Assignment Project1

Part I

(a).

The implementations of the 5 algorithms for the K-th order polynomial are done in Python. The source code is uploaded.

In the package, the ‘LeastSquare.py’ contains the process for solving 5 regression algorithms; the ‘Prediction.py’ contains the process for predicting values given *samp\_x*, and ‘DrawPlot.py’ contains the process for plotting some figures.

(b).

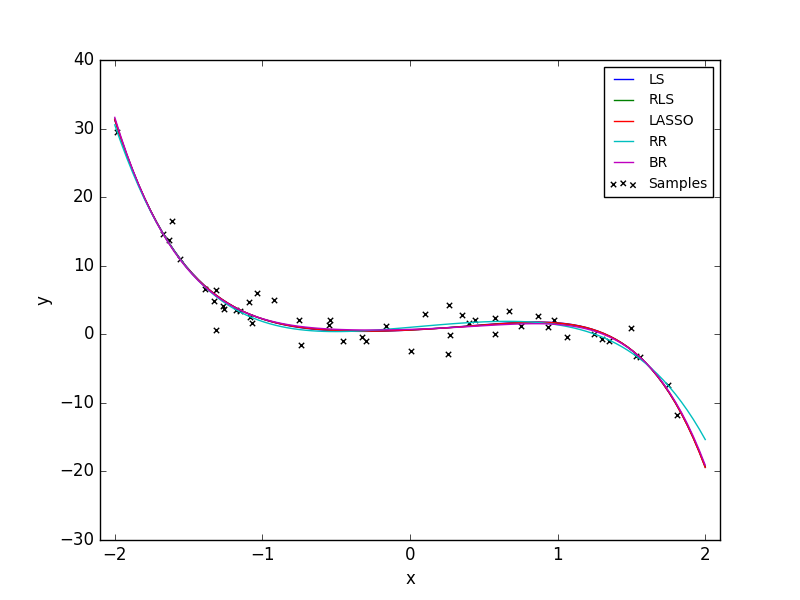
Figure 1 reports the results of the estimated function using *poly\_x* as input (For solving LP, I use Python scipy.optimize.linprog package and for solving QP, I use Python cvxopt package.).

Figure 1. Five estimated polynomial functions along with the sampled data.

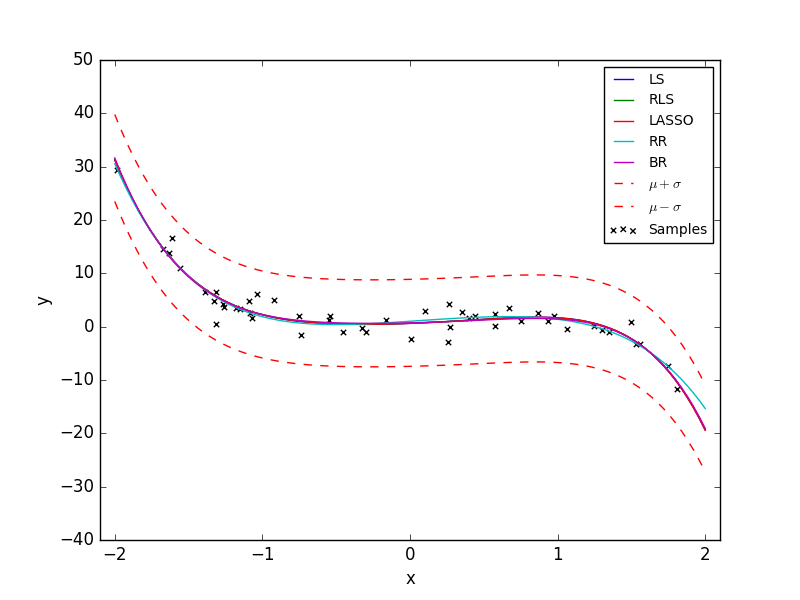
From Fig.1, one can observe that all 5 regression algorithms could obtain functions that fit the sampled data well, however there are some minor differences on the shapes of the estimated polynomial functions.

Figure 2. Standard deviation around the mean for BR.

Figure 2 shows the standard deviation around the mean (the red line is the value of \mu, and the blue lines is the value of \mu + \sigma and \mu - \sigma).

Table 1 presents the how MSE is related by the value of hyper parameters in RLS, LASSO and BR. We could observe that with the increase of λ, the MSE of RLS first decreases and then increases. The MSE of LASSO increases along with λ, whereas the increase of α reduces the MSE of BR. Therefore, I choose λ=0.5 for RLS, λ=0.2 for LASSO and α=2.5 for BR.

Table 1. Different settings on hyper parameters.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | 0.2 | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 |
| RLS | 0.4079 | 0.4076 | 0.4086 | 0.4114 | 0.4156 | 0.4210 |
| LASSO | 0.4176 | 0.4346 | 0.4746 | 0.5147 | 0.5191 | 0.5213 |
| BR | 0.8562 | 0.5579 | 0.4591 | 0.4320 | 0.4210 | 0.4156 |

Table 2 shows the MSE between the learned function outputs and the true outputs, averaged over all input values in *poly\_x*. Apparently, the RR obtains the greatest MSE, which indicates that its ability to fit the function is the worst among 5 algorithms in this samples.

Table 2. MSE between the learned function outputs and the true outputs.

|  |  |
| --- | --- |
| Methods for regression | MSE |
| RR | 0.4086 |
| RLS | 0.4076 |
| LASSO | 0.4176 |
| RR | 0.7680 |
| BR | 0.4156 |

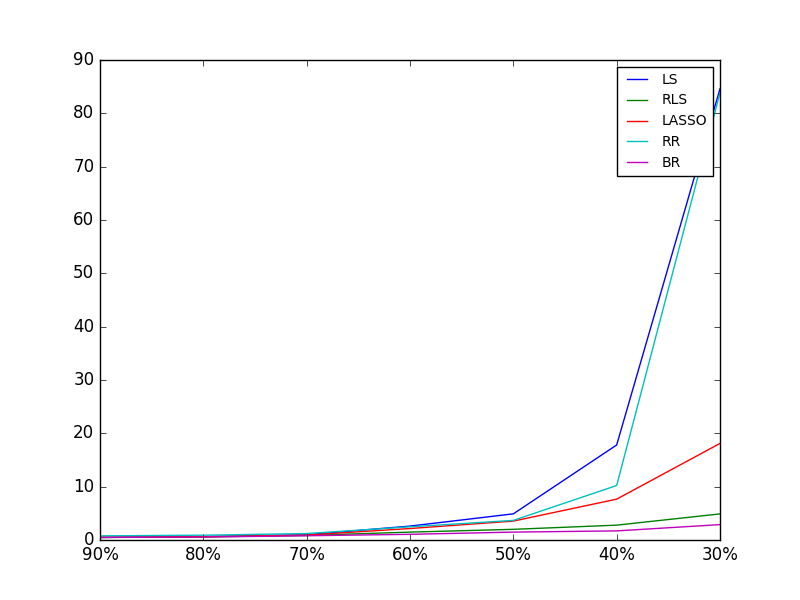
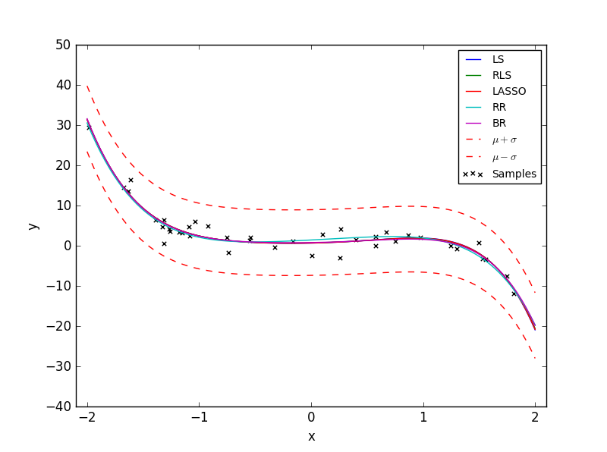
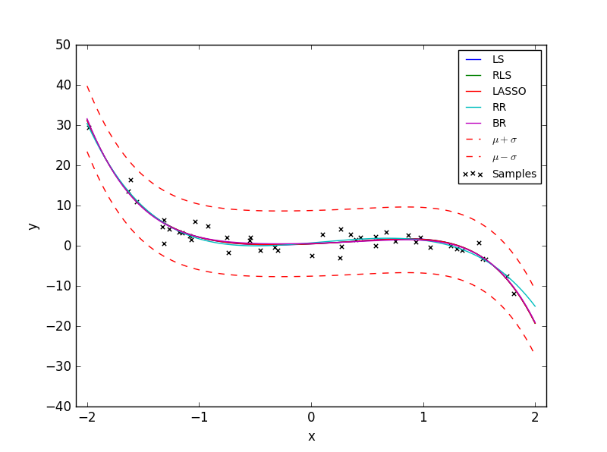
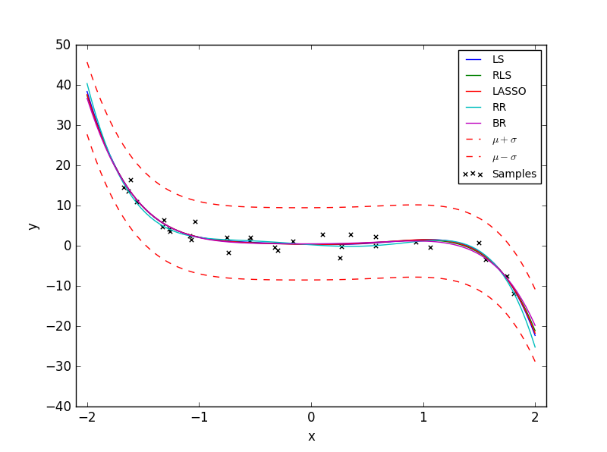
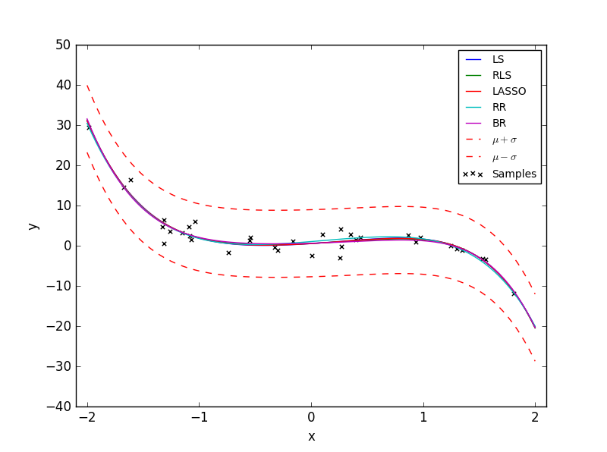
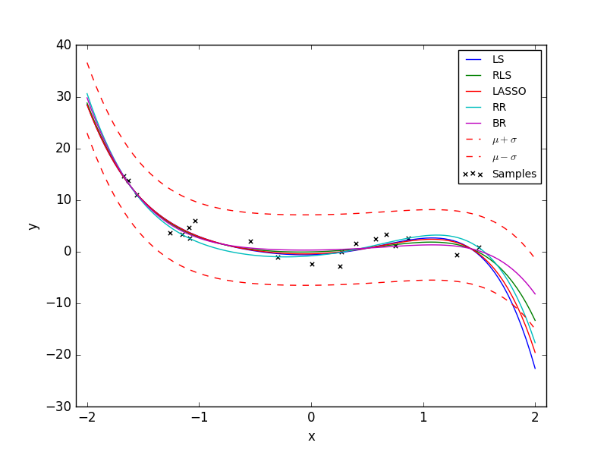
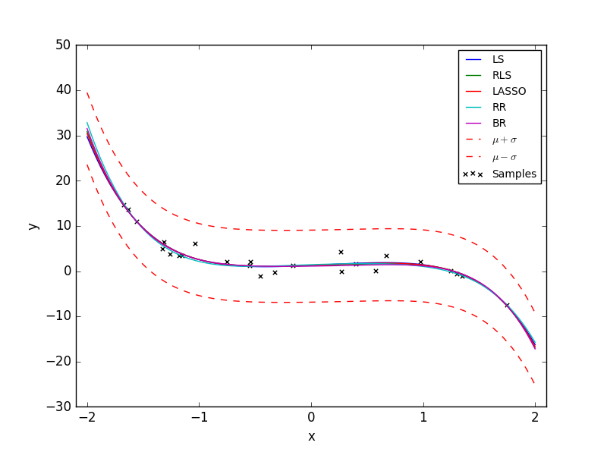
(c)

Figure 3. Averaged MSE with using 90%, 80%, 70%, 60%,   
50%, 40% and 30% of sampled data.

Figure 3 shows the averaged MSE after 100 iterations using 90%, 80%, 70%, 60%, 50%, 40% and 30% of sampled data, respectively. Apparently, MSE becomes less when the size of the training set becomes larger for all 5 regressions. When there are very less sample data, the MSE of RR and LS are extremely greater than that of other three methods. Therefore, it seems that RR and LS are easier to be overfitted and the other 3 regressions are more robust when less samples are given.







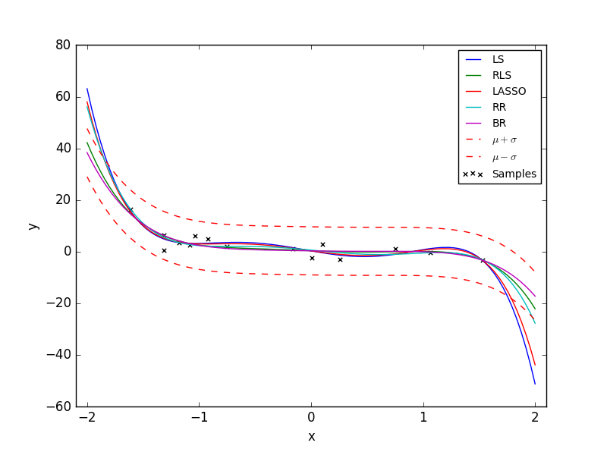


Figure 4. The estimated functions using 90%, 80%, 70%, 60%,   
50%, 40% and 30% of sampled data.

From Figure 4, we could observe that when only 30% sampled data are used for training, the estimated functions of RR, LS and LASSO are in a complicated shape which change the directions a few times. Although all 5 regressed functions fit the sampled data very well, they could not perfectly predict the test data because of overfitting problem.

(d)

We randomly chose 10%, 20%, 30% points in samp\_y and add a random large number which satisfies a uniform distribution.

Table 3. MSE with 10%,20% and 30% noisy data.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 10% | 20% | 30% |
| LS | 8.361 | 105.753 | 360.135 |
| RLS | 7.308 | 95.528 | 354.105 |
| LASSO | 8.221 | 105.26 | 359.890 |
| RR | 0.800 | 0.9093 | 181.121 |
| BR | 5.881 | 80.746 | 340.64 |

Apparently, when the percentage of the noisy data is larger, the MSE of all regressions are greater as well. However, the RR is the most robust one because from 10% noisy data to 20% noisy data, it merely has no increase on MSE whereas the other 4 methods have an extreme increase on MSE. Even when there is 30% noisy data, the RR still outperforms other 4 methods.

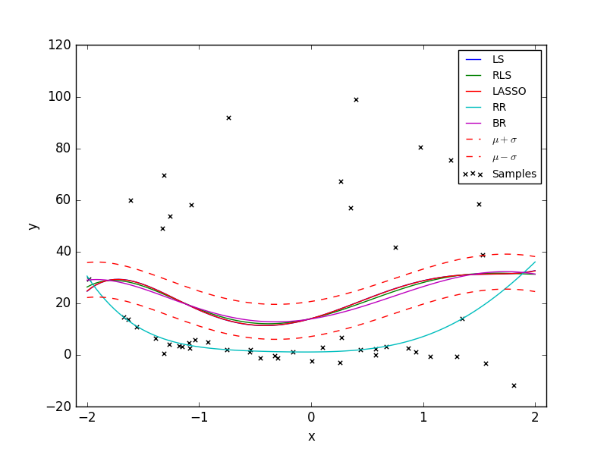
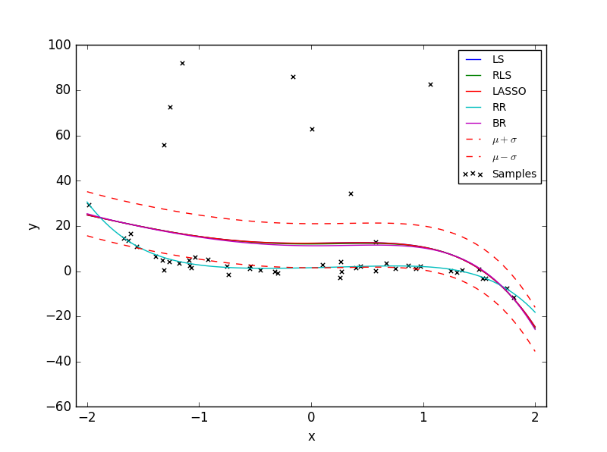
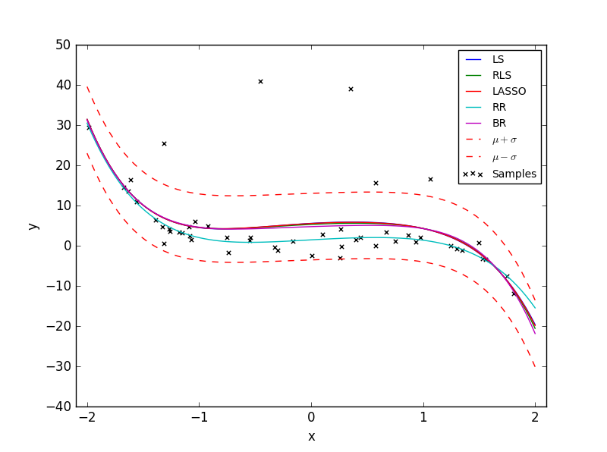


Figure 5. Given 10%, 20% and 30% noisy data, the estimated value function

The most sensitive methods are LS, RLS and LASSO according to the results. All these 3 methods utilize a quadratic objective value function to optimize the parameters. When the number of outliers increases, the objective functions have to be fitted to the outliers and the predicted perform could be worse. On the contrary, the RR method utilizes L1 norm so that the outliers have less effect on the objective function. As for BR method, when the number of outliers is not too many, it still could estimate parameters which are partly bad (in the second figure, most of sampled data are in the region between \mu + \sigma and \mu - \sigma). However, when the number of outliers increases, the estimation of parameters could be affected by the outliers so that there are no sampled data are in the region between \mu + \sigma and \mu - \sigma.

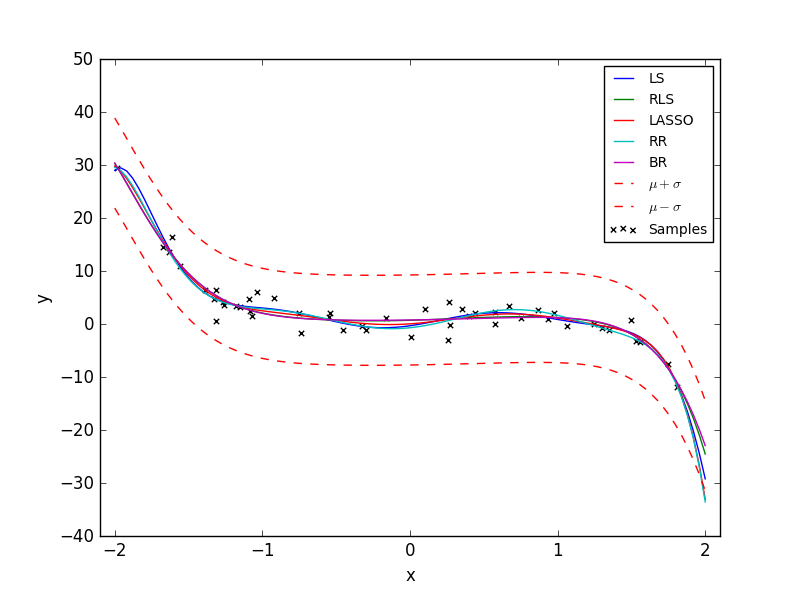
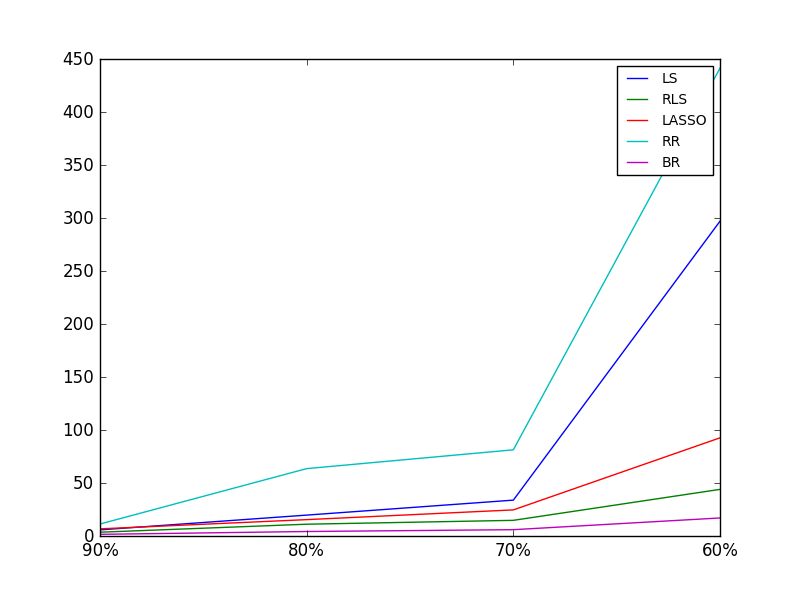
(e)

Figure 6. 9th order estimated polynomial function.

When we use 5 methods to estimate a higher order polynomial function, the estimated functions are seemed to be more complicated. To see the performances of the methods, we use 90%, 80%, 70% and 60% sampled data and repeat 100 times, then we calculate the averaged MSE.

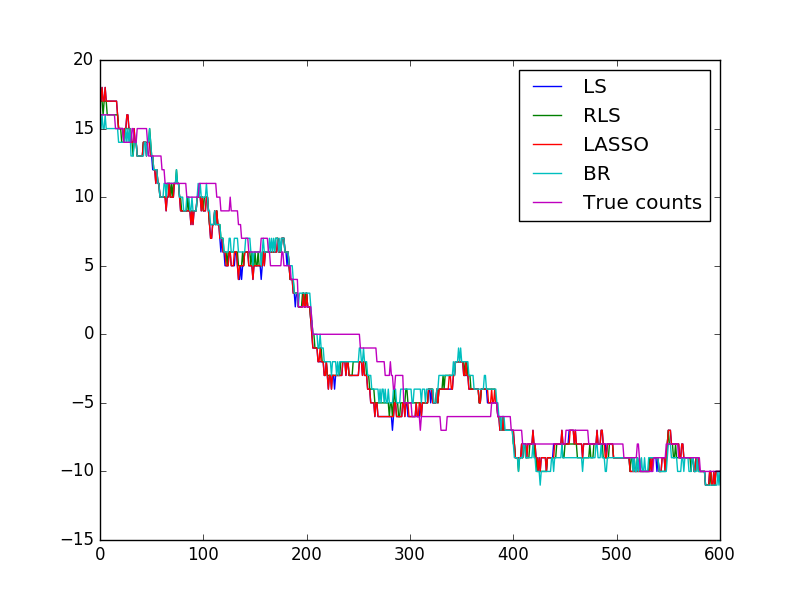
Figure 7. Averaged MSE for different sample sizes.

It seems that LASSO, RLS and BR methods are more robust than other methods because the MSE are much lower than that of other methods. In Figure 6, the functions obtained by RR and LS seems the most overfitted. The reason is that both of RR and LS have no regularization parts. When the number of parameters to be estimated increases, to obtain a less objective value, the model has to be more complicated to fit more sampled data.

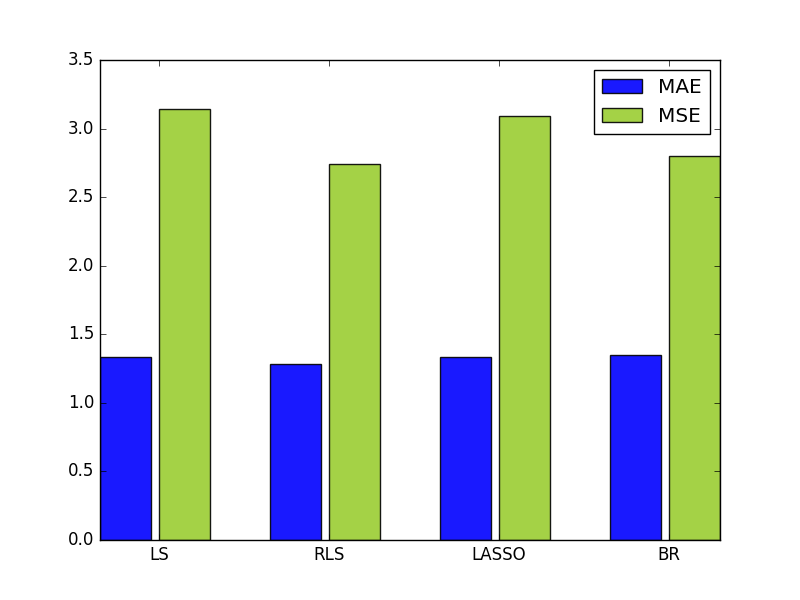
Part II

(a)

We use LS, RLS, LASSO and BR to predict the counts in a picture. The parameters for RLS, LASSO and BR are respectively λ=0.5, λ=0.2 and α=2.5. The predicted counts and the true counts are compared in figure 8. (The reason why I leave out RR method is that the RR method has no solution when I solve the linear program.)

Figure 8. The comparison between true counts and predicted counts.

We could observe that the regression methods are good tools in that problem. The predicted counts almost coincide with the true ones. Figure 9 reports the MAE and MSE of each regression. In terms of MAE, RLS seems to have the least errors and other three methods obtain similar results, i.e., MAE is around 1.33. For MSE, both RLS and BR have lower errors compared to the other two methods.

Figure 9. MAE and MSE.

(b)

I wonder (1) if the more complicated model could obtain more precise model, (2) and if so, will the randomly generated ‘feature’ affect the precision? For (1), we consider more features by adding squared elements, cubed element and interacted elements (x1x2, x2x3,…,x8x9) to feature vectors. The new feature vectors are now have more dimensions, thus more complicated model will be obtained. The results are presented in 10 to 12.

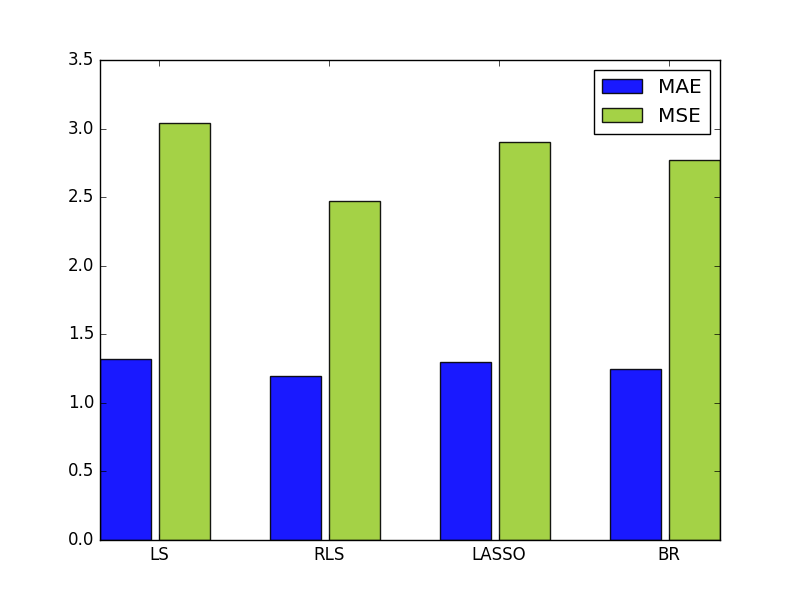
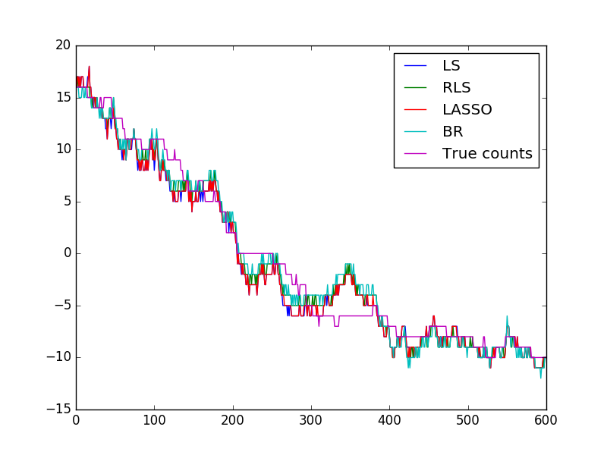


Figure 10. Squared features.

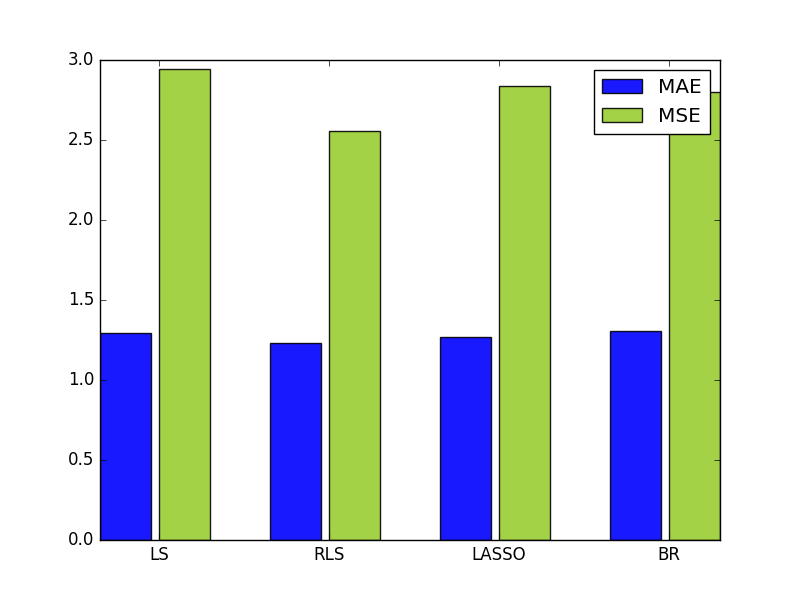
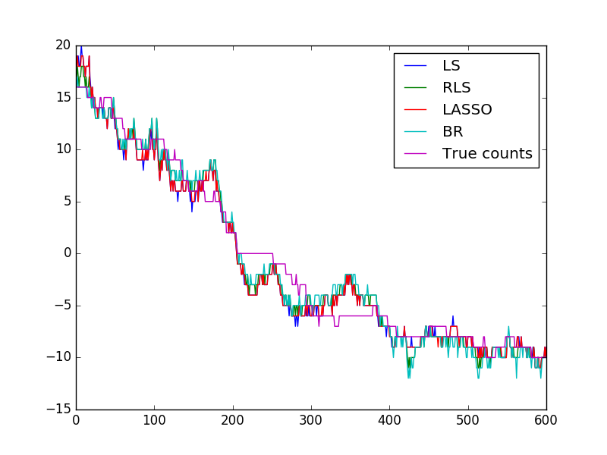


Figure 11. Cubed features.

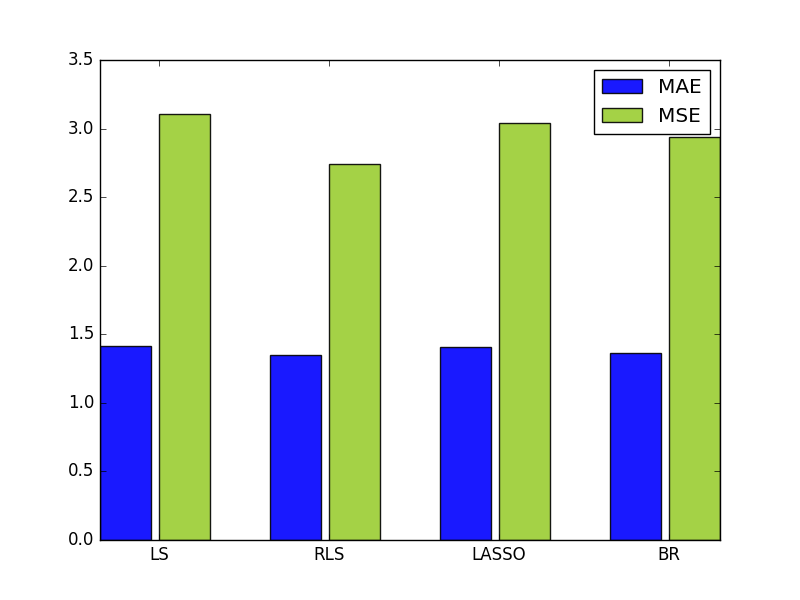
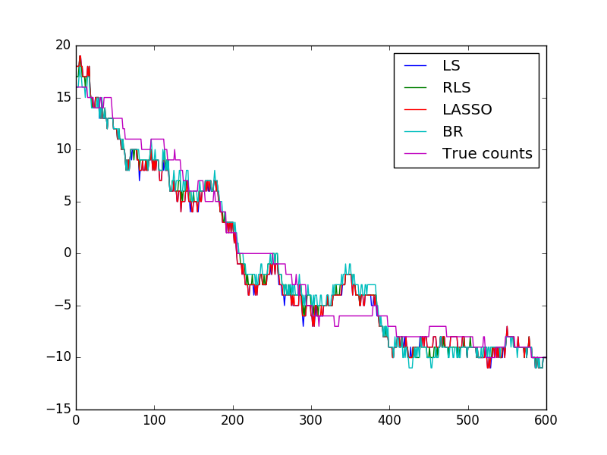


Figure 12. Interacted features

For (2), we randomly generate nine features in a uniform distribution and obtain the predicted counts. Totally, we obtain 5 feature vectors for each regression method. Figure 13 presents the predicted counts and MAE\MSE.

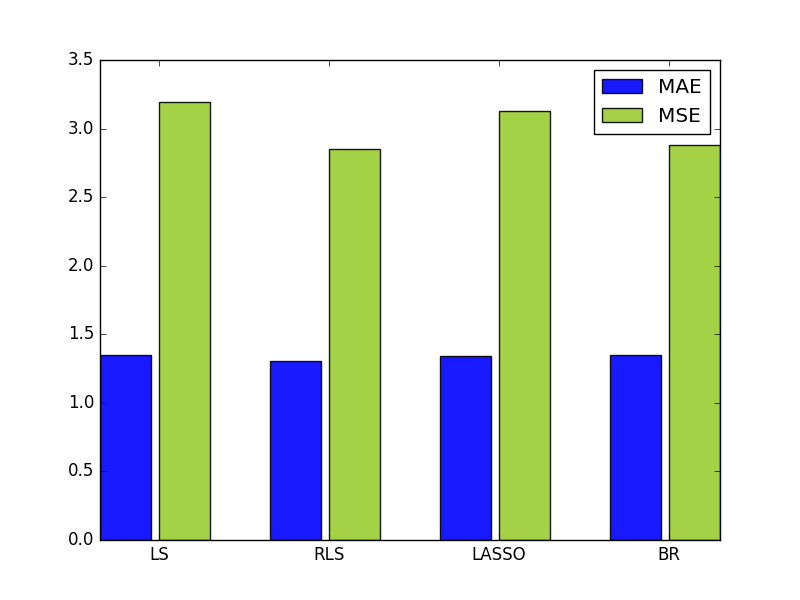
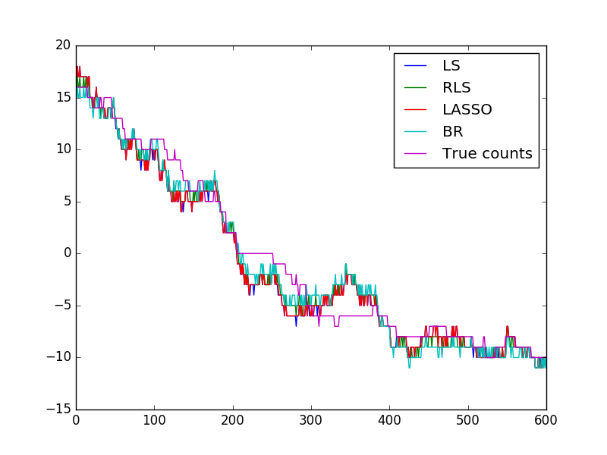


Figure 13. Randomly feature.

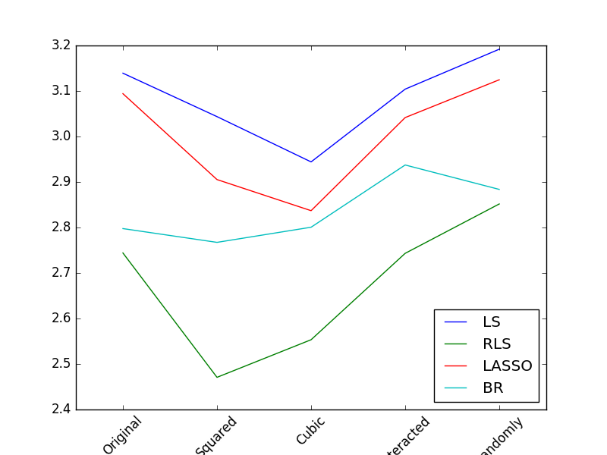
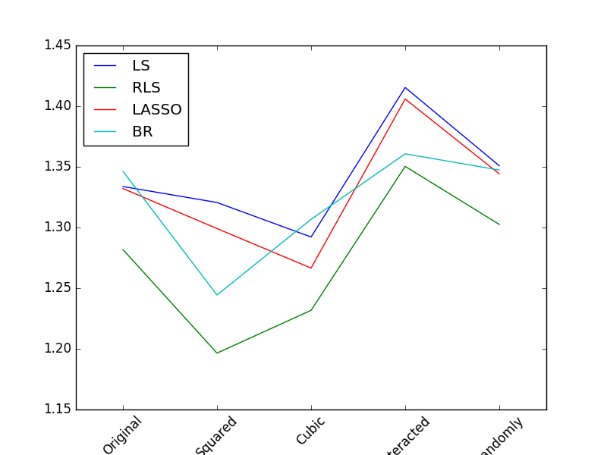


Figure 14. A comparison between different feature vectors.

In Figure 14, we compare different feature vectors in term of MAE and MSE. The less MAE and MSE, the better the predicted counts. For MAE, when we import squared features, all four regression methods obtain more precise results than the original one. When the cubed features are imported, only LS and LASSO methods could better results than squared features. But all methods still obtain better results than the original one. We should note that, it is not always that the more features the better the results. When we import interacted features and random features, all regression methods obtain extremely bad results. Obviously, even the RLS and BR could not prevent the functions to be overfitted. For MSE, the conclusions are similar to MAE. One interesting point is that the RLS always obtain the best results than other methods. The BR is the second and the LS is the worst.

In summary, the conclusions are as follows:

* In this problem, adding some suitable features could increase the performance. However, what features are added should be carefully chosen. In the proposed 5 feature vectors, the squared one seems the best.
* In this problem, both MAE and MSE demonstrate that the RLS is a good method to obtain more precision predictive counts, no matter what features are given.