Email for Rui

Peian Lu

May 15, 2022

1 Lasso

I'm just thinking of the modelling part as an optimization problem. For Lasso, we are trying to minimize the function:

$$LOSS(\beta) = (Y - X\beta)^2 + \lambda|\beta|_1 \tag{1}$$

Do we need to set up the constrain of data that only provides a positive outcome of $\hat{y}$, due to the chemical meaning of the output? From my guess, would it be possible that if we don't count the negative prediction into 'loss function', the estimation of coefficients will return a better fitting under a new/revised criteria? Or, it can express as follows:

$$min : [Y - Relu(X\beta)]^2 + \lambda|\beta|_1 \tag{2}$$

where

$$Relu(a) = \begin{cases} a, & a \geq 0 \\ 0, & a < 0 \end{cases} \tag{3}$$

I'm uncertain about whether this correction will work. Just intuitively speaking, if the intercept of the regression model is extremely negative, it needs the predictors to be large enough to output the positive number. For example, at the first stage(e.g. first 35 hours) of the fermentation process, when all the environmental variables are not that activated, the output of the regression model will be negative, which is not helpful at all. Or I'm imaging another 2-D circumstance the majority of the points lie in the top-right space of the first quadrant with a relation of $f(x) = 10x - 50$(a steep slope), with some other points laying within $x \leq 2$ just fluctuate around 0 (of course, strictly non-negative). For a general fitting, we might get a model of f(x) = 8x - 30, which is a trade-off between those majority and fluctuating points at the initial interval. However, if we adjust our criteria of minimizing the squared error by excluding those negative predictions, then we can get better fitting under the newly defined loss function. Or in other words, we may capture more trends with an inevitable price of the rigid nature of linear regression(it can't always be positive).

2 Implementation

2.1 Coordinate Descent

I think I can try to look for some algorithms like coordinate descent to implement the Lasso by hand(https://myweb.uiowa.edu/pbreheny/7600/s16/notes/2-17.pdf), so that I can manipulate the loss at each run. The drawback is that it will lose the support of convenient frameworks like tidy-verse or mlr to auto-tuning and bench-marking the model. But I think I can set up the CV by hand and it might not be a big problem.

2.2 Source code about glmnet

Digging into the source code in R (https://github.com/cran/glmnet) may help to utilize the existed package, but my first attempts at modification didn't give me sensible results since it obtained a large number of regularization term λ; hence it made most of coefficients 0. The problems I'm encountering may be hard to detect and solve, so perhaps I'll just try to implement my version of Lasso if you think the revision of the loss function will be helpful.

3 Multicollinearity

I'm still confused about how multicollinearity affects the precision of estimation by Lasso. It's said that Lasso in glmnet package conducts the algorithm like Least Angle Regression that starts with an empty model and keeps adding predictors until some criteria are met. We can run the following test in figure 1 that even if we add completely correlated variables into Lasso, it can still come up with an accurate result. Only the order of variables determines which variables are discarded or remain. Because the glmnet scales the data by default, all the $x_{1,2,3}$ become the same after scaling, and then the algorithm updates the coefficients from back to front; hence only the first estimator will be kept, and the rest will be set to 0 earlier. So I'm wondering whether we should select the variables before we feed data into the model and how to select them if necessary. I only have VIF in hand to detect multicollinearity, and not confident about how to set the threshold, and neither don't know how much information may be potentially lost. I guess it is unlike the case of normal linear regression, multicollinearity causes the singularity of the matrix $(X^T X)$, and subsequently, we can't get the analytical solution.

3.1 Including time series variables

Can I understand the points you mentioned early in following way: For a single predictor,

$$Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{1t-1} + \cdot + \beta_n X_{1t-n+1}$$

if we pick n as the 'shifting windows'. So we can use OLS or Lasso to fit the coefficients. For our dataset, many variables are highly correlated with

```
> library(glmnet)
> x1 = runif(100, 1, 2)
> x2 = 2*x1
> x3 = 2*x2
> x_train = cbind(x1, x2,x3)
> y = 100*x1 + 100 + runif(100)
> lasso = cv.glmnet(x_train, y, alpha = 1 )
> coef(lasso);x_train = cbind(x3,x2, x1);lasso = cv.glmnet(
4 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) 1.049752e+02
x1          9.703968e+01
x2          1.560607e-15
x3          .
> coef(lasso)
4 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) 1.049752e+02
x3          2.425992e+01
x2          1.560607e-15
x1          .
```

Figure 1: simple test

themselves as time shifts. So if the multicollinearity can't be solved by Lasso, can we still contain more variables at different times?

3.2 Group effect

I just heard that the elastic net might balance well to maintain the group effect among variables which seems to be suitable in the fermentation process where many variables are contributing collectively rather than separately(and this calls my memory of using SVM since it will fit better in non-linearity case) and the lag effect among chemical variables. Elastic net may be a direction to follow next.

4 Kernel method

I'm learning relevant technics and seeing what we can do if we apply the transformation of the input X. This method can solve the rigid nature of linear regression mentioned before that might force the output to become non-negative. So far, I had just tried it in glmnet once but failed in Cholesky factorization(probably due to multicollinearity). I will go further, as you said before, adding quadratic terms or some transformation. I hope it will give us an interpretable transformation; otherwise, it will make little difference with those 'black-box' methods that can't explain satisfactorily.

5 Summary

I wish at least I express my points clearly. Sorry about my poor English. The modification is not only on the Box-Cox of Y after we've gained a model, but the process of fitting the model before we can calculate $\hat{Y}$. Besides, the prediction seems to already follow a normal distribution, but it goes wrongly at starting period of ferment when the value organic acetic is small. Also there are some other methods I can give a try, but I'm less aware of what will happen. Right now I'm sort of lost, having no ideas how to explore relationships among these data. I hope everything will be solved as I expand my arsenal(learning more knowledge). Thanks for your time reading this emai.