SEM B 2021-2022

CS5487 Machine Learning: Principles and Practice

ASSIGNMENT 1
REGRESSION

Supervised by: Prof. Antoni Chan

Prepared by:

WANG Xuezhen SID: 56199738

Part 1 Polynomial function

a. The implementation can be found in the codes file, in which I use the some seperated files to implement the feature transformation function and 5 regression functions, namely:

   1. Transformation Transform.m
   2. least-squares (LS) LS.m
   3. regularized LS (RLS) RLS.m
   4. L1-regularized LS (LASSO) LASSO.m
   5. robust regression (RR) RR.m
   6. Bayesian regression (BR) BR.m

b. In this section, I train the regression model, predict the test output as well as plot it for visualizing purpose. Regarding the hyperparameters, i.e. $\lambda_{RLS}, \lambda_{LASSO}, \alpha_{BR}$, they are selected by function chosingHyper.m. (assuming we know real outputs of testset, i.e. sampy which is true for this assignment, though in reality, like mentioned in part 3, it is not the case).

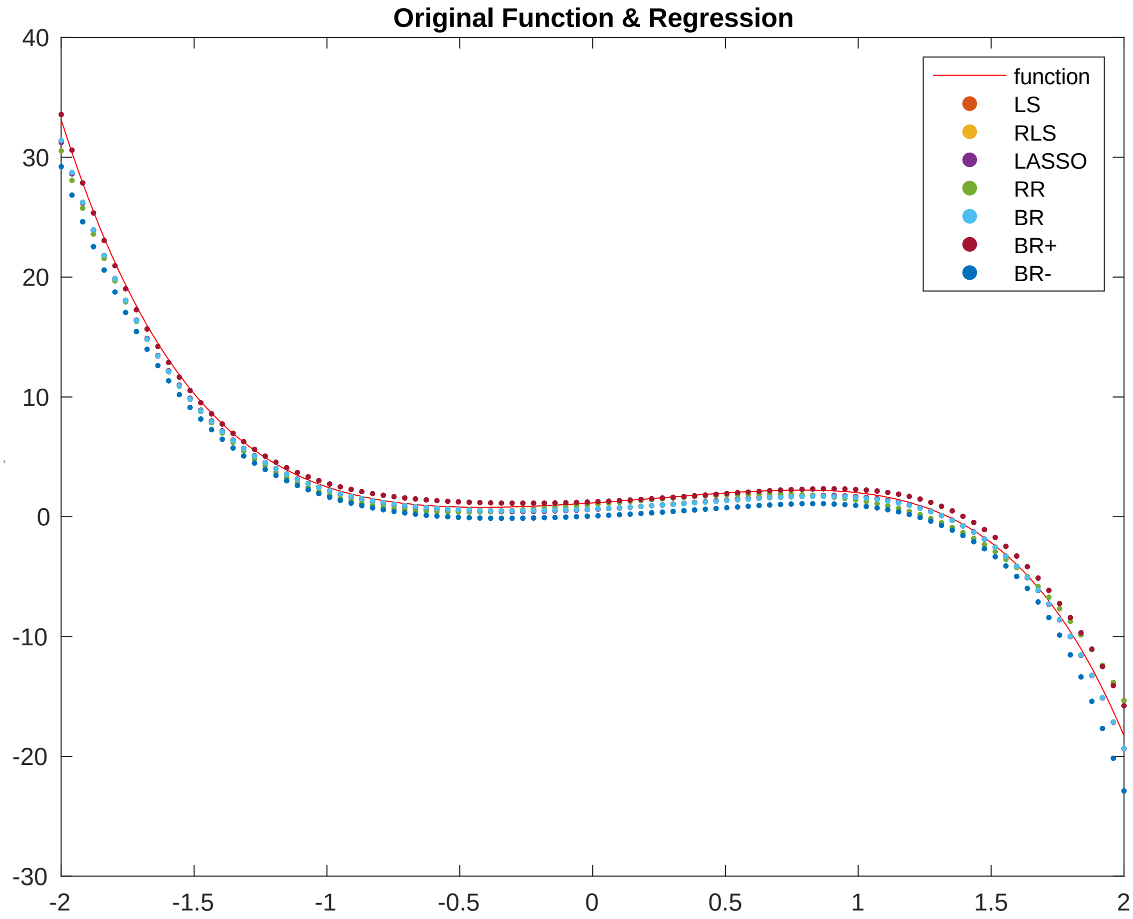

Figure 1 Outputs and original function when using full data set

Notice that the BR+ and BR- are the mean value of predictive value of the bayesian regression plus/minus the standard deviation which is required in the question.

The mean-squared error over all inputs are plotted below:

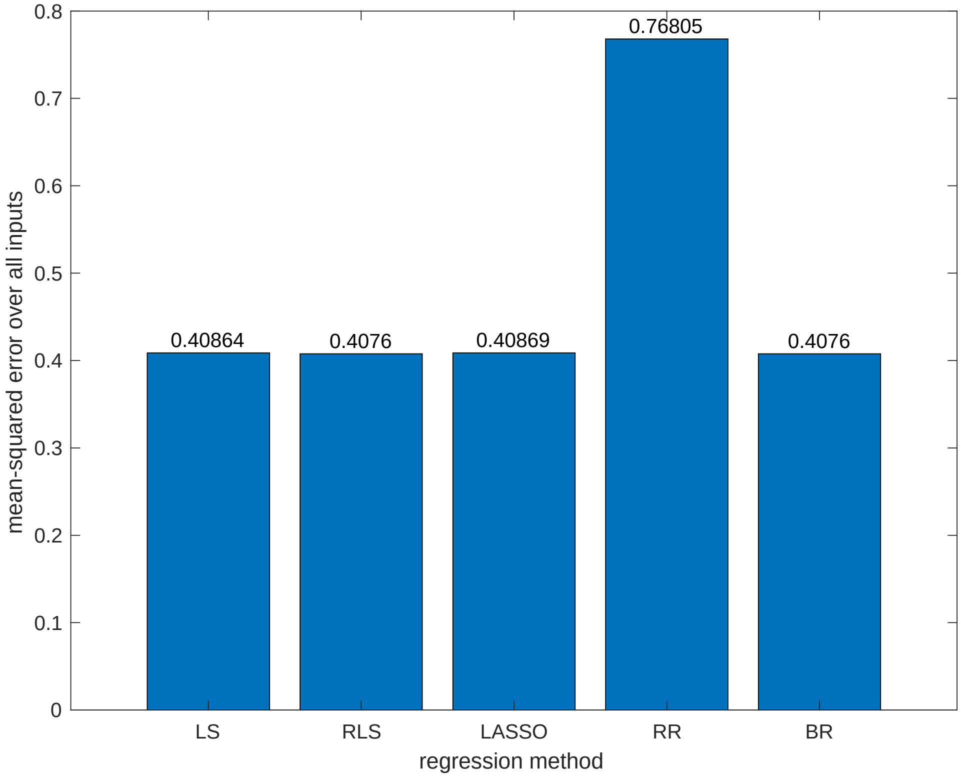

Figure 2 MSE over input size

c. In this subsection, we manually add 3 outlier to the training data and ovserve the behavior of each regression method. ***Notice that for simplity of the figures, we do not plot the BR+ and BR- in this section. though they can easility be done as before (I leave the code block in the attached code file, hence uncommenting it can get BR+ and BR-).*** We first plot the training outputs for several training data of size 80%, 60%, 40%, 20% and 10% of the original training data size. To be more clear, we use unfilled circle in this section to observe the pattern. Then we will analyze based on the plot that which models are more robust with less data and which model tend to overfit.

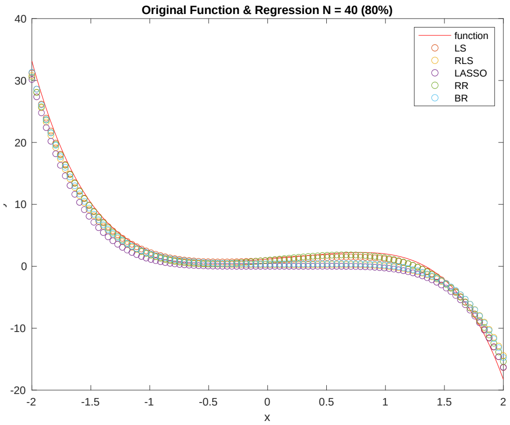

Figure 3 80% Size

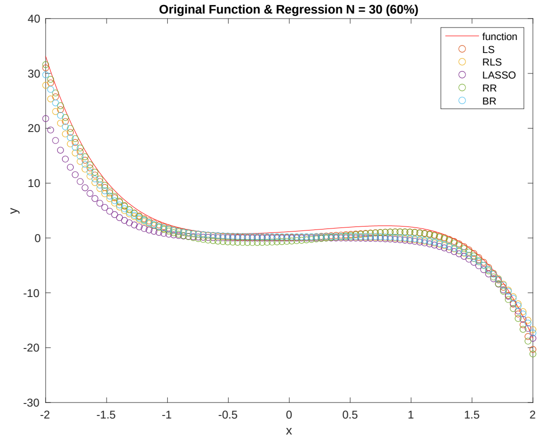

Figure 4 60% Size

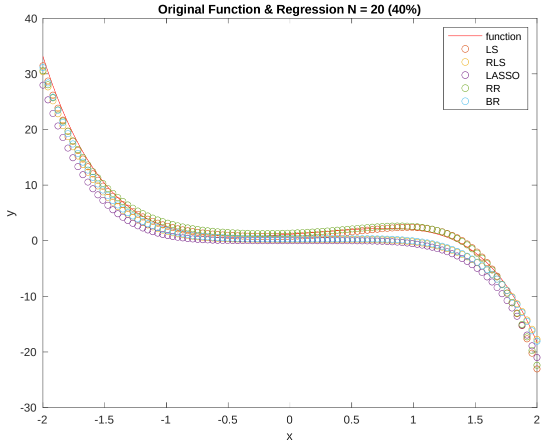

Figure 5 40% Size

From above 3 figures, we may observe that when the training data size are 80%, 60% and 40%, the regressions still fit the original function well. However, when the size decreases more, some models suffer severe overfitting problem.

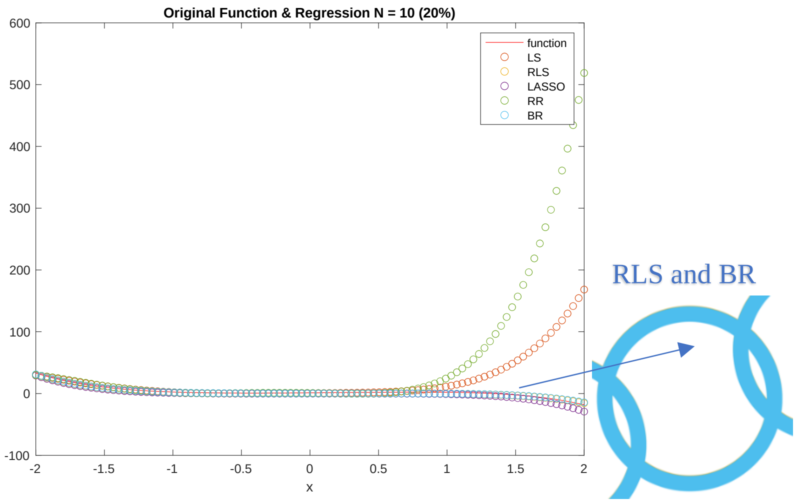

Figure 6 20% Size

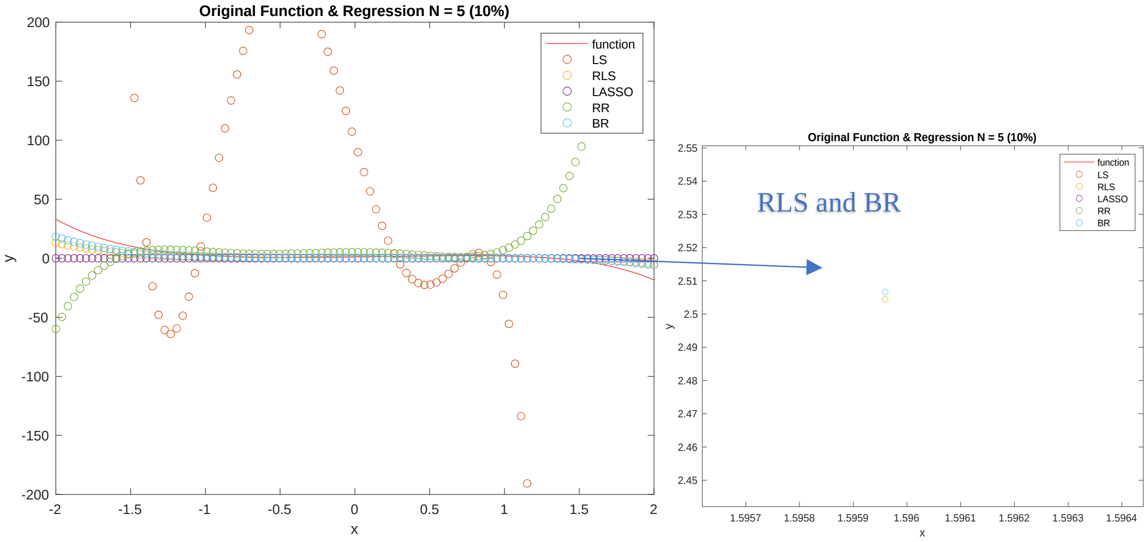

Figure 7 10% Size

As some regression methods fit the function extremely bad, in order to put all outputs into one plot figure, the y scale is huge such that some points belong to different regressions cannot be separated clearly. Therefore, we zoom them in so that we may find which regression method performs similar to the others when there are less data.

By examining the figures, we may conclude from this example that the Bayesian Regression, Regularized Least Square and LASSO are robust with less data and the Least Square and the Robust Regression tend to underfit.

Next, we train 5 additional models using different randomly separated data from sampx and sampy to predict the output correspondingly and average their mean-squared error to get the averaged mean-squared errors for 5 different regression methods.

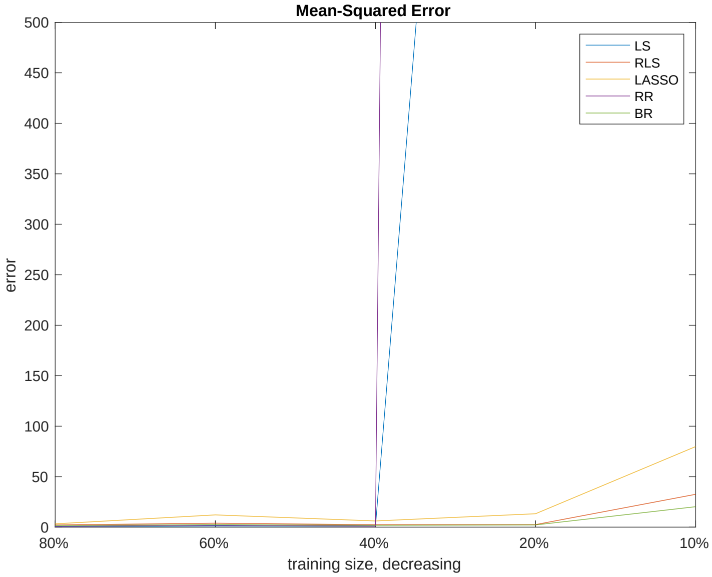

Figure 8 averaged MSE for regression methods

After plotting the average mean-squared errors, some interesting patterns can be found:

1. It can be seen from the plot that, generally, the Bayesian regression and the ridge regression (regularized least square) and LASSO perform much better than the other two methods. From PS 3, we learned that regularized least square has another interpretation for MAP when using gaussian distribution as its prior. Furthermore, LASSO has another interpretation for MAP when using Laplacian distribution as its prior. Hence, we may think that adding a prior for MLE and its equivalent form in regression can be more robust when the training data is less.
2. Intuitively, we analyze why RLS and LASSO can have regularization effects, i.e., reducing the risk of underfitting. they add a hyperparameter λ multiplied by $\theta^T\theta$ or $|\theta|$, hence, these regressions avoid the parameters from rising too high. Alternatively, we may analyze Bayesian Regression, RLS and LASSO from the prior point of view. Namely, all of them have a prior belief added, hence, the final result is not totally determined by the training data. When the training data size is small, this can prevent the underfitting somehow.
3. Trends: From the figures above, we may observe that for almost all regression methods, their mean-squared errors tend to increase when the training data size decreases. Except for LASSO when the data size decreases from 60% to 40% of the original.
4. Another finding is that RLS, BR, LASSO have similar behavior which may be aggregated to that all of them have an equivalent Bayesian representation.

d. In this subsection, we investigate their performance when some outliers present. Specifically, we manually add 3 outliers which are significantly biased to observe how the regressions perform.

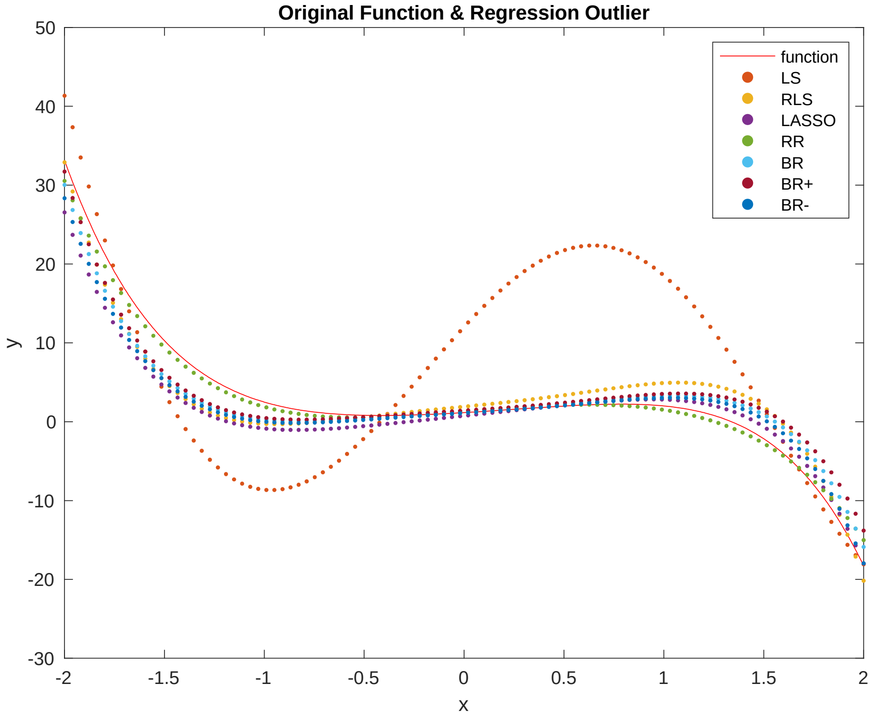

Figure 9 Outputs and original function when using full data set with outliers

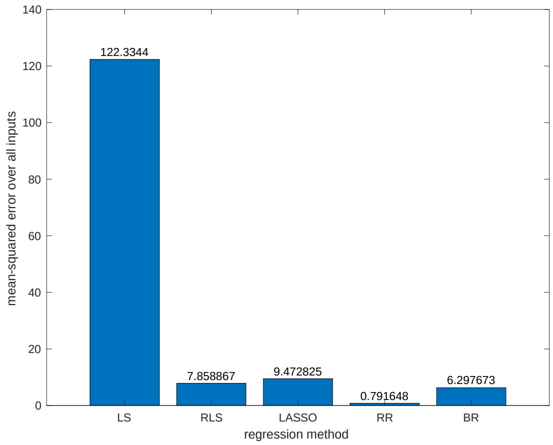

Figure 10 MSE over input size (added outlier)

From the above 2 figures, we can easily see that there is a huge difference between RR and the other 4 methods. Hence, robust regression is most robust to the outliers.

Reasons:

1. L1-norm assigns equal weight to each sample data. However, for other regressions, the difference between the predicted output and the real output is squared which causes higher weight for those predicted output which are more biased from the real value. Therefore, some outliers can significantly collapse the model.
2. In terms of its corresponding MLE estimation, the Laplacian distribution has “fatter” tails compared with gaussian distribution. Therefore, it gives the likelihood that some data are outliers.

Besides, RLS, BR and LASSO also have some robust features compared with LS.

Reasons:

1. The reason might be that they all have a Bayesian way to represent the mode. BR weights over all possible θ, and the other two also have a prior belief put on them.

We may also observe that LS is most likely to overfit.

Reasons:

1. For LS, there is no regularization factor as all. All its objective is to minimize the squared error. Hence, it is very sensitive to the outlier since several of them can contribute a lot to the parameter estimation.

e. In this subsection, we will make the model more complex, i.e., to increase its order to K=10. We will analyze why some models are robust to overfitting compared with other models by investigating the parameter values.

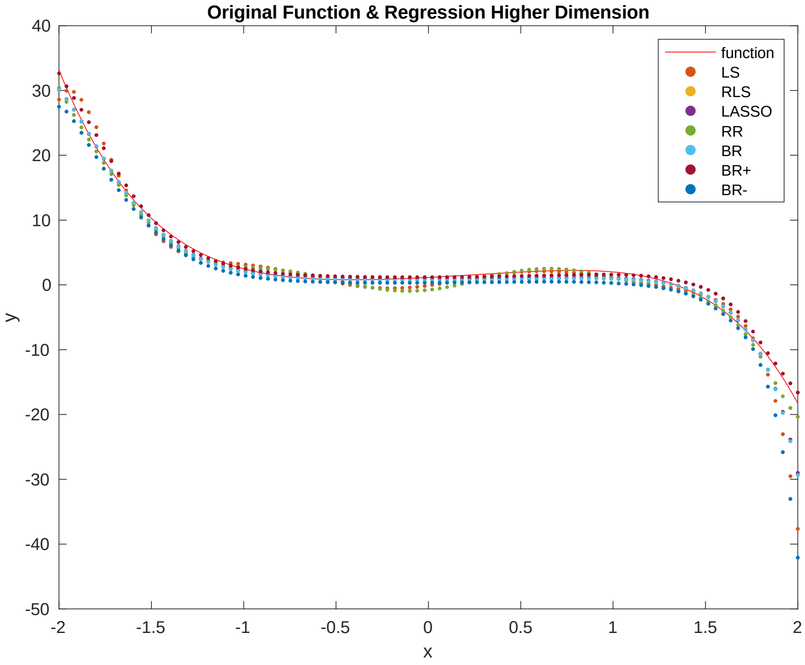

Figure 11 Outputs and original function when using full data set with high order

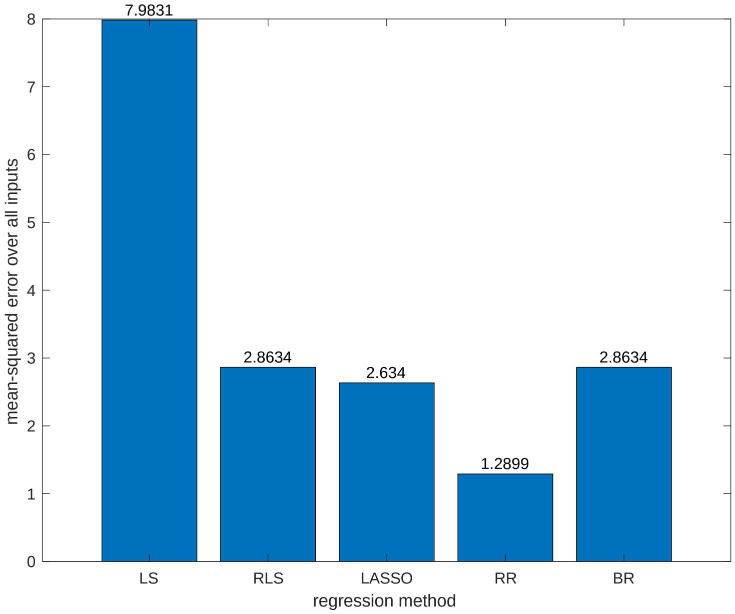

Figure 12 MSE over input size (high order)

From the first figure, i.e., Figure 11, we conclude that LASSO has the best estimation as it almost obeys the original trend which means that it does not fluctuate up and down like the other regression such that RLS, BR and RR.

From the second figure, i.e., Figure 12, we may conclude from another perspective that RR, LASSO, BR, RLS have good performances. Notice that this perspective purely concentrates on the mean-squared-error which overlooks the “trends”.

We investigate their θ value as well as the true theta value to see why they are robust.

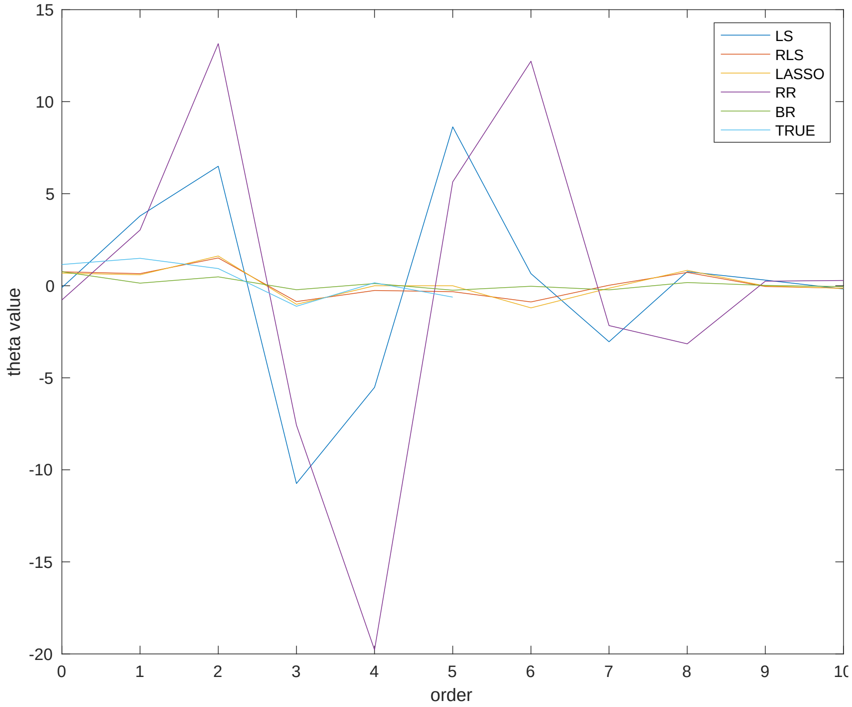

Figure 13 Theta for order 0 to 10

We can learn from the above figure that the curves for LASSO, RLS and BR tends to fit the original θ' curve. For RR, though its mean-squared error is small, it is very different from the true θ' value which may explain why we see it fluctuates a lot around the real function as some of its large coefficients drag it up and down.

We all know that **Low bias and low variance will give a balanced model, whereas high bias leads to underfitting, and high variance lead to overfitting.** Inspecting the coefficients, we can see that Lasso, RLR and BR have shrunk the coefficients, and thus the coefficients are close to zero which have the similar scale as the real coefficients. On the contrary, LS and RR still have a substantial value of the coefficient which means that their generalization are very poor, as they will cause high variance. By adding a penalty and decreasing the coefficients, the strategies provide a well-fitting model. To prevent overfitting, a proper balance of Bias and Variance must be maintained.

Part 2 A real world regression problem – counting people

1. In the first section, we set the transformation function to be $\phi(x) = x$, and use different regression methods to prefict the result.

   We first display the mean-squared error and the mean-absolute error to see which regression method tends to work the best.

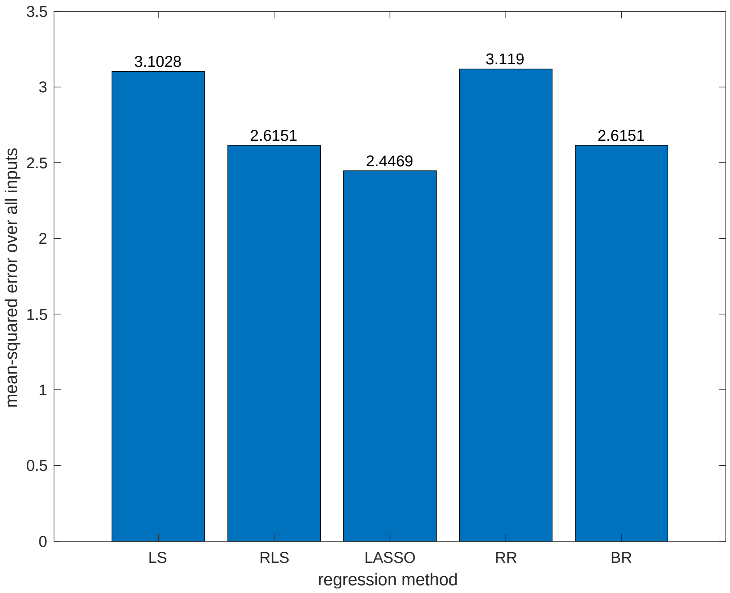

Figure 14 mean-squared error

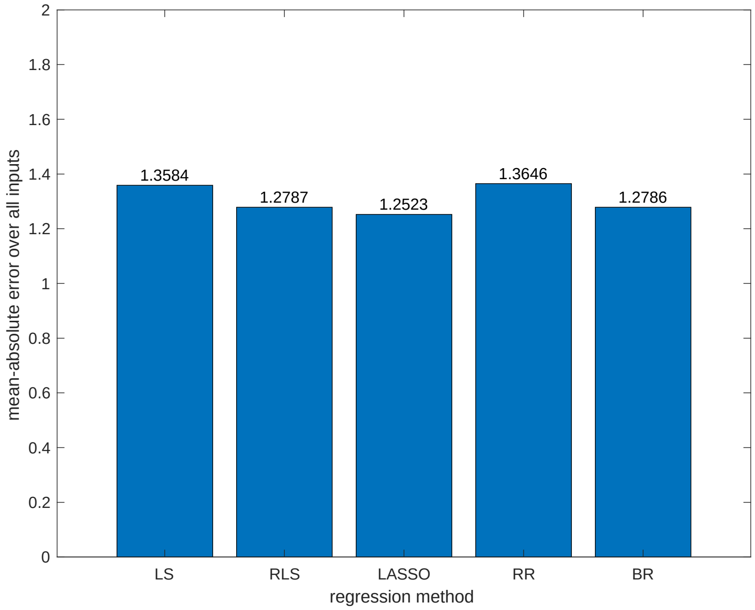

Figure 15 mean-abs error

From the above 2 figures, we may observe that LASSO tends to work the best. This may indicate that there are some features which are completely useless and setting them to zero can increase the accuracy.

Then we plot the predicted value and the real value and observe some interesting patterns.

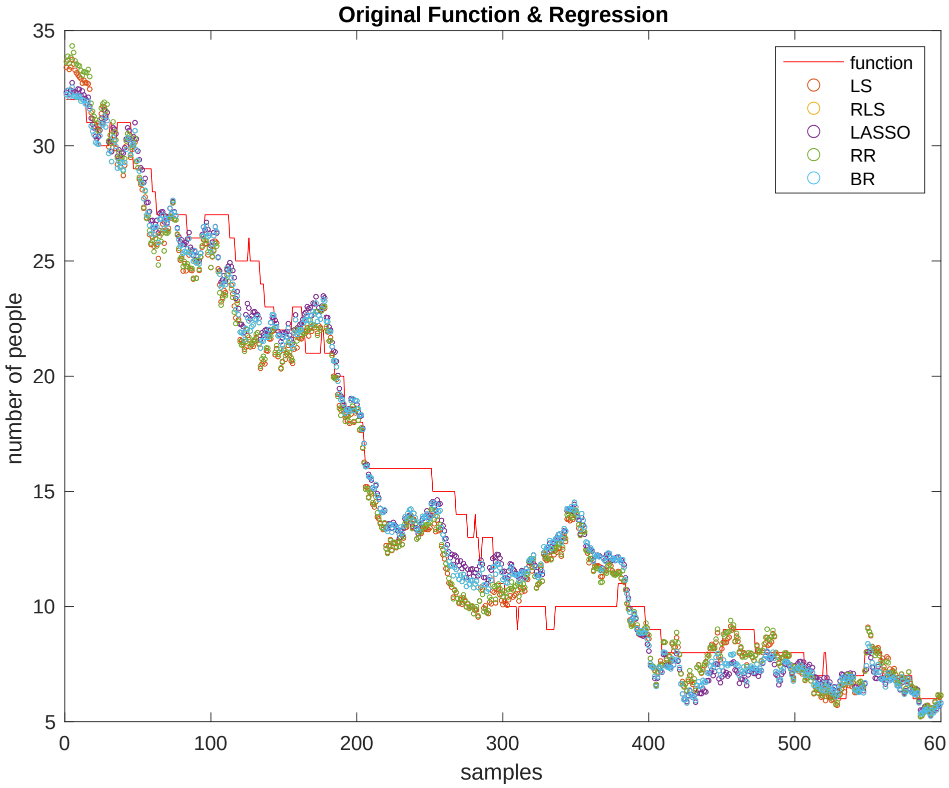

Figure 16 Outputs and original function when using full original transformation

Findings:

- The green markers and the red markers have the similar output values which is indicated in the plot that those markers are close to each other. On the other hand, the yellow markers, purple markers and blue markers have the similar output values which is indicated in the plot that those markers cluster together. The reason behind may be that LS and RR corresponds to MLE for some distributions, whereas the other three corresponds to MAP for some different prior distributions. Hence, they are two different families.
- It can be seen from the above figure that all the predictive models behave well when the number of people is huge or small, nevertheless, when the number of people is in the middle, i.e., around 10 to 15 and 20 to 25, all models have predicted values far from the real value.
- It is easy to see that the yellow markers are not clear in the above picture, if we zoom in it a little bit, i.e., the below picture, we may find that the yellow markers are very close to the blue markers. This makes sense, as in terms of the MAP represeantation, BR's posterior distribution is exactly what we apply MAP to in order to get the equivalent RLS.

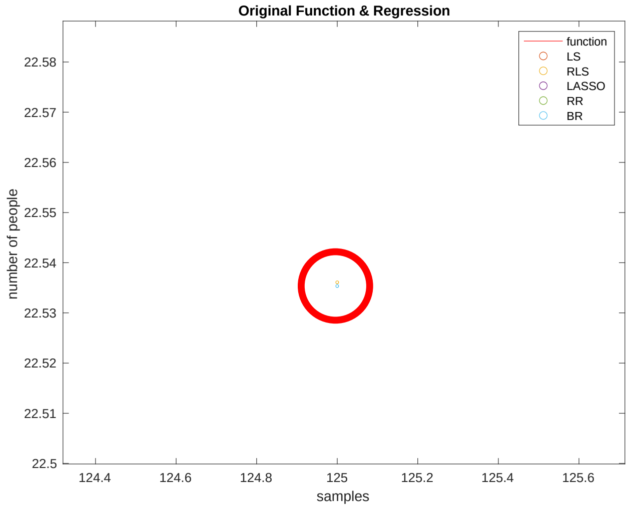

Figure 17 zoom in

2. In this subsection, we use some other transformation function to see if we can get better result.

   i) We first invstigate when $\phi(x) = [x_1, \ldots, x_2, x_1^2, \ldots, x_9^2]^T$, and we can see from the MSE and the MAE that the performance is better compared with when we use $\phi(x) = x$.

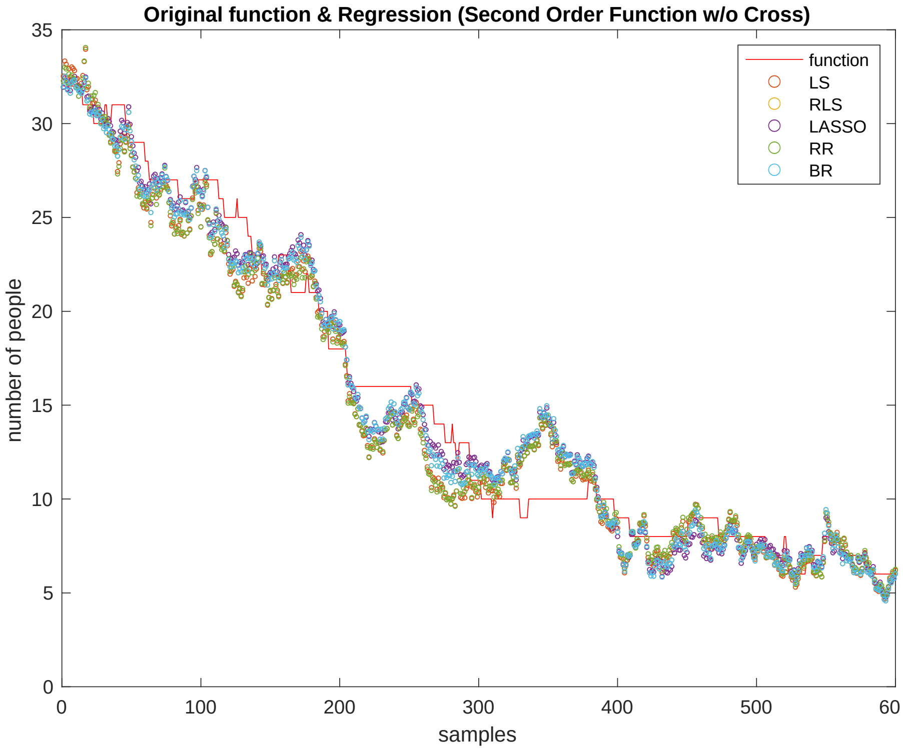

Figure 18 Outputs and original function when using second order no cross term transformation

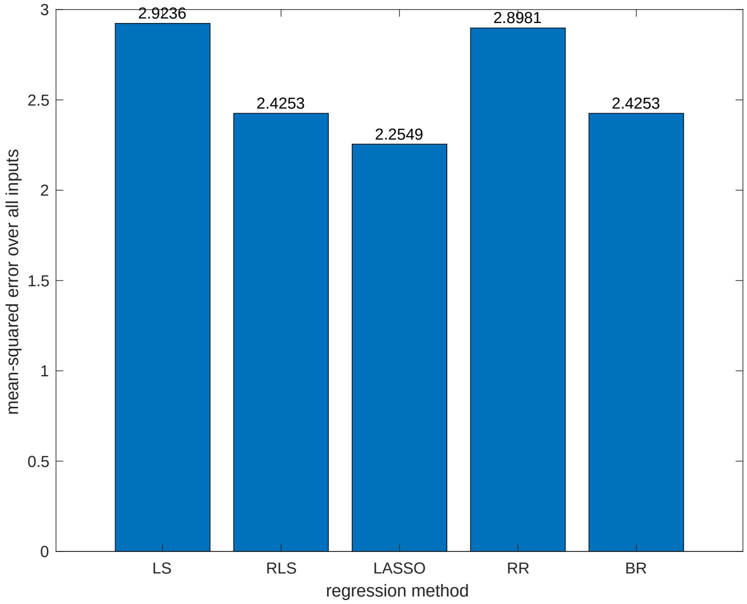

Figure 19 mean-squared error

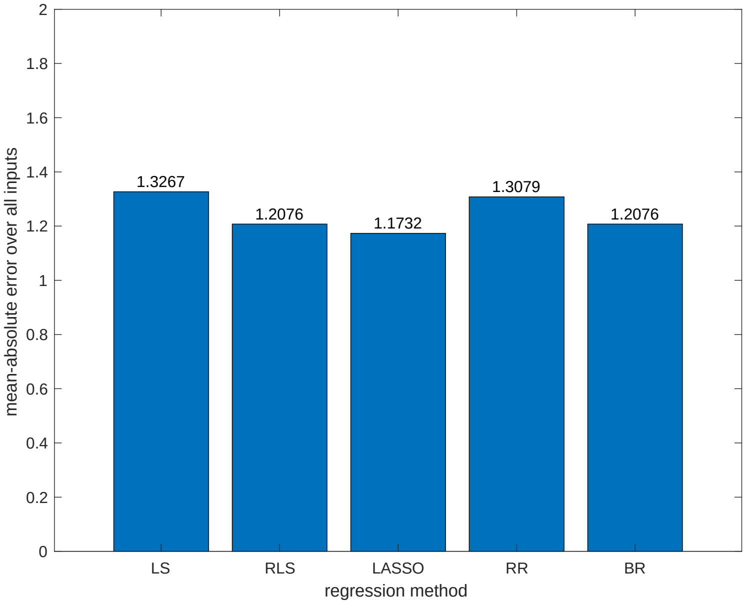

Figure 20 mean-absolute error

ii) We then invstigate when $\phi(x) = [x_1, \dots, x_2, x_1x_1, x_1x_2, \dots, x_9x_9]^T$, and we can see from the MSE and the MAE that the performance is worse compared with when we use $\phi(x) = x$ and without the cross term. Hence, it may suffer from the problem of overfitting.

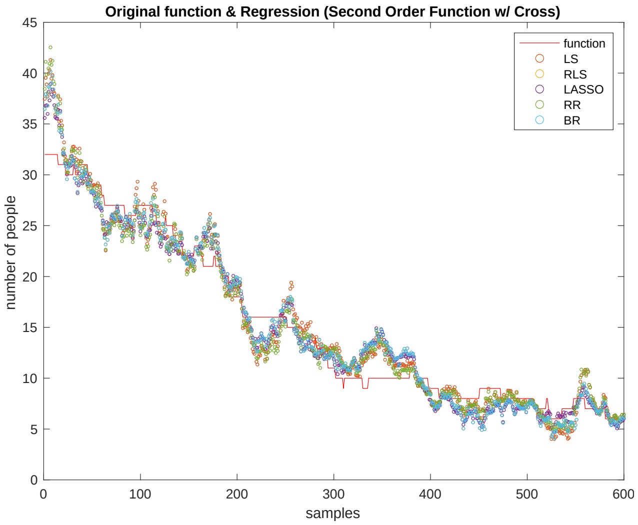

Figure 21 Outputs and original function when using second order with cross term transformation

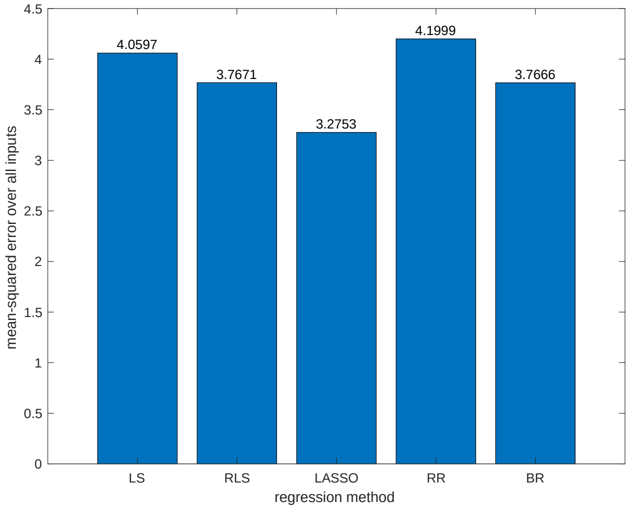

Figure 22 mean-squared error

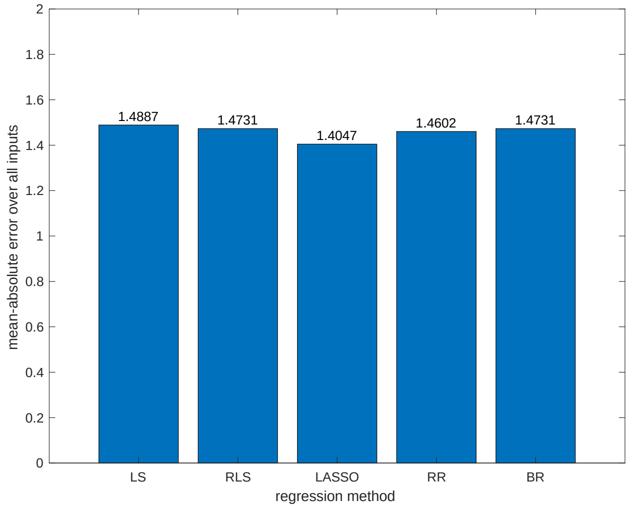

Figure 23 mean-absolute error