y;

%iteration

while (1)

maxCorr = -1;

maxI = 0;

%choose

for i = 1 : m

corrTemp = corr(R , X(:, i));

if (abs(corrTemp > maxCorr))

maxCorr = corrTemp;

maxI = i;

end

end

%end condition

if (maxCorr <= precision)

break;

end

%update

Beta(maxI, 0) = Beta(maxI, 0) + sign(corrMax) \* epsilon;

R = R - sign(corrMax) \* epsilon \* X(:, i);

end

result = Beta;

end