CS5487 Programming Assignment 1

NIE Wan 57120844

October 2022

1 Part 1 Polynomial function

(a)

Implementation of the 5 regression algorithms can be found at the *utils.py* file in the source code, which was also uploaded on Canvas.

(b)

The mean-squared errors (MSEs) of different regression methods are given in Figure 1.

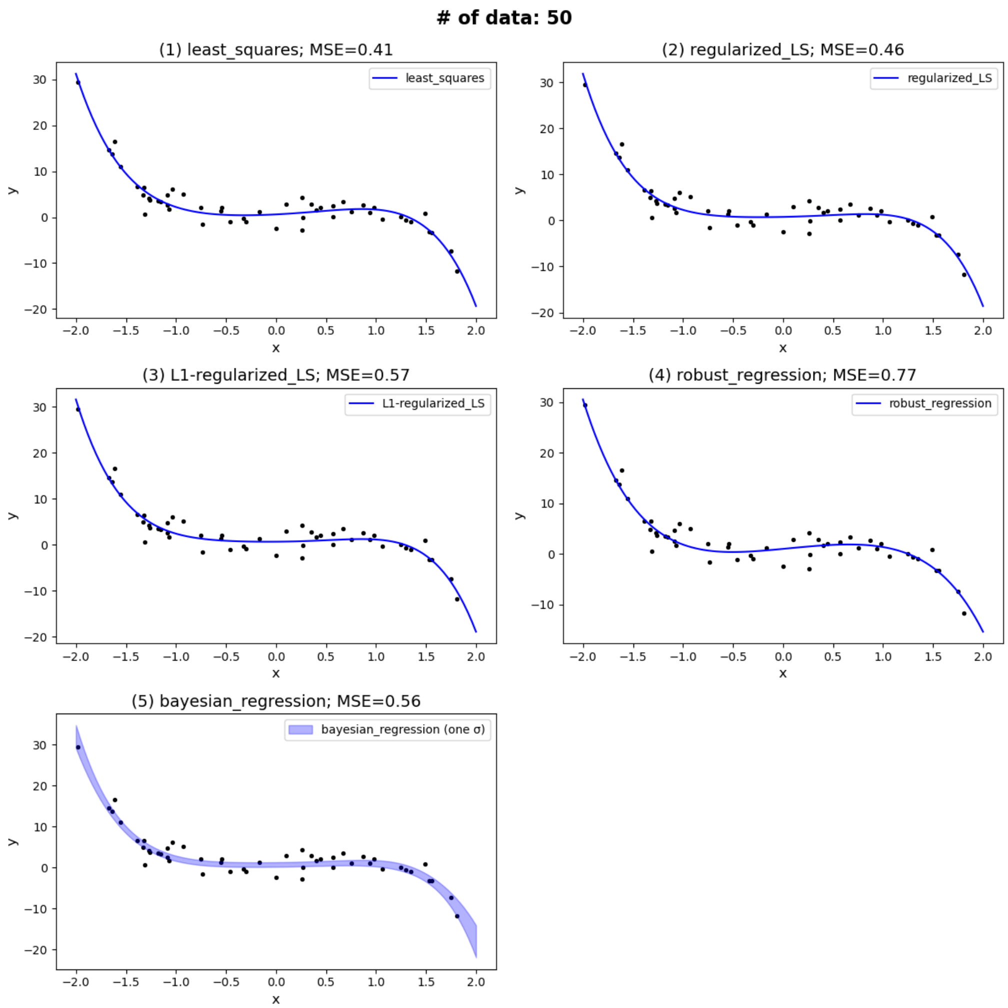

Figure 1: Regression results of different methods ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

(c)

According to the experiment results in Figures 2 to 6, we can find that:

(i) *regularized LS*, *bayesian regression* and *L1-regularized LS* have better performance when the dataset is small.

(ii) *least squares* and *robust regression* are more likely to overfit when the dataset is small, illustrated by the larger MSEs and twisted function curves.

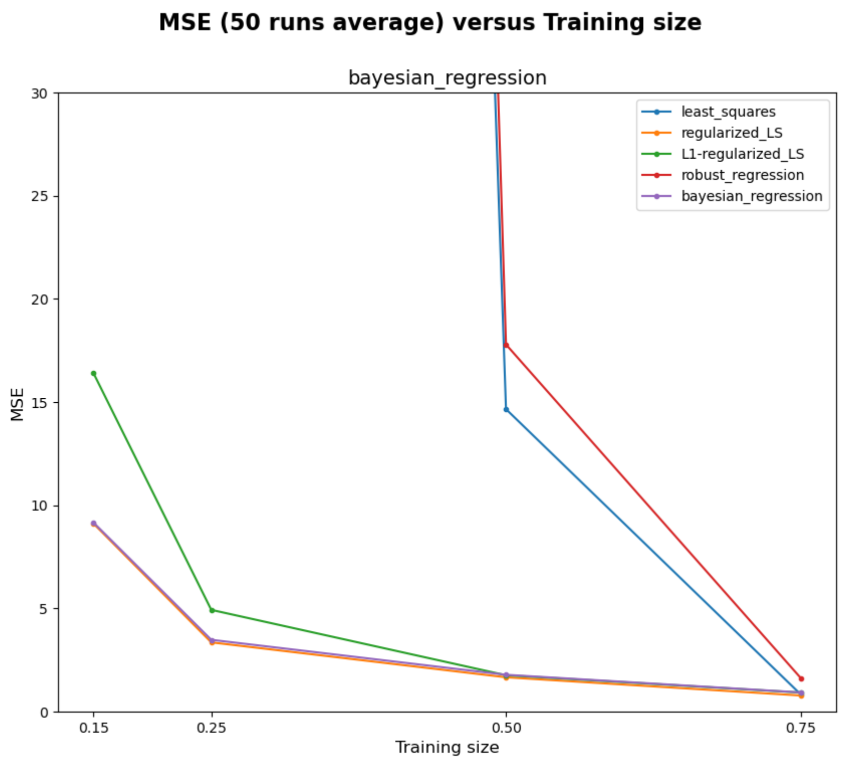

Figure 2: MSE versus training size ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

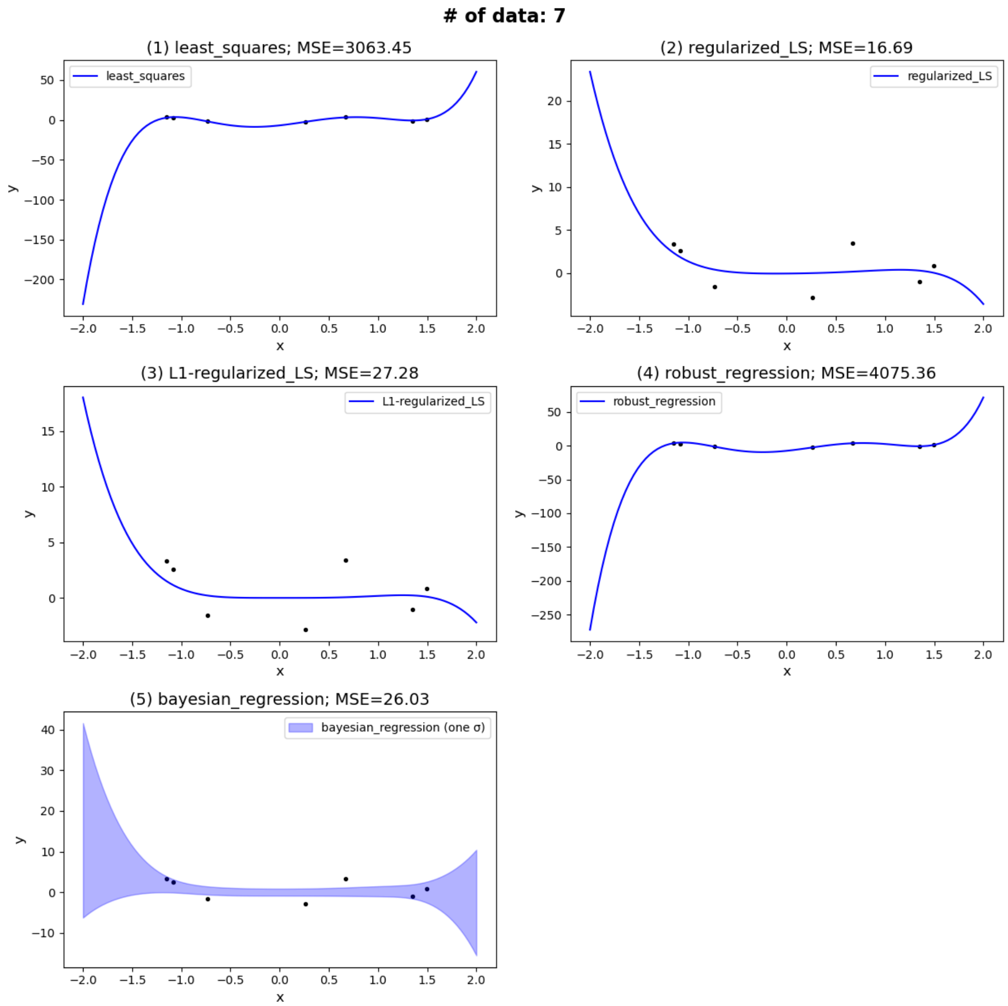

Figure 3: %15 random samples ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

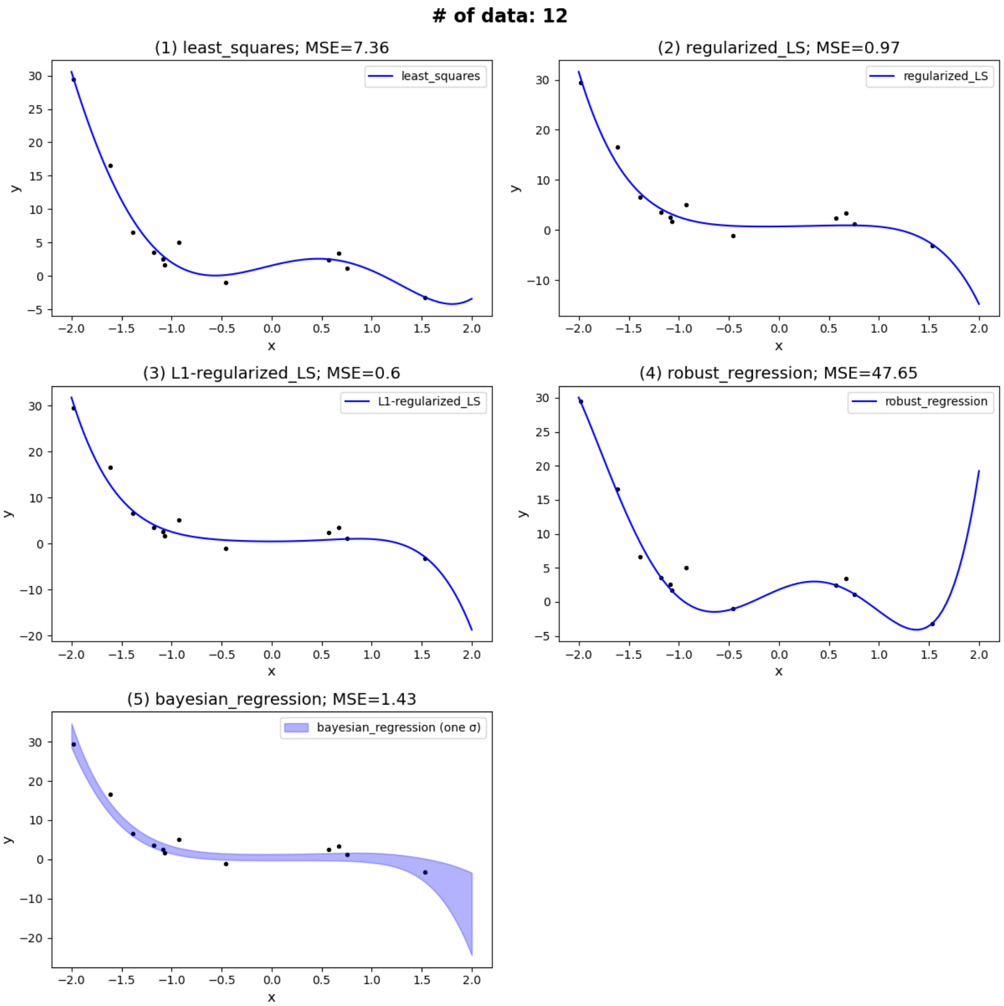

Figure 4: %25 random samples ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

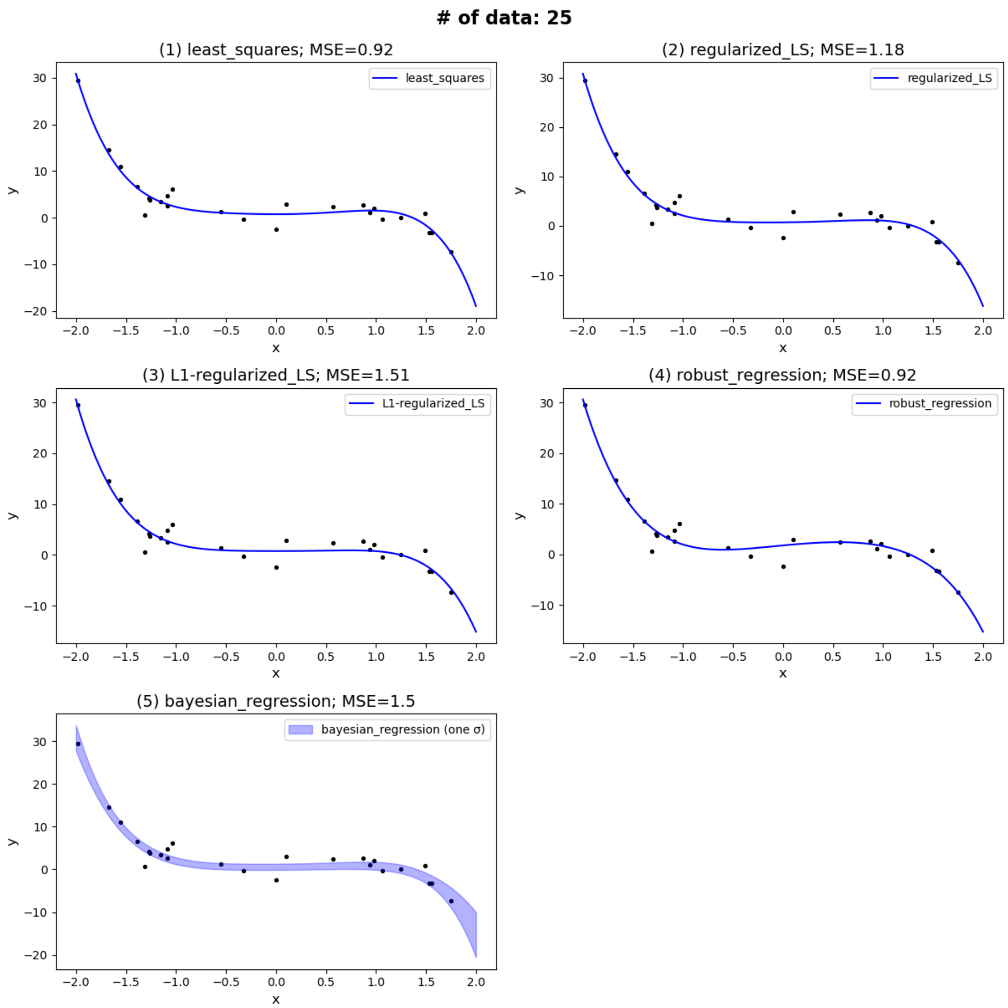

Figure 5: %50 random samples ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

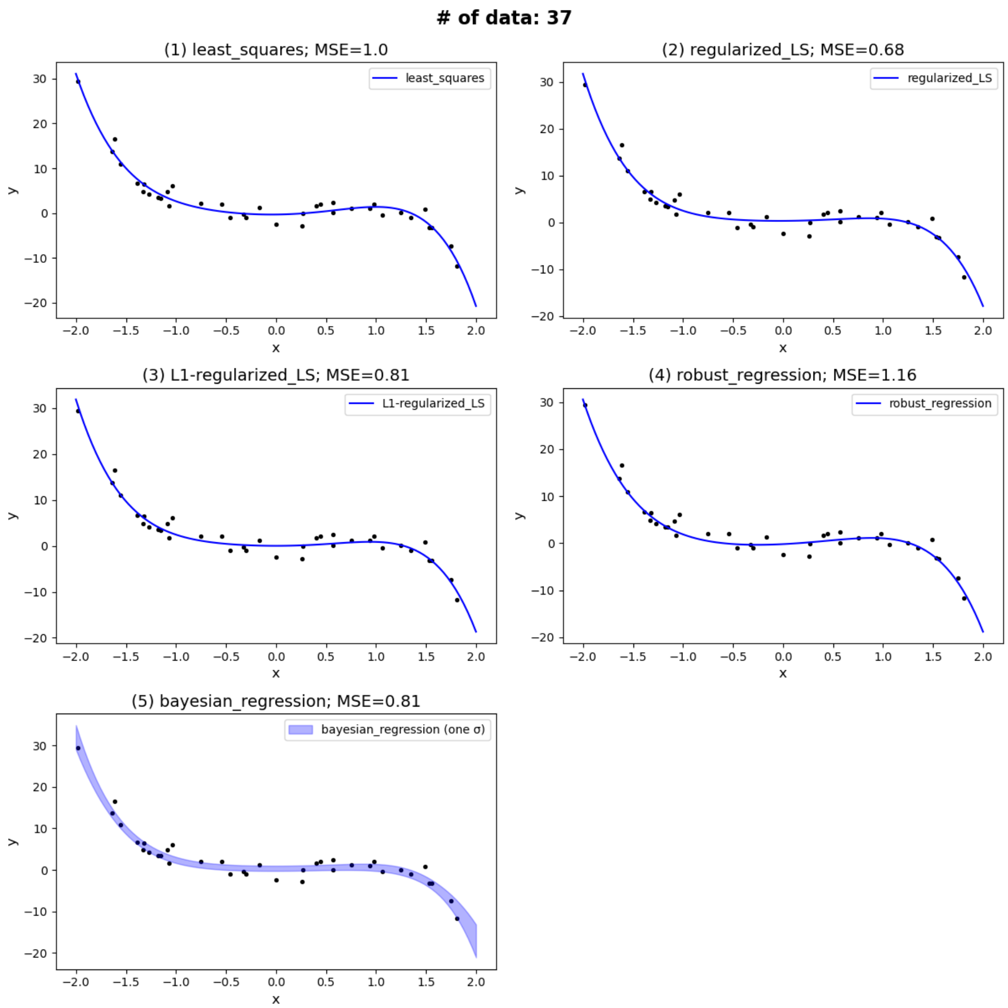

Figure 6: %75 random samples ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

(d)

We added 4 outliers in the sample data. According to the experiment results in Figure 7, we can find that:

(i) *robust regression* and *bayesian regression* are more robust to the presence of outliers compared with the other three methods.

(ii) *robust regression* has a L_1 norm of estimation errors, i.e. $||y - \boldsymbol{\phi}^T\theta||_1$, which reduces the impact of outliers. The prior knowledge of θ in *bayesian regression* also limits the data-driven effects.

(iii) *least squares* is the most sensitive method because its L_2 norm is prone to large estimation errors if there are outliers in the training sample. Although *regularized LS* and *L1-regularized LS* are also using the L_2 norm, the regularization term of θ can reduce the impact of the outliers.

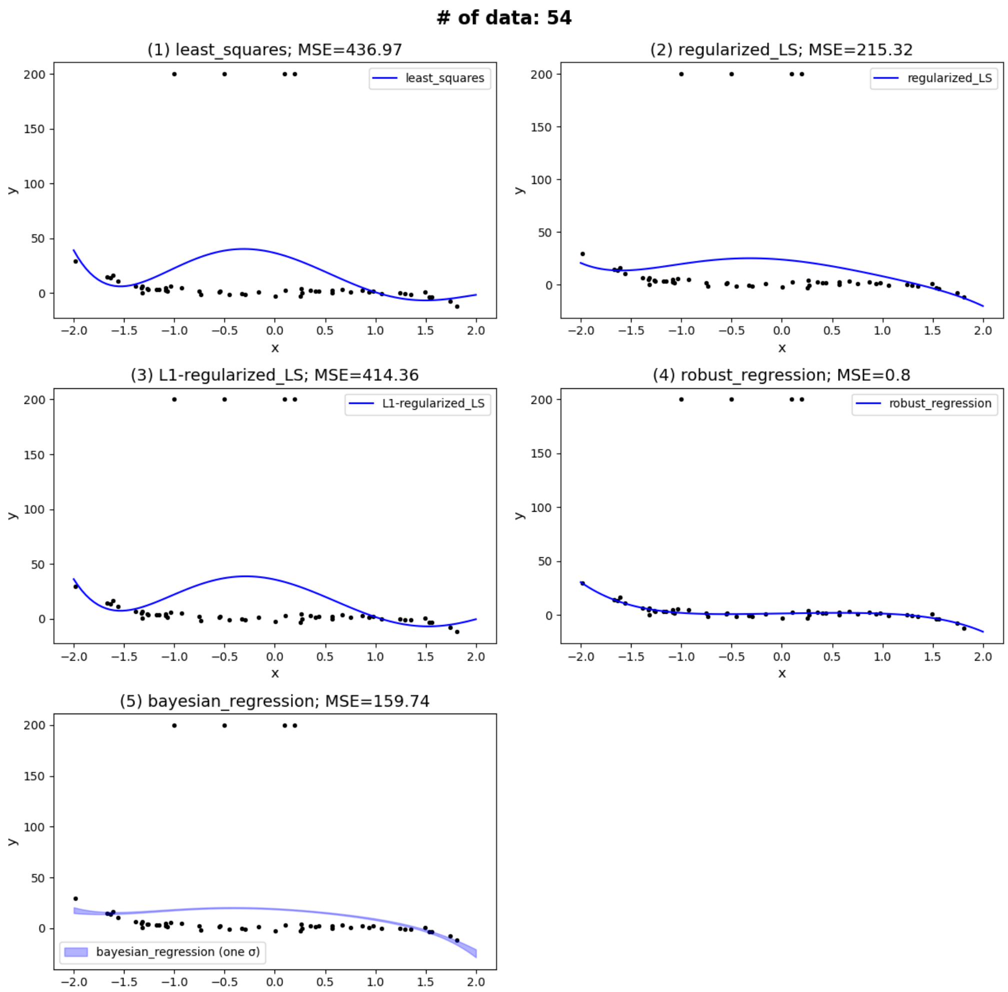

Figure 7: Regression results with outliers ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

(e)

According to the experiment results in Figures 8 and 9, we can find that:

(i) *least squares* tends to overfit the data when learning a more complex model. The function curve of *robust regression* twists more times than the true function curve. Therefore, *robust regression* also tends to overfit the data.

(ii) *regularized LS*, *L1-regularized LS* and *bayesian regression* do not overfit the data.

(iii) The above two observations can be verified by Figure 9 that both *least squares* and *robust regression* have large parameter components, i.e., $|\hat{\theta}_i| > 10$, while other methods' parameters are small.

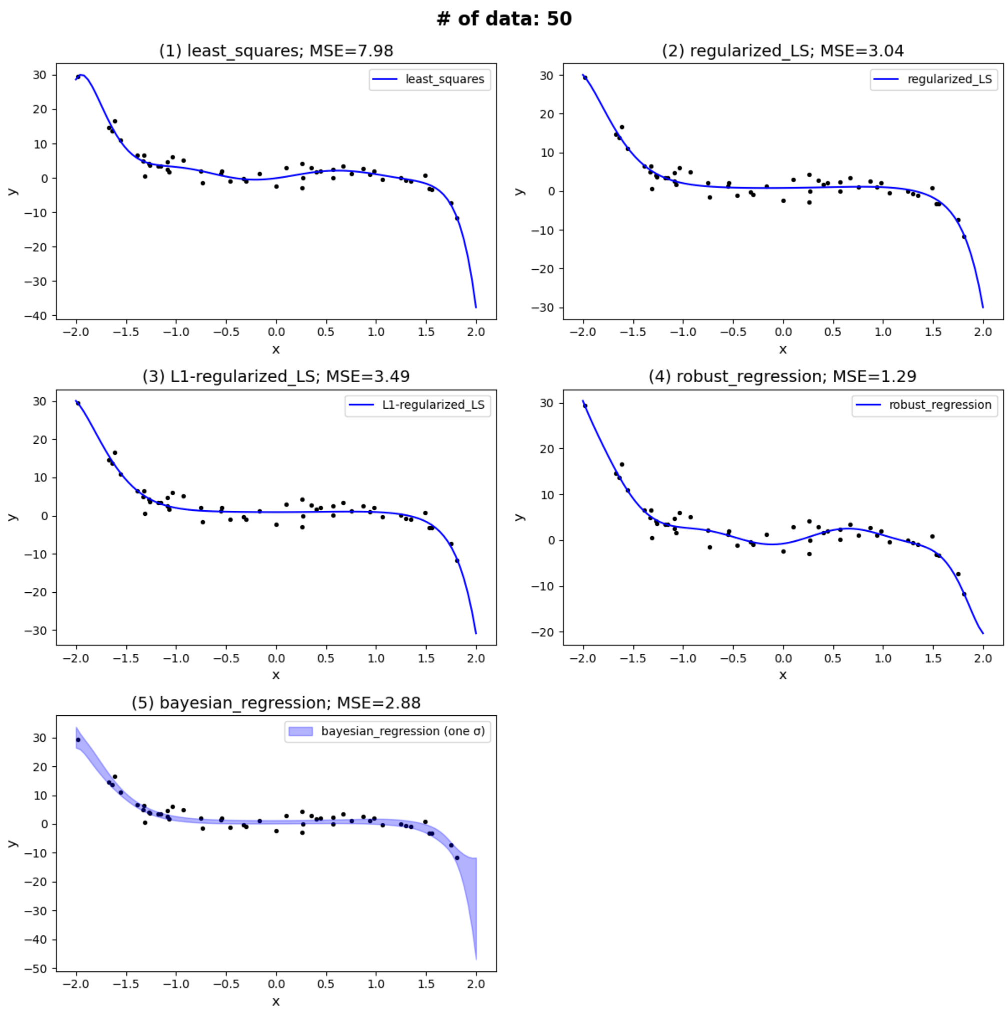

Figure 8: Regression results of 10th order ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

	least_squares	regularized_LS	L1-regularized_LS	robust_regression	bayesian_regression_mu	bayesian_regression_deviation
1	-0.099	0.82	0.903	-0.772	0.735	0.338
2	3.794	0.219	0.0	3.025	0.115	0.571
3	6.491	0.601	0.411	13.149	0.448	0.719
4	-10.745	-0.292	-0.0	-7.576	-0.202	0.717
5	-5.522	0.067	0.0	-19.78	0.137	0.692
6	8.632	-0.265	-0.394	5.642	-0.237	0.661
7	0.654	-0.135	0.0	12.194	0.004	0.648
8	-3.046	-0.192	-0.157	-2.164	-0.227	0.306
9	0.764	0.24	0.201	-3.157	0.154	0.234
10	0.31	0.01	0.004	0.245	0.016	0.013
11	-0.175	-0.056	-0.053	0.291	-0.043	0.008

Figure 9: Model parameters of different methods ($\lambda = 5, \alpha = 1, \sigma^2 = 10$).

2 Part 2 A real world regression problem – counting people

(a)

The mean-squared errors (MSEs) of different regression methods are given in Figure 10. According to the experiment results, we can find that:

(i) *regularized LS* works the best.

(ii) Performance of different methods is similar.

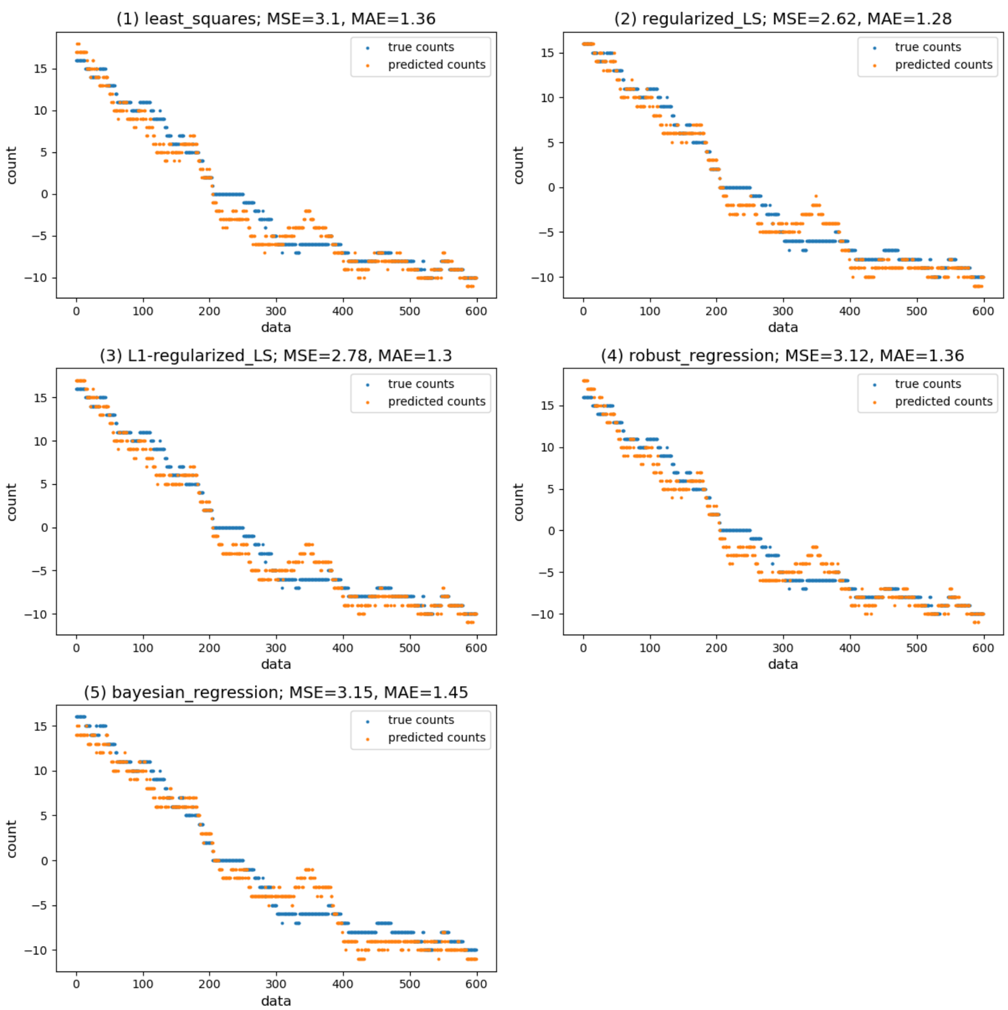

Figure 10: Regression results of different methods with identity mapping function, i.e., $\phi(x) = x$ ($\lambda = 1, \alpha = 1, \sigma^2 = 5$).

(b1)

Here, we tried the 2nd-order polynomial mapping function to transform the input features, i.e., $\phi(x) = \left[x_1, \ldots, x_9, x_1^2, \ldots, x_9^2\right]^T$. Comparing the experiment results in Figures 10 and 11, we can find that:

(i) The 2nd-order polynomial mapping function has a better performance. The strong ability of the 2nd-order function to capture input features may account for this result.

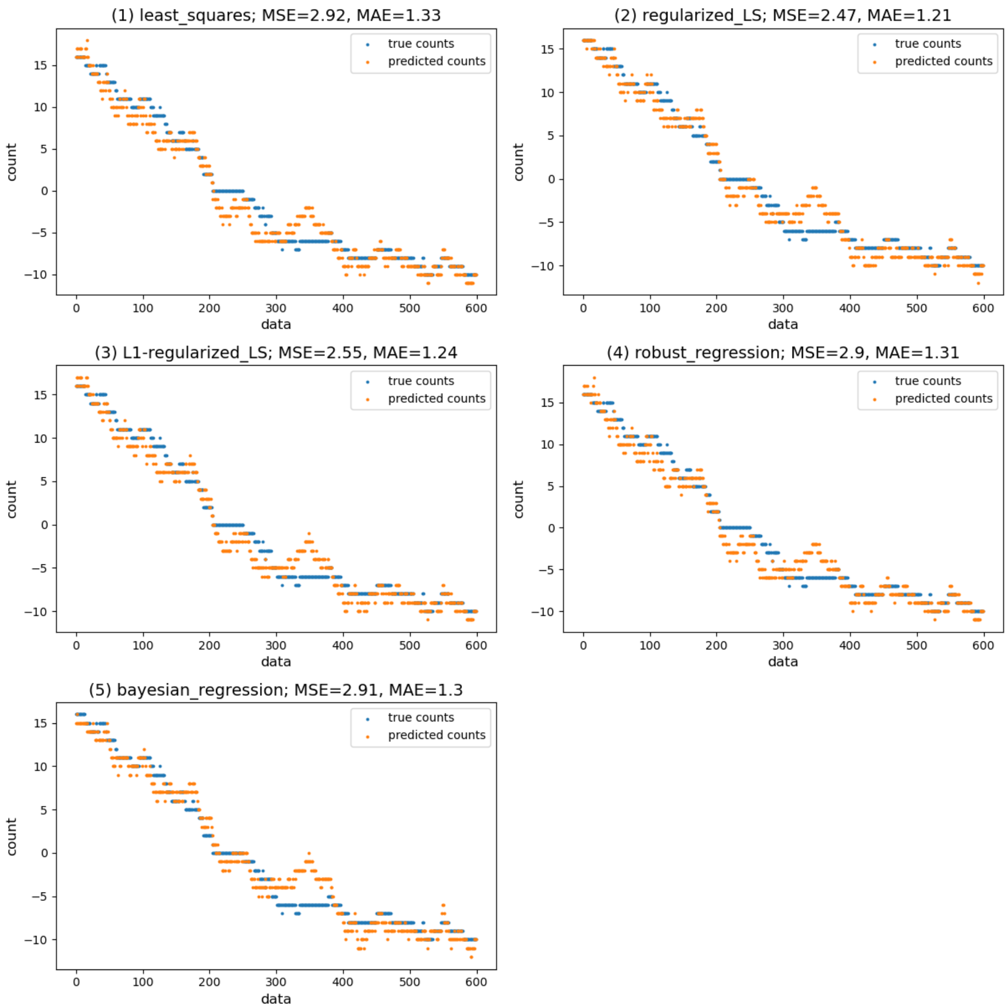

Figure 11: Regression results of different methods with the 2nd-order polynomial mapping function, i.e., $\phi(x) = \left[x_1, \ldots, x_9, x_1^2, \ldots, x_9^2\right]^T$ $(\lambda = 1, \alpha = 1, \sigma^2 = 5)$.

(b2)

Here, we tried the 3rd-order polynomial mapping function to transform the input features, i.e., $\phi(x) = \left[x_1, \ldots, x_9, x_1^2, \ldots, x_9^2, x_1^3, \ldots, x_9^3\right]^T$. Comparing the experiment results in Figures 11 and 12, we can find that:

(i) The 3rd-order polynomial mapping function only brings a small improvement to *L1-regularized LS*.

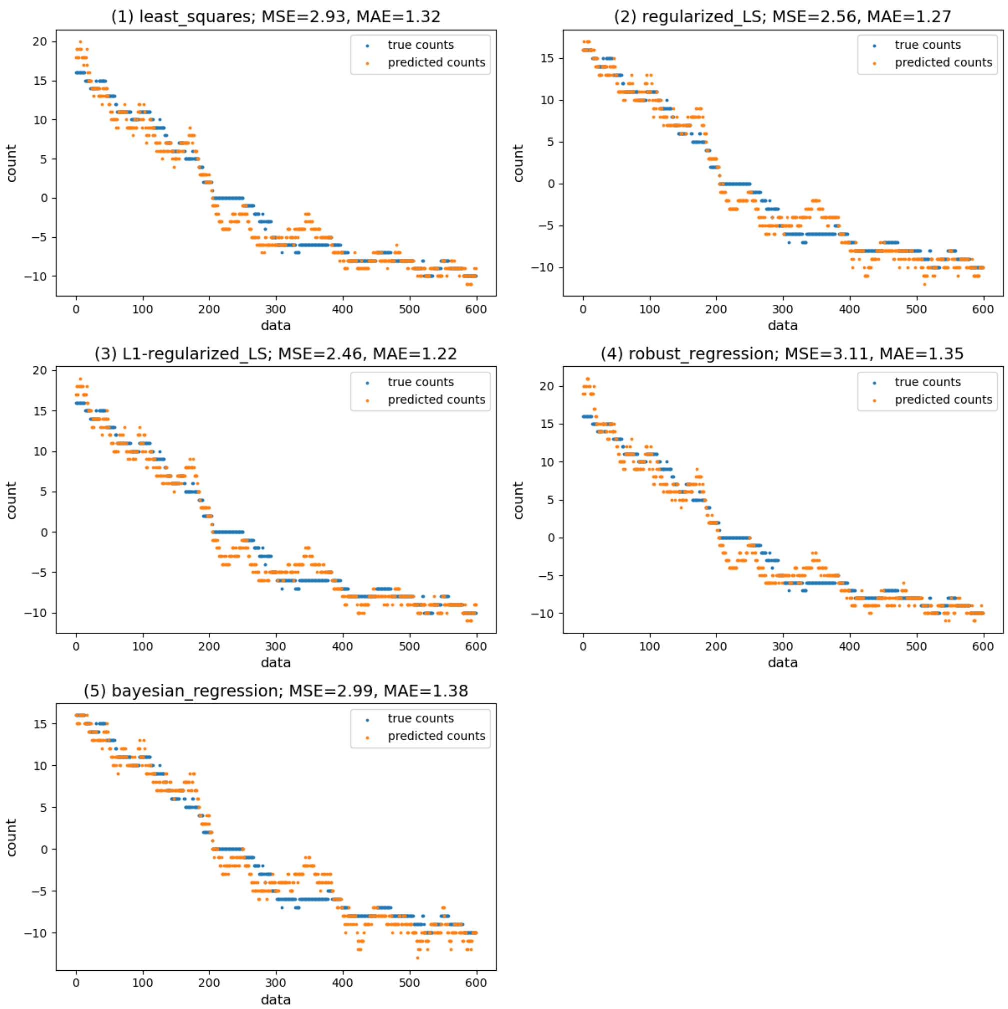

Figure 12: Regression results of different methods with the 3rd order polynomial mapping function, i.e., $\phi(x) = \left[x_1, \ldots, x_9, x_1^2, \ldots, x_9^2, x_1^3, \ldots, x_9^3\right]^T$ ($\lambda = 1, \alpha = 1, \sigma^2 = 5$).