CS5487 PROGRAMMING ASSIGNMENT 1: REGRESSION	
WANG Yue	56359462

PART ONE: Polynomial function

◆ (a) Implement 5 regression algorithm

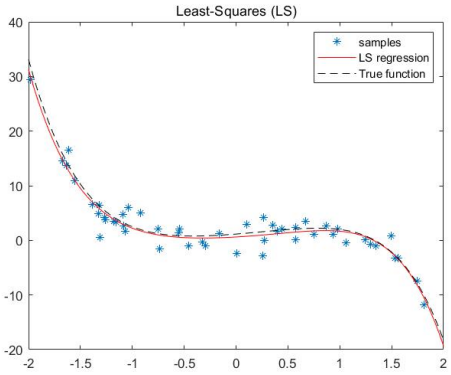
Implementation written in matlab is attached in the source code files.

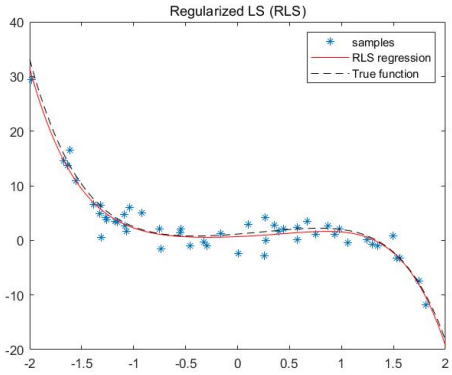
File function:

Part_one (subfile)	This file is the code that implements part one question.
main_1b.m	Main function to call the 5 regression function via predefined parameters for the 1(b) question.
main_1c.m	Main function to call the 5 regression function for the 1(c) question.
main_1c_AveErrFig.m	Draw the figure of MSE versus training size.
main_1d.m	Main function to call the 5 regression function for the 1(d) question.
main_1e.m	Main function to call the 5 regression function for the 1(e) question.
LS.m / RLS.m / LASSO.m / RR.m / BR.m	Linear Square function / Regularized LS function / LASSO function / Robust Regression function / Bayesian Regression function for the 1st question.
plotLS.m / plotRLS.m / plotRR.m / plotBR.m / plotLASSO.m	Draw LS / RLS / RR / BR / LASSO regression function and calculate the MSE.
Part_two (subfile)	This file is the code that implements part two question.
main_2a.m	Main function to call the 5 regression function via predefined parameters for the 2(a) question.
main_2b1.m	Main function to call the 5 regression function for the 2(b) question. (using $\Phi(x)=[x_1,x_2,x_3,...,x_1^2,x_2^2,x_3^2,...]^T$)
main_2b2.m	Main function to call the 5 regression function for the 2(b) question. (using $\Phi(x)=[x_1,x_2,x_3,...,x_1x_2,x_1x_3,x_2x_3,...,x_1^2,x_2^2,x_3^2,...]^T$)
main_2b3.m	Main function to call the 5 regression function for the 2(b) question. (using $\Phi(x)=\tanh(x)=\frac{e^x+e^{-x}}{e^x-e^{-x}}$)

◆ (b) Regression plots and hyper-parameters tunning

<1> use sample data to estimate 5-th order poly function





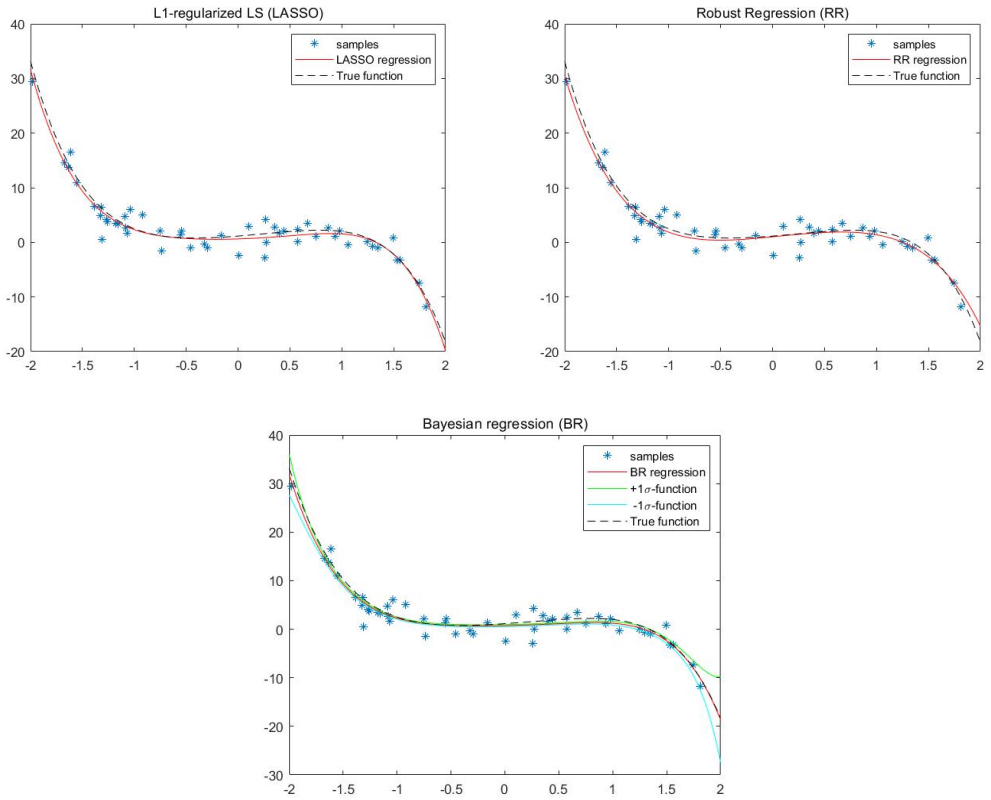


Figure set 1b.1 – Predict function of 5 different regression methods

Table 1b.2 – MES of 5 different regression methods		
Regression function		MES
Least-Squares(LS)		0.408644
Regularized LS (RLS)	(Lambda=1)	0.408633
L1-regularized LS (LASSO)	(Lambda=1)	0.475354
Robust regression (RR)		0.768046
Bayesian regression (BR)	(alpha-1, sigma^2=5)	0.459158

Figure set 1b.1 are plots of predictive functions of 5 different regression methods. Table 1b.2 shows the MSE of these 5 different regression methods. We can see that LS, RLS, LASSO and BR have similar MSE, while RR has a relatively larger number around 0.7680. The hyper-parameters in methods RLS, LASSO and BR have not been adjusted at this time, so next, let's adjust the hyper-parameters to make the result better.

<2> hyper-parameters tunning

Table 1b.3 – MES of different hyper–parameter lambda in RLS								
lambda	0.1	0.25	0.5	0.75	1	2	5	10
MSE of RLS	0.408237	0.407827	0.407600	0.407882	0.408633	0.415603	0.459158	0.557904

Table 1b.4 – MES of different hyper–parameter lambda in LASSO								
lambda	0.1	0.25	0.5	0.75	1	2	5	10
MSE.LASSO	0.413901	0.421799	0.436859	0.454710	0.475354	0.519129	0.569842	0.729741

Table 1b.5 – MES of different hyper–parameter alpha and sigmaˆ2 in BR								
alpha	0.1	0.1	0.1	0.1	0.5	0.5	0.5	0.5
sigma^2	0.1	0.5	1	10	0.1	0.5	1	10
MSE of BR	0.408633	0.459158	0.557904	1.754277	0.407939	0.408633	0.415603	0.762126
alpha	1	1	1	1	5	5	5	5
sigma^2	0.1	0.5	1	10	0.1	0.5	1	10
MSE of BR	0.408237	0.407600	0.408633	0.557904	0.408553	0.408237	0.407939	0.415603

Tables 1b.3-5 record the MSE results for setting different hyper-parameters in RLS, LASSO and BR. In RLS, the prediction is better when lambda is 0.5. In LASSO, the prediction is better when lambda is 0.1. In BR, the prediction is better when alpha is 1 and sigma^2 is 0.25. Values in mauve shade corresponds to 'optimal choice' in these experiments. (note: LS and RR required no hyper-parameters)

◆ (c) Regression using subset of training data and learning curves

In this experiment, subset sizes of samples are 10%, 25%, 50%, 75%. For each size of subset, this experiment run 5 trials of different random subsets and take the average error.

Hyper-parameters setting: lambda(RLS)=10, lambda(LASSO)=2, alpha(BR)=1, sigma_square(BR)=1.

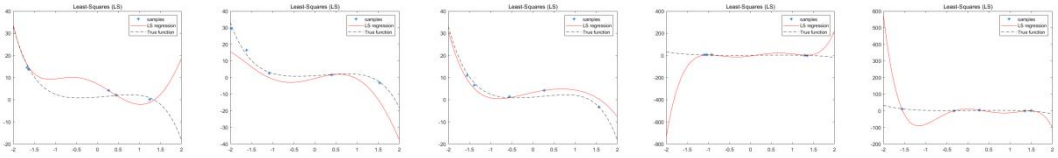


Figure 1c.1 – Predict function of LS when subset size of samples is 10%

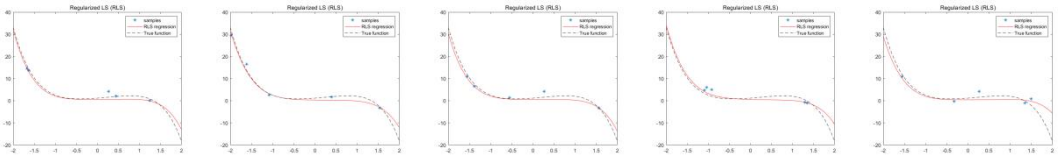


Figure 1c.2 – Predict function of RLS when subset size of samples is 10%

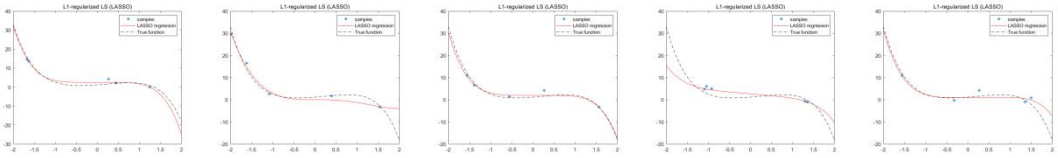


Figure 1c.3 – Predict function of LASSO when subset size of samples is 10%

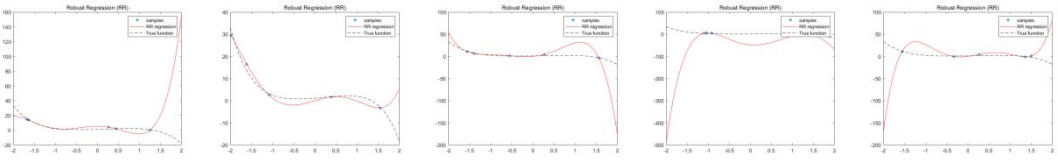


Figure 1c.4 – Predict function of RR when subset size of samples is 10%

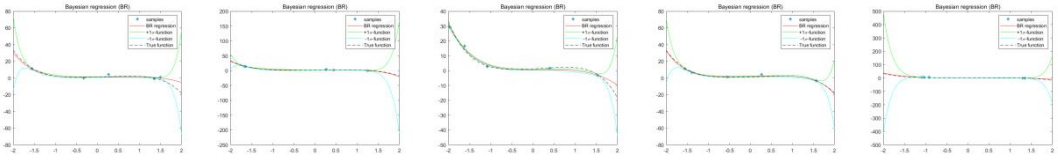


Figure 1c.5 – Predict function of BR when subset size of samples is 10%

Table 1c.6 – MES of different regression methods when subset sizes of samples are 10%						
	MSE(1)	MSE(2)	MSE(3)	MSE(4)	MSE(5)	Average MSE(about)
LS	10859.299	82.745	57.205	9.726	26538.018	7509.398
RLS	8.847	1.851	2.446	2.132	3.368	3.728
LASSO	6.029	3.344	9.233	0.912	18.539	7.611
RR	1743.823	1267.166	20.341	957.261	12612.150	3320.148
BR	8.057	0.951405	3.435	0.350	6.243	3.807

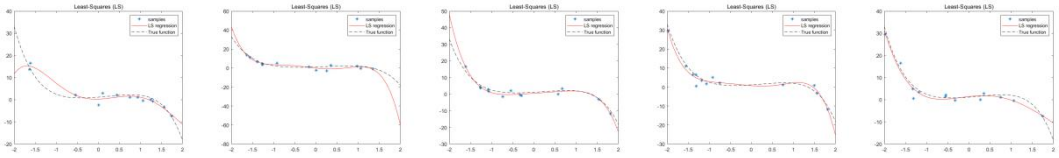


Figure 1c.7 – Predict function of LS when subset size of samples is 25%

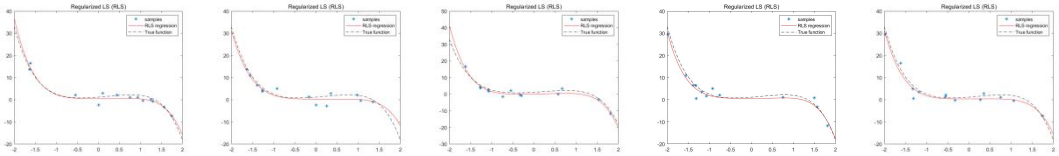


Figure 1c.8 – Predict function of RLS when subset size of samples is 25%

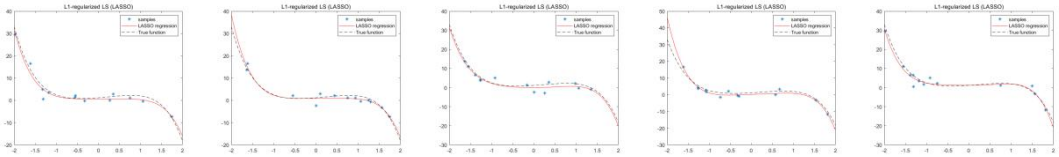


Figure 1c.9 – Predict function of LASSO when subset size of samples is 25%

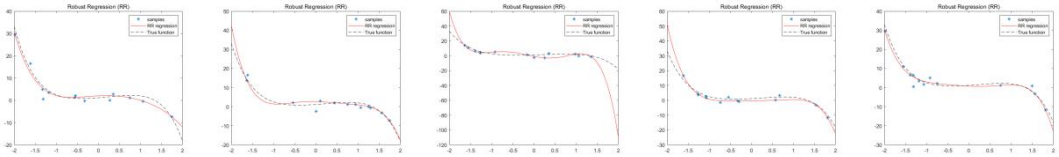


Figure 1c.10 – Predict function of RR when subset size of samples is 25%

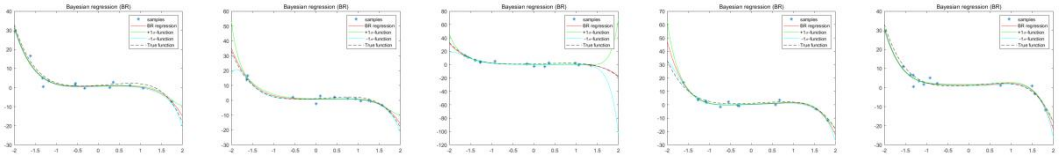


Figure 1c.11 – Predict function of BR when subset size of samples is 25%

Table 1c.12 – MES of different regression methods when subset sizes of samples are 25%						
	MSE(1)	MSE(2)	MSE(3)	MSE(4)	MSE(5)	Average MSE(about)
LS	19.444	70.531	9.527	3.198	3.169	21.173
RLS	1.383	3.013	3.916	1.410	1.719	2.288
LASSO	1.607	1.320	7.798	1.250	1.391	2.673
RR	3.618	330.33	13.676	1.800	2.536	70.392
BR	0.911	1.062	8.723	1.502	1.355	2.710

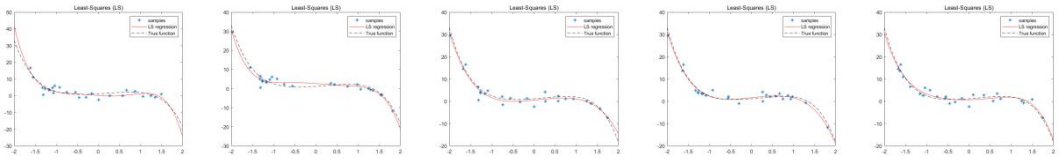


Figure 1c.13 – Predict function of LS when subset size of samples is 50%

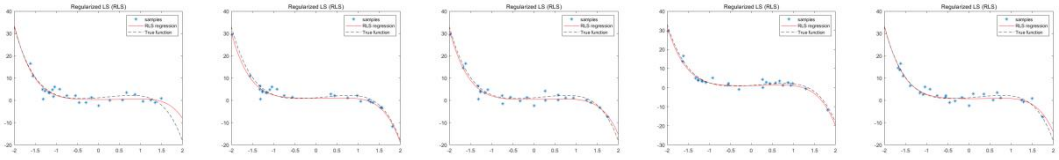


Figure 1c.14 – Predict function of RLS when subset size of samples is 50%

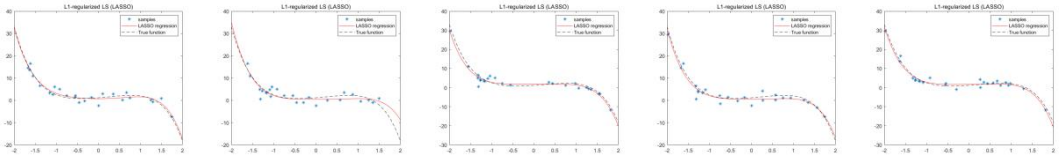


Figure 1c.15 – Predict function of LASSO when subset size of samples is 50%

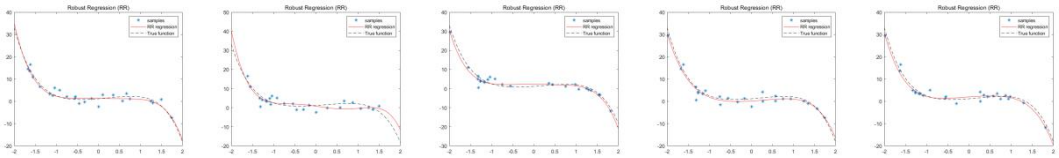


Figure 1c.16 – Predict function of RR when subset size of samples is 50%

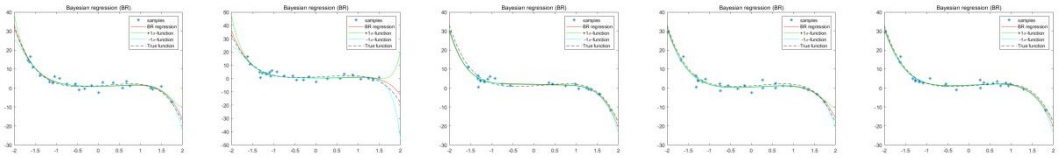


Figure 1c.17 – Predict function of BR when subset size of samples is 50%

Table 1c.18 – MES of different regression methods when subset sizes of samples are 50%

	MSE(1)	MSE(2)	MSE(3)	MSE(4)	MSE(5)	Average MSE(about)
LS	3.822	2.746	0.983	0.680	0.533	1.752
RLS	5.423	1.280	1.351	0.721	1.047	1.964
LASSO	5.002	1.397	0.973	0.824	0.376	1.714
RR	5.944	1.890	1.035	1.243	0.491	2.120
BR	2.854	1.653	0.861	0.633	0.432	1.286

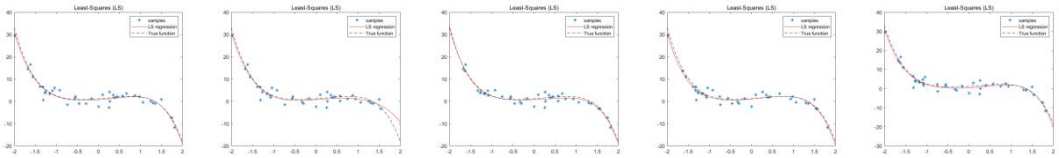


Figure 1c.19 – Predict function of LS when subset size of samples is 75%

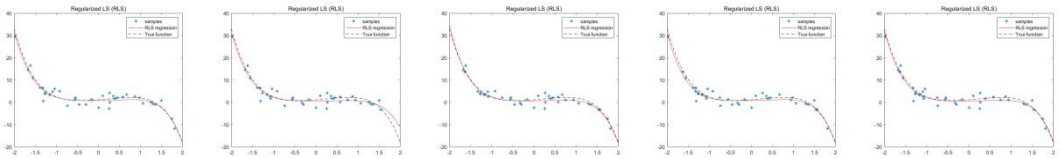


Figure 1c.20 – Predict function of RLS when subset size of samples is 75%

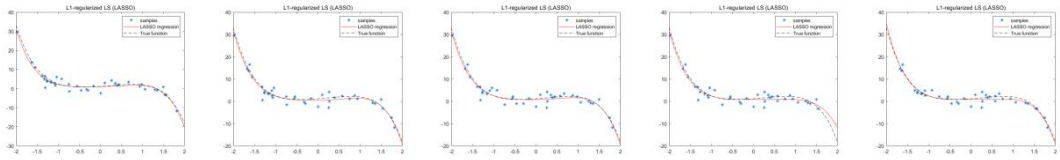


Figure 1c.21 – Predict function of LASSO when subset size of samples is 75%

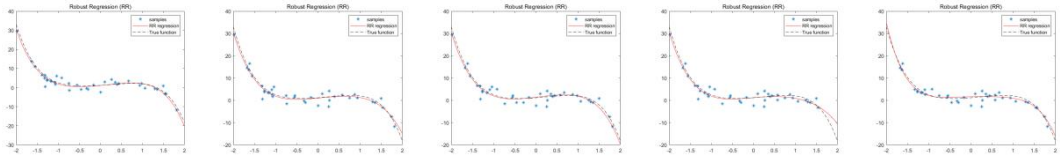


Figure 1c.22 – Predict function of RR when subset size of samples is 75%

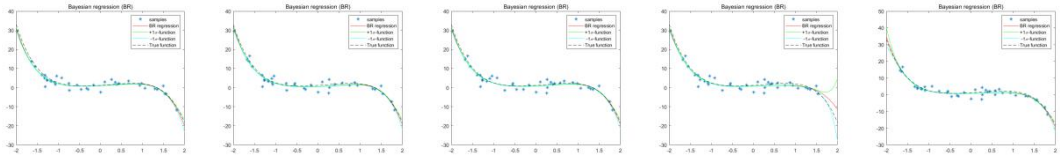


Figure 1c.23 – Predict function of BR when subset size of samples is 75%

Table 1c.24 – MES of different regression methods when subset sizes of samples are 75%						
	MSE(1)	MSE(2)	MSE(3)	MSE(4)	MSE(5)	Average MSE(about)
LS	0.346	3.487	0.253	0.703	0.490	1.055
RLS	0.522	2.706	0.565	0.922	0.687	1.080
LASSO	0.444	2.313	0.476	0.886	0.557	0.935
RR	0.830	2.586	0.550	0.862	0.854	1.136
BR	0.348	2.328	0.350	0.763	0.456	1.136

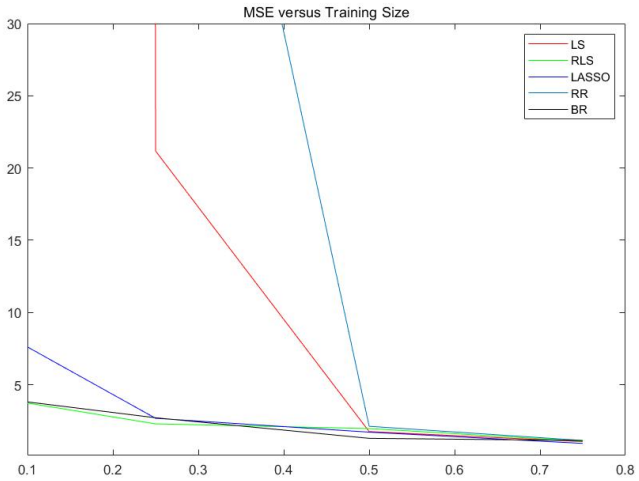


Figure 1c.25 – MSE of 5 different regression methods with 10%, 25%, 50% and 75% subset of samples

Conclusion:

(i) From Table 1c.6, Table 1c.12, figure1c.2, figure1c.8, figure1c.5 and figure1c.11, we can infer that RLS and BR have better regression performances when the size of dataset is small, for they have intuitively 'closer' line in the prediction function plots and much smaller MSE indicated by the learning

curves. The regression performance of LASSO is also good when use 25% subset of samples. These may indicate that RLS, LASSO and BR have better resistances against overfitting.

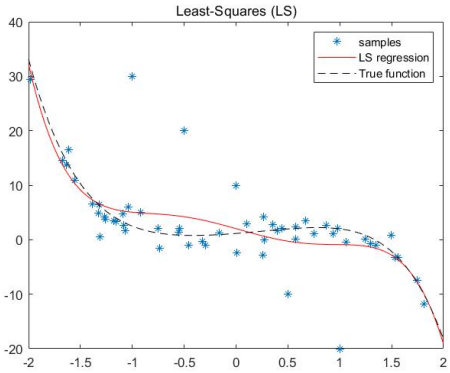
(ii) On the other hand, we can see that when the percentage using the sample is greater than 50%, all of these 5 methods worked well.

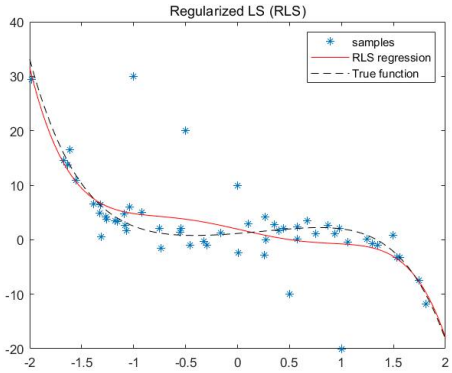
(iii) Bayesian Regression tend to be robust even when dataset is small due to the existence of prior knowledge. However, LS, RLS and LASSO are data-driven training methods.

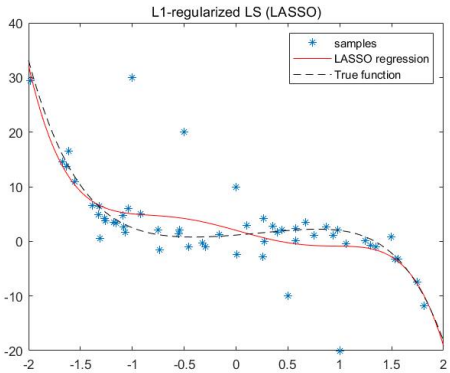
◆ (d) Add some outliers output values

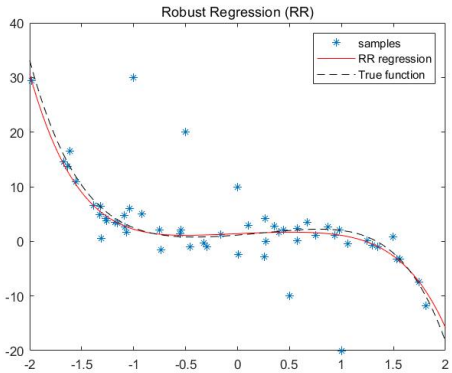
In this section, 5 outliers are added to the training set is: outliers_x = [-1, -0.5, 0, 0.5, 1]; outliers_y = [30; 20; 10; -10; -20].

Hyper-parameters setting: lambda(RLS)=0.5, lambda(LASSO)=0.1, alpha(BR)=1, sigma_square(BR)=10.









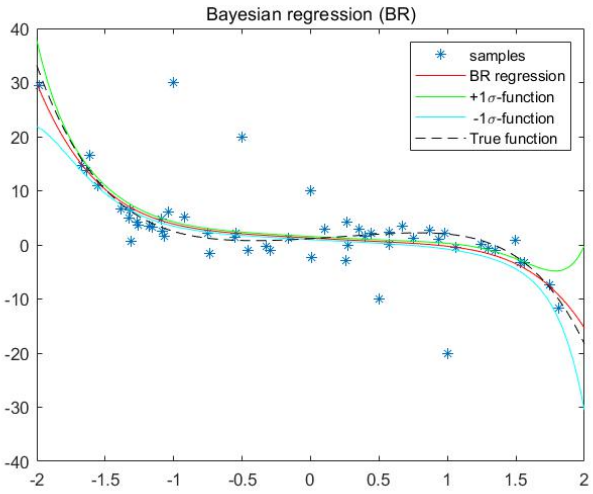


Figure 1d.1 – Predict function of 5 different regression methods when adds some outliers output values

Table 1d.2 – MES of different regression methods when adds some outliers output values					
	LS	RLS	LASSO	RR	BR
MSE	4.509528	3.764370	4.455100	0.932676	2.166514

Conclusion:

(i) From figure 1d.1 we can see that RR have the best performance to resist outliers for it has a similar curve with normal experiment. What's more, in table 1d.2, MSE of RR stays at around 0.9 which is much smaller than other four methods, this is another evidence that RR is robust when some outliers are added to the dataset.

(ii) Besides RR, BR also has a good performance to resist outliers.

(iii) LS, RLS and LASSO are more sensitive.

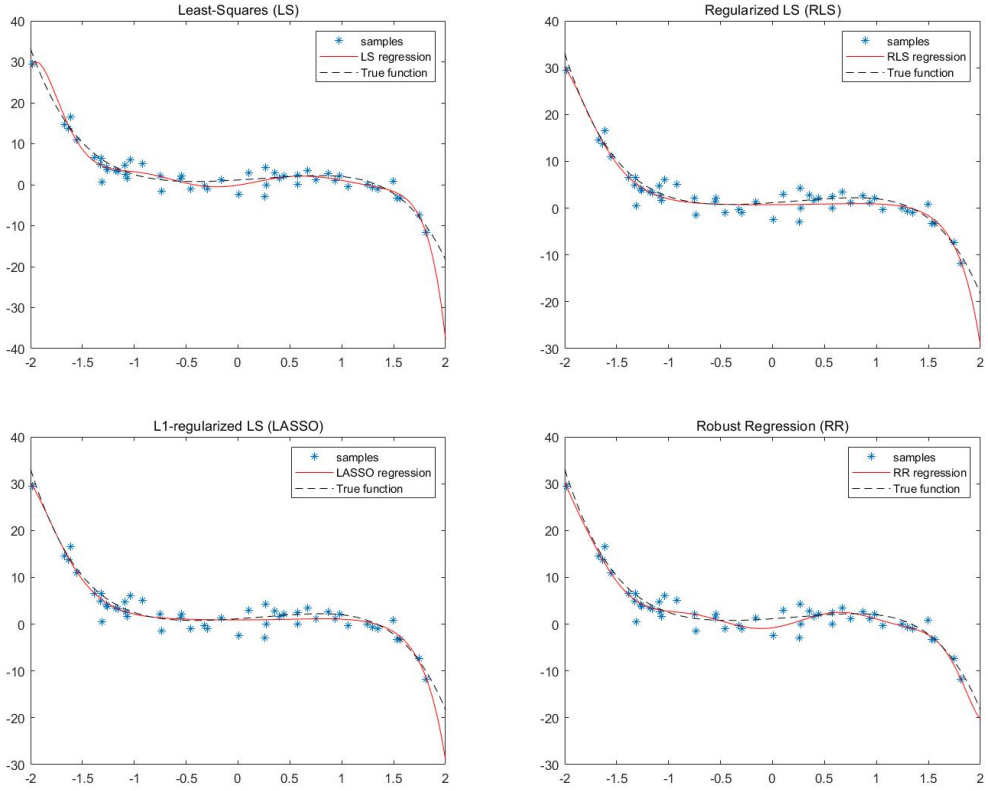
(iv) Reason for the robust of RR: Robust Regression is designed to limit the effects of outliers. I think may the Robust Regression didn't use the squared difference between predicted value and true value, while others are all based on the Least square method.

◆ (e) Higher-order polynomial (10th order)

In this section, repeat (b) but estimate a higher-order polynomial(10th order).

So there are 11 estimated parameters which are the coefficients of x^n, n is integers from 0 to 10.

Hyper-parameters setting: lambda(RLS)=10, lambda(LASSO)=1.8, alpha(BR)=0.1, sigma_square(BR)=1.



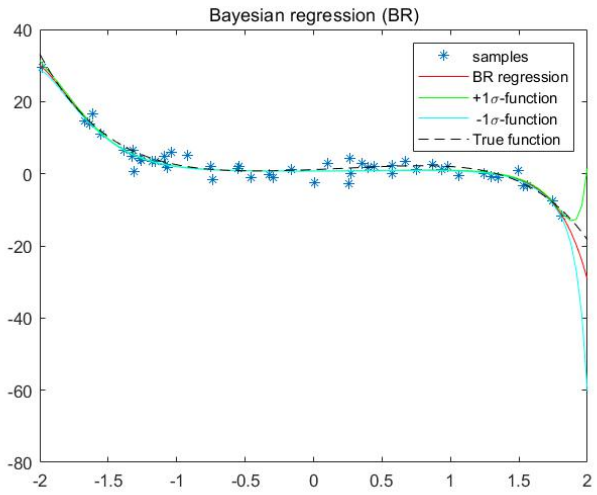


Figure 1e.1 – Predict function of 5 different regression methods (10th order)

Table 1e.2 – MSE of different regression method when adds some outliers output values					
	LS	RLS	LASSO	RR	BR
MSE	7.983107	2.876626	2.637049	1.289857	2.876626

Table 1e.3 – Estimated parameter values of 5 different regression methods (10th order)						
	LS	RLS	LASSO	RR	BR	**TRUE**
parameter of X^0	-0.099375	0.734810	0.912690	-0.772141	0.734810	1.152434
parameter of X^1	3.793512	0.115234	5.809251e-09	3.025202	0.115234	1.486292
parameter of X^2	6.491220	0.448380	0.559474	13.148852	0.448380	0.929505
parameter of X^3	-10.744773	-0.201685	-1.97390e-12	-7.575929	-0.201685	-1.113441
parameter of X^4	-5.521677	0.136530	-3.64918e-13	-19.780027	0.136530	0.159804
parameter of X^5	8.632263	-0.237231	-0.354926	5.642388	-0.237231	-0.617880
parameter of X^6	0.653713	0.004109	-8.18093e-13	12.194465	0.004109	0
parameter of X^7	-3.045798	-0.227347	-0.201194	-2.164199	-0.227347	0
parameter of X^8	0.763549	0.154080	0.169457	-3.157239	0.154080	0
parameter of X^9	0.310281	0.016373	0.014773	0.245386	0.016373	0
parameter of X^{10}	-0.175137	-0.043011	-0.044899	0.290522	-0.043011	0

Conclusion:

(i) From Table 1e.2 and Table 1e.3 we can see that LS has the largest MSE, combing with figure 1e.1 we can infer that LS tends to overfit when learning a more complex model.

(ii) RLS, LASSO and BR have similar MSE around 2.7, combing with figure 1e.1 we can infer that RLS, LASSO and BR can avoid get overfitting.

(iii) For Bayesian Regression,due to the existence of prior knowledge,it has the better performance.

(iv) The MSE of RR in table 1e.2 is small, but when observing the figure 1e.1, we can find the RR curve is twisted and different to the true function. Therefore, RR tends to overfit when the order of function is high-order. The reason why it has a low MSE is that the curve is close to the true function on its two terminals when we scope the extreme data points which contributes a lot to the MSE.

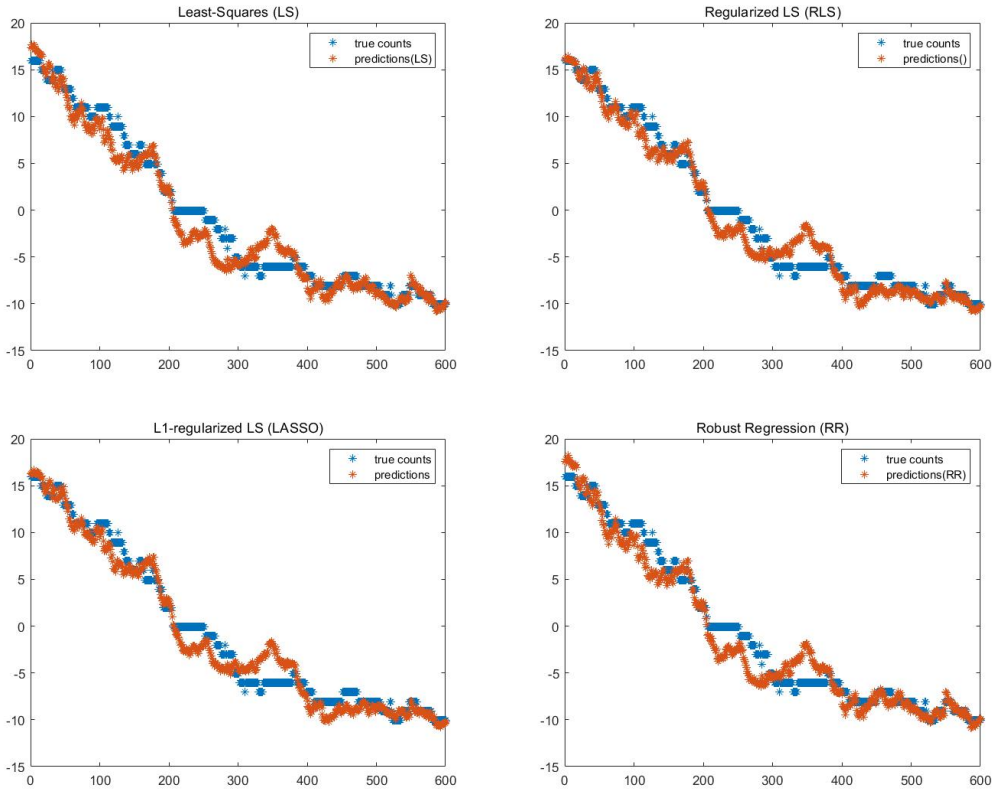
(v) From table 1e.3, we can see that RR an LS tend to have large values in higher order scalars. Because we have knowledge that true function is a 5-order function, a well fitted set of parameter values should small when the order is higher than 5. Phenomena on LS and RR infer that these two estimators tend to overfit the data. On the contrary, RLS, LASSO and BR can avoid get overfitting to some extent.

PART TWO: real world regression problem – counting people

◆ (a) using the features directly

In this section, the original feature is used to predict the number of people.

Hyper-parameters setting: lambda(RLS)=0.9, lambda(LASSO)=4, alpha(BR)=5, sigma_square(BR)=5.



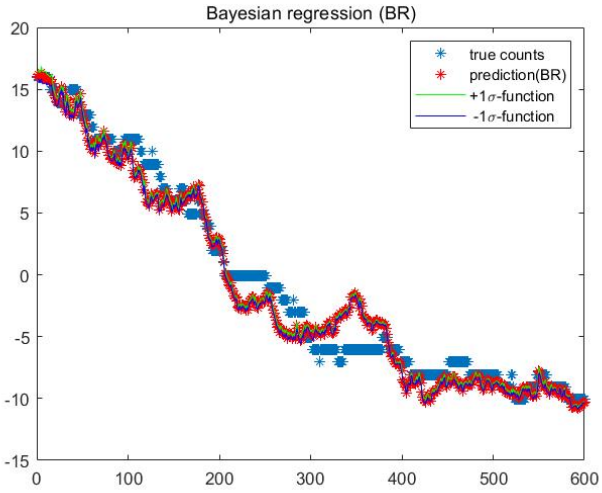


Figure 2a.1 – Predict function of 5 different regression methods (using the features directly)

Table 2a.2 – MSE and MAE of 5 different regression methods (using the features directly)					
	LS	RLS	LASSO	RR	BR
MSE	3.102838	2.615528	2.449133	3.118998	2.618734
MAE	1.358444	1.279765	1.252033	1.364567	1.282433

Conclusion:

(i) According to the table 2a.2, we can see that LASSO works best compared to other methods.

(ii) From figure 2a.1 and table 2a.2, we can see that shape of prediction plots is similar across all methods, and the difference of MSE and MAE between different methods is not significant.

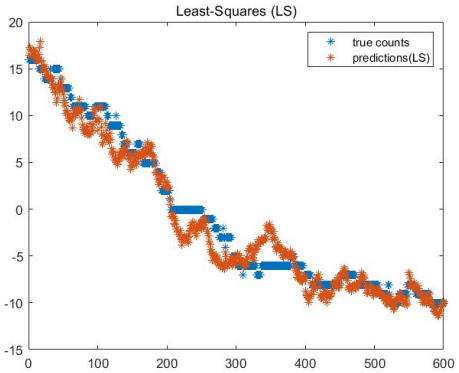
(iii) From figure 2a.1, most inaccurate predictions are located in the region between -7 to 0 (sample 200 - 400). If we can improve the performance in this region, the global performance may increase.

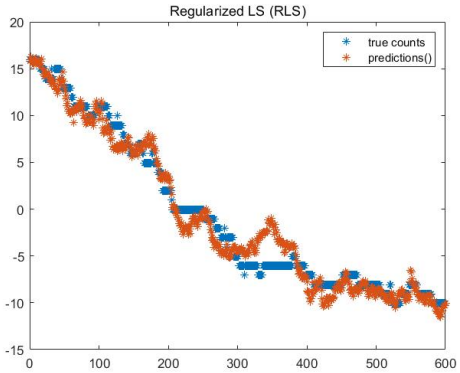
◆ (b) using some other feature transformations (3 kinds of)

<1> using $\Phi(x) = [x_1, x_2, x_3, ..., x_1^2, x_2^2, x_3^2, ...]^T$ as feature transformation 1

This section predicts the number of people by using $\Phi(x) = [x_1, x_2, x_3, ..., x_1^2, x_2^2, x_3^2, ...]^T$ feature transformation. And this feature transformation has 18 parameters.

Hyper-parameters setting: lambda(RLS)=0.9, lambda(LASSO)=4, alpha(BR)=5, sigma_square(BR)=5.





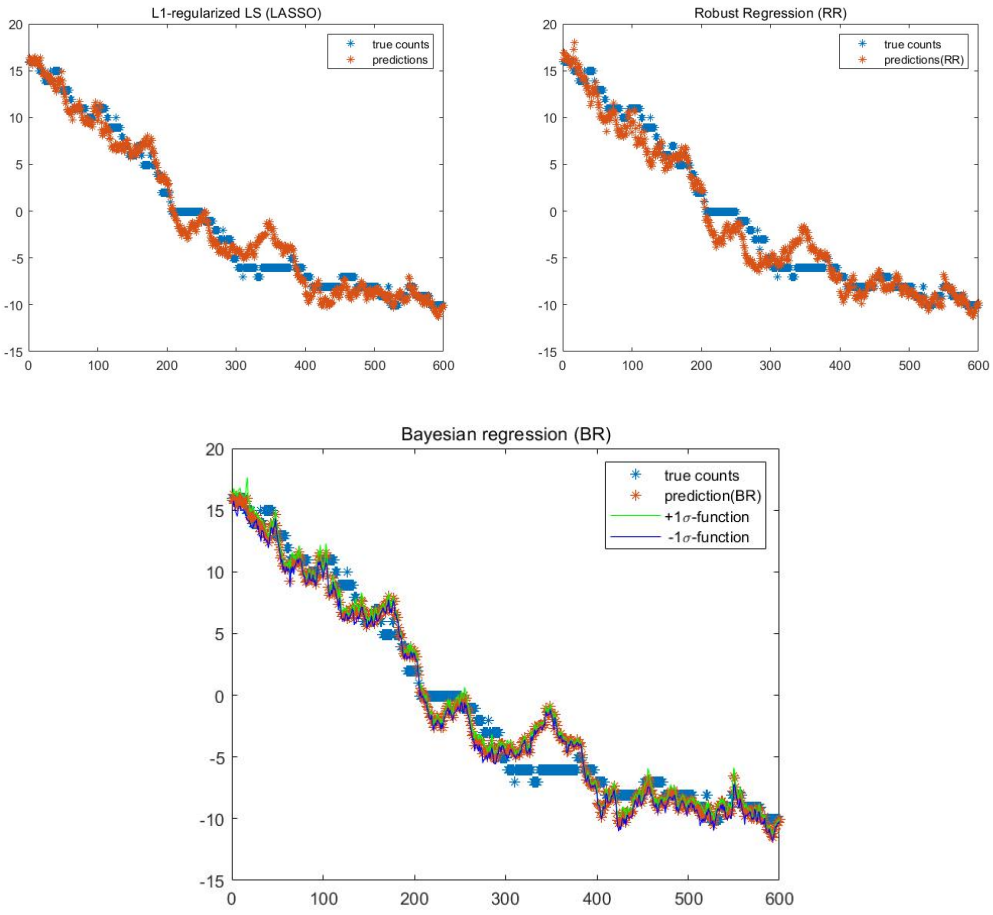


Figure 2b.1 – Predict function of 5 different regression methods (using the features transformation 1)

Table 2b. 2– MSE and MAE of 5 different regression methods (using the features transformation 1)					
	LS	RLS	LASSO	RR	BR
MSE	2.923594	2.453200	2.282151	2.898128	2.467774
MAE	1.326746	1.211205	1.177985	1.307929	1.213699

Comparing with the question 2(a), we can see the performance of 5 different regression methods have improved a little bit when using the features transformation 1.

<2> using $\Phi(x) = [x_1, x_2, x_3, ..., x_1x_2, x_1x_3, x_2x_3, ..., x_1^2, x_2^2, x_3^2, ...]^T$ as feature transformation 2

This section predicts the number of people by using $\Phi(x) = [x_1, x_2, x_3, ..., x_1x_2, x_1x_3, x_2x_3, ..., x_1^2, x_2^2, x_3^2, ...]^T$ feature transformation. This feature transformation has 54 parameters. (9+45)

Hyper-parameters setting: lambda(RLS)=0.9, lambda(LASSO)=4, alpha(BR)=5, sigma_square(BR)=5.

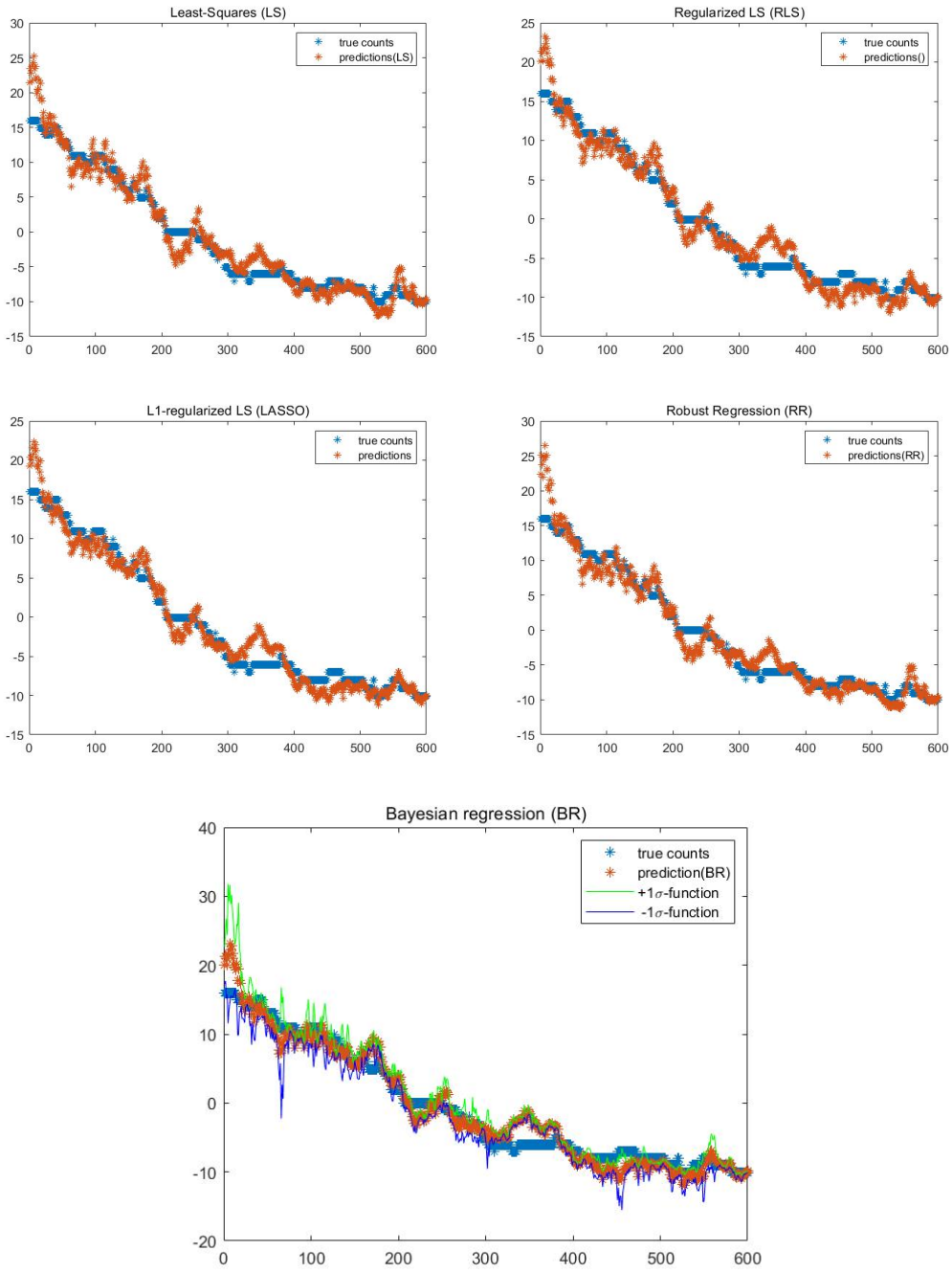


Figure 2b.3 – Predict function of 5 different regression methods (using the features transformation 2)

Table 2b. 4– MSE and MAE of 5 different regression methods (using the features transformation 2)					
	LS	RLS	LASSO	RR	BR
MSE	4.059715	3.687656	3.162133	4.199906	3.671580
MAE	1.488691	1.479903	1.391435	1.460152	1.482448

Comparing with the question 2(a), we can see that, although the parameters have been multiplied several times, performance of 5 different regression methods is a little worse when using the features transformation 2.

<3> using $\Phi(x) = \tanh(x) = \frac{e^{x} + e^{-x}}{e^{x} - e^{-x}}$ **as feature transformation 3**

This section predicts the number of people by using above feature transformation.

This feature transformation has 9 parameters.

Hyper-parameters setting: lambda(RLS)=0.6, lambda(LASSO)=4, alpha(BR)=10, sigma_square(BR)=1.

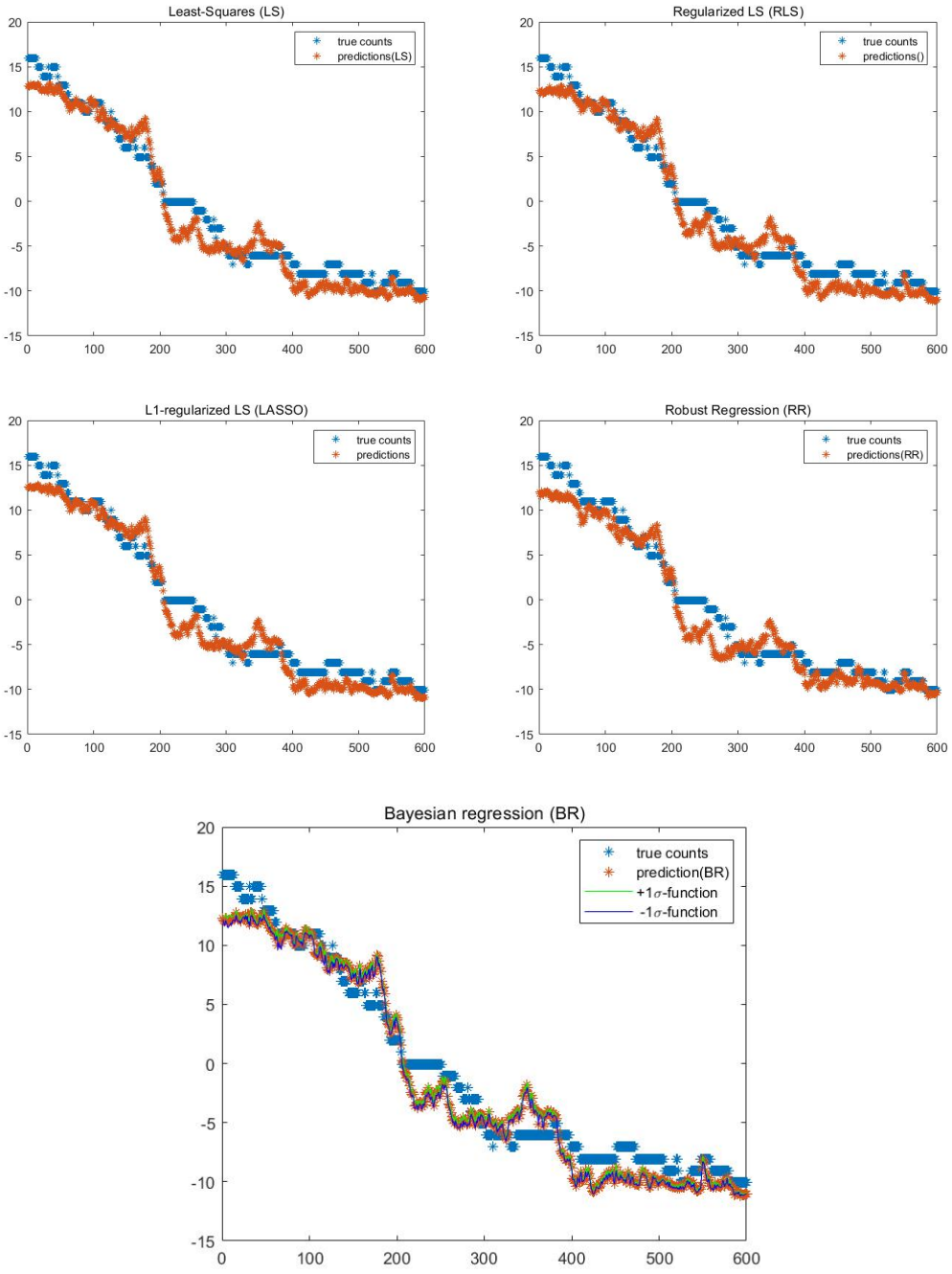


Figure 2b.5 – Predict function of 5 different regression methods (using the features transformation 3)

Table 2b. 6– MSE and MAE of 5 different regression methods (using the features transformation 3)

	LS	RLS	LASSO	RR	BR
MSE	3.588952	3.651371	3.531240	4.164170	3.588815
MAE	1.574367	1.623676	1.578469	1.628856	1.575346

Comparing with the question 2(a), we can see that the performance of 5 different regression methods is a little worse when using the features transformation 3.