

Report on Programme Assignment 1*

Liu Ziquan
54874777

October 8, 2017

1 Polynomial function

1.1 Task a

For least-squares, Bayesian and regularized least-squares regression, there are closed-form expressions for parameters. But in LASSO and robust regression, we need to adopt convex optimization techniques to get a solution. Initial forms of the two optimization problems cannot be solved by linear programming or quadratic programming so we need to transform the problems into linear programming and quadratic programming.

First, we give the quadratic form of LASSO. The original problem of LASSO is

$$\min_{\theta} \|y - \Phi^T \theta\|^2 + \lambda \|\theta\|_1 \quad (1)$$

Since y is a constant with respect to θ , we can expand the L_2 norm and write the optimization problem as

$$\min_{\theta} \theta^T Q \theta + c^T \theta + \lambda \|\theta\|_1 \quad (2)$$

where $Q = \Phi \Phi^T$ and $c^T = -2y^T \Phi^T$. The $\|\theta\|_1$ is L_1 norm which means sum of absolute values of elements of θ . To obtain L_1 norm of θ , we assume 2 variables θ^+ and θ^- , s.t.

$$\theta_i^+ = \frac{|\theta_i| + \theta_i}{2}, \theta_i^- = \frac{|\theta_i| - \theta_i}{2} \quad (3)$$

So both θ_i^+ and θ_i^- are non-negative numbers and $\theta_i^+ + \theta_i^- = |\theta_i|$. In this way, we can eliminate absolute values in the objective function. But at the same time, we will add constraints on the decision variables. The original optimization problem of LASSO can be written as the following form:

$$\begin{aligned} \min_{\mathbf{x}} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{f}^T \mathbf{x} \\ \text{s.t. } \mathbf{G} \mathbf{x} \leq \mathbf{S} \end{aligned} \quad (4)$$

where $\mathbf{x} = \begin{bmatrix} \theta^+ \\ \theta^- \end{bmatrix}$, $\mathbf{H} = \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix}$, $\mathbf{f}^T = [c^T \ -c^T]$. The inequality constraints ensure that $\theta_i^+, \theta_i^- \geq 0$ and $\theta_i < S_i$, where $\begin{bmatrix} I_D & I_D \\ -I_{2D} & -I_D \end{bmatrix}$, $\mathbf{S} = \begin{bmatrix} \mathbf{s}_D \\ \mathbf{0}_{2D} \end{bmatrix}$. This is a QP problem that can be solved by standard QP solver. The hyperparameter is s instead of λ .

Now we transform robust regression to LP form. $|y_i - \phi_i^T \theta| \leq u_i$ is equivalent to $-u_i \leq y_i - \phi_i^T \theta \leq u_i$. So the original problem can be written as

$$\begin{aligned} \min_{\mathbf{x}} [\mathbf{1}_n^T \quad \mathbf{0}_D^T] \mathbf{x} \\ \text{s.t. } \mathbf{P} \mathbf{x} \leq \mathbf{Y} \end{aligned} \quad (5)$$

where $\mathbf{x} = \begin{bmatrix} u \\ \theta \end{bmatrix}$, $\mathbf{P} = \begin{bmatrix} -I_n & \Phi^T \\ -I_n & -\Phi^T \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} y \\ -y \end{bmatrix}$. This is a LP problem that can be solved by standard LP solver.

*The programme language I use is Python. *cvxopt*, *numpy* and *matplotlib* are used in the programme.

1.2 Task b

In this part, we test the 5 regression algorithms written in Task a. Through testing different hyperparameters in BR, LASSO and regularized LS, we adopt hyperparameters as $\alpha = 1$, $s = 1$, $\lambda = 1.8$. We train the regression algorithms on sample data and test on true data. Figure 1 depicts training and testing result. For BR, standard deviation is plotted as errorbar. The prediction line of LSE and Lasso overlap with the true function line to a large extent. But RR, BR and regularized SE do not overlap with the true function line so perfectly.

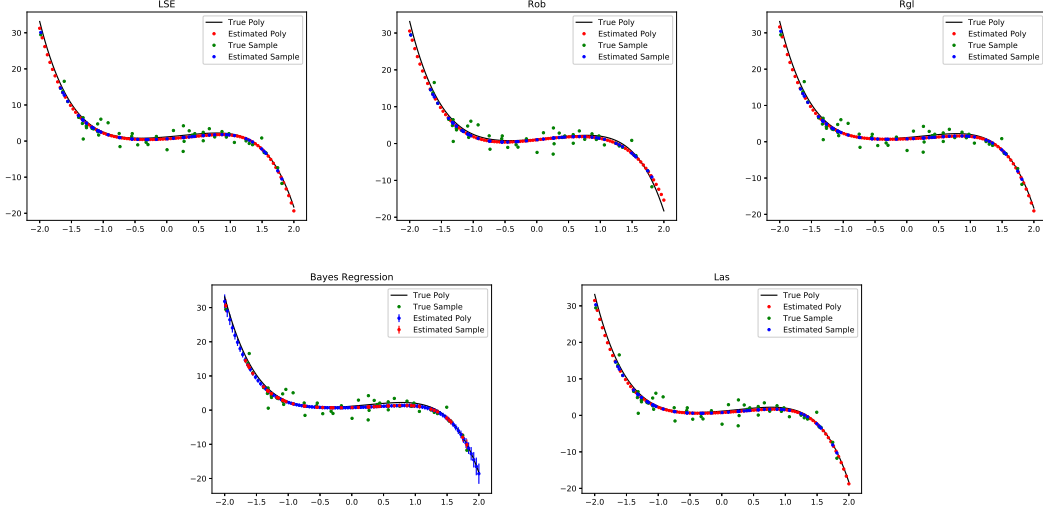


Figure 1: Regression Result of Task b

The mean-squared error(MSE) of 5 algorithms on the test set is shown in Table 1. From the mean-squared error, we can see that Lasso performs best in the 5 algorithms. Robust regression has the lowest accuracy. But RR is the most robust one to outliers, which will be shown in next section. LSE, regularized SE and BR have the same level of accuracy.

RR	LSE	Regularized SE	BR	Lasso
0.7680	0.4086	0.3916	0.4592	0.3596

Table 1: Predict Error on Test Set

1.3 Task c

In this section, we change the size of training set to see which algorithm tends to overfit. First, we sample data randomly from the whole training set, whose size is 50 data points, to generate 4 subsets with 6, 10, 20 and 30 data points respectively. Like in Task b, we plot their results on testing set in Figure 2-5. When training on 6 and 10 points, estimate functions of LSE and RR are far from true function, which means LSE and RR are not robust with less data. BR, regularized SE and Lasso perform well on less data, which means the 3 methods are robust to insufficient data.

The above result is obtained through once random sample from original data so the result includes random factors. To get more stable result, we run random sample 5000 times at each subset size from 8 data points to 50 data points and plot mean-squared error(MSE) versus training size in Figure 6. LSE and RR have the same feature: when training size is small, prediction error is very large while when training size reaches a certain amount, prediction error will become relatively small. BR is the most accurate algorithm among the 5 when training on a small data set. Regularized SE has the same accurate level as BR but a little worse. Lasso's error is larger than BR and regularized SE on less training data. But when training size becomes larger, Lasso's accuracy improves faster than BR and regularized SE.

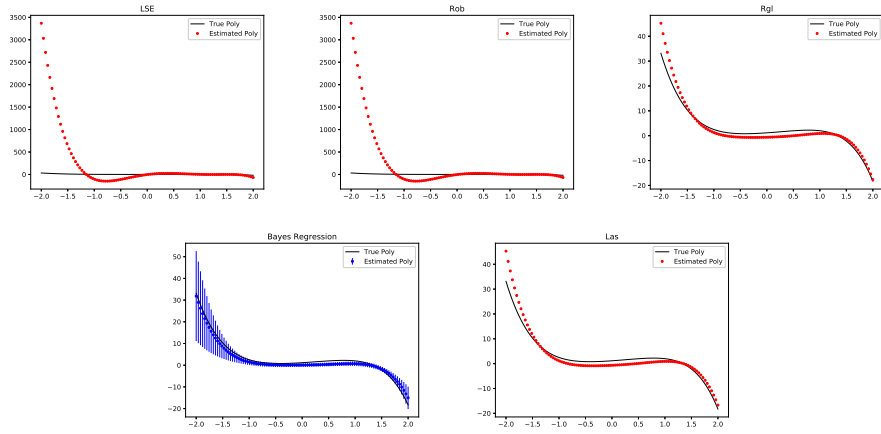


Figure 2: Training on 6 points

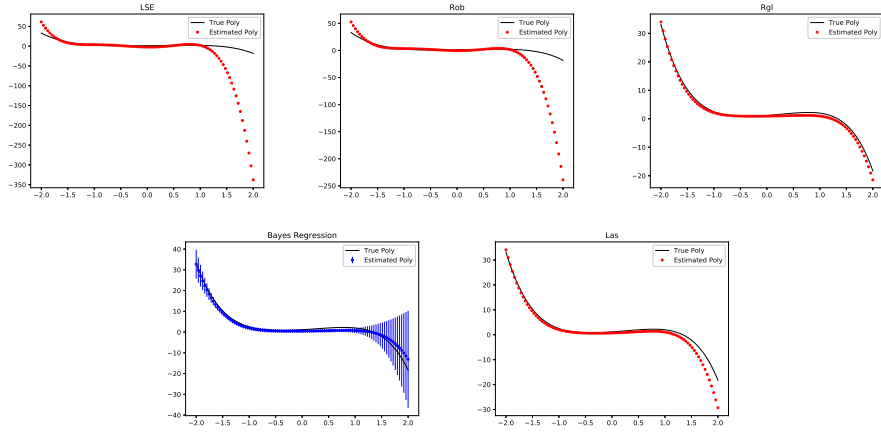


Figure 3: Training on 10 points

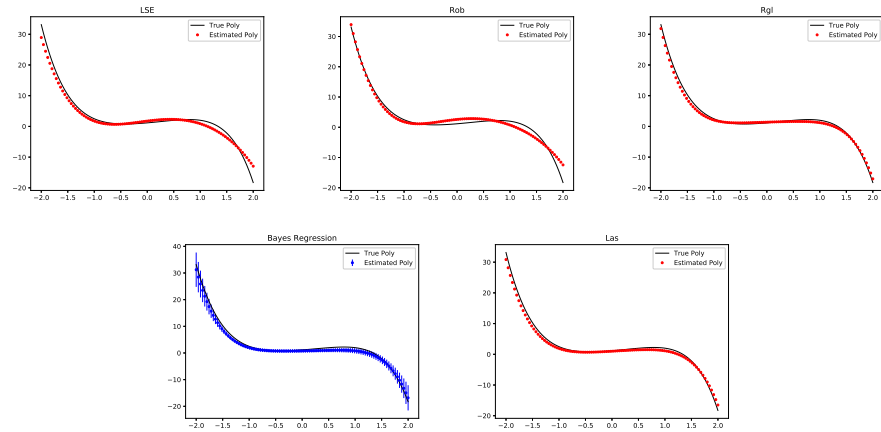


Figure 4: Training on 20 points

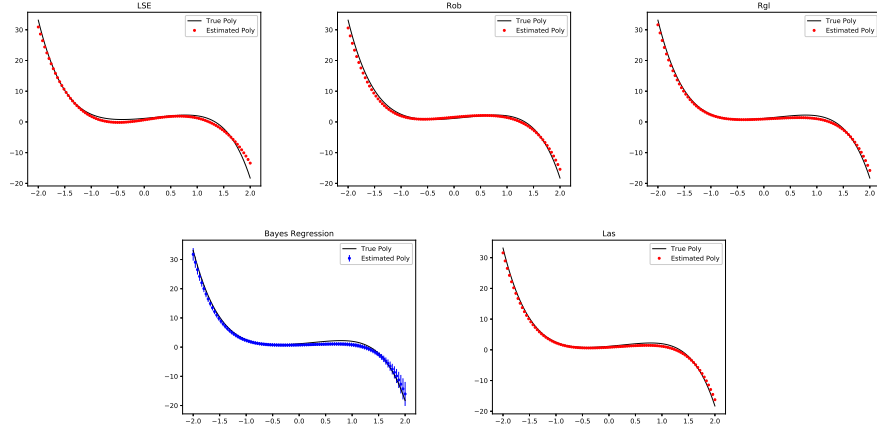


Figure 5: Training on 30 points

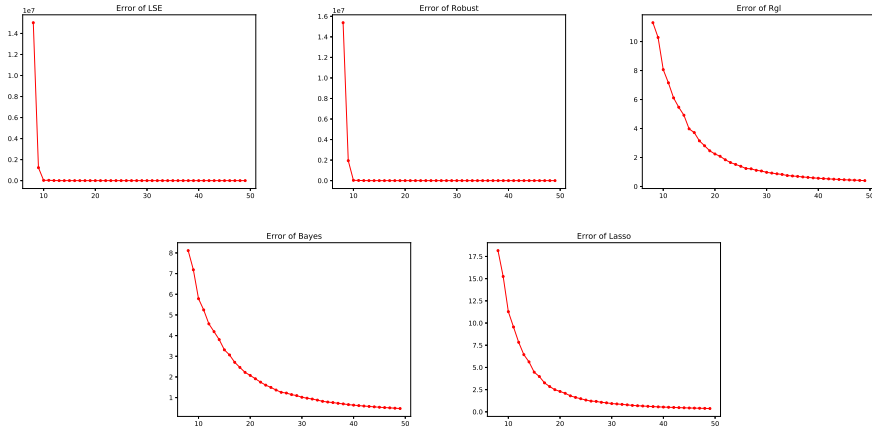


Figure 6: Error vs. training data size

1.4 Task d

In this section, we set some outliers to the training set to test robustness of regression algorithms to abnormal values. In the experiment, we randomly select 5 data points from training set and set their y values to be randomly large compared to normal sample values. The result is shown in Figure 7. MSE of each model is also given in figures

The result shows that LSE, regularized SE and Bayes Regression are all very sensitive to outliers since there are no overlap between estimate function and true function. The most sensitive algorithm is LSE. Estimate function of Lasso has a small region which is overlapped with true function. RR is the most robust algorithm to outliers among the 5 methods. The reason for robustness of RR is that when there are abnormal values in training set, the residual is measured by L_1 norm, which makes it less sensitive to large residuals compared to L_2 norm. That is why LSE is most sensitive to outliers: it only tends to minimize L_2 norm of residual, which is very large when there are outliers in training set. When the residual of outliers is large, it will affect the optimization result to a very large extent.

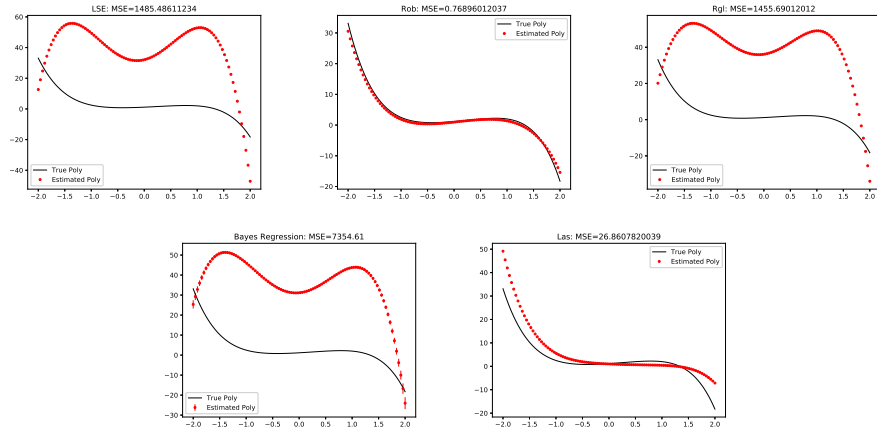


Figure 7: Result on Training Set with Anomaly

1.5 Task e

Task e is to try high-order polynomial to model the function and to see which algorithm is more prone to overfitting. To do this task, we just need to change the dimension of feature and perform regression. We obtain the following result when using 10th order polynomial in Figure 8. Both prediction function and MSE are given in the figure. Lasso is the worst algorithm considering its largest MSE. But the estimate function of Lasso is not complex because of its regularization term. Regularized SE and BR also learned less complex functions than LSE and RR because of regularization and their MSEs are at the middle level. Both LSE and RR learned complex functions but LSE's MSE is much larger than RR's, which shows RR's robustness to complex model.

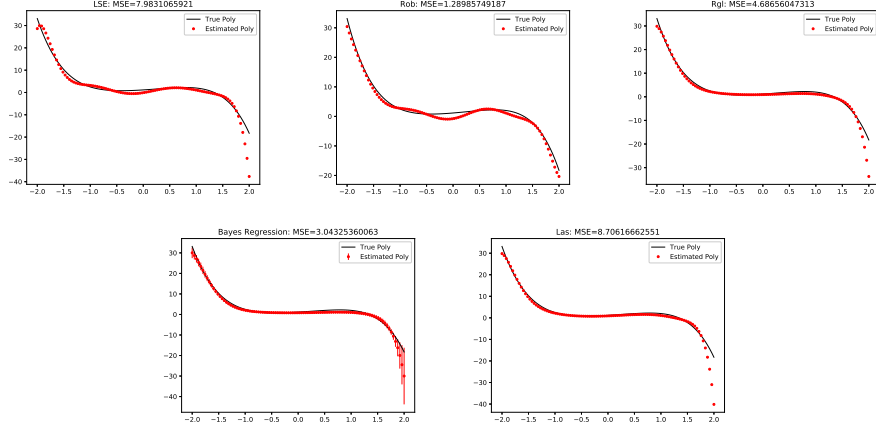


Figure 8: Result of 10th order

2 Counting People

Based on regression model in Part 1, we learn a regression function and predict number of people in an image. Since we already have regression codes, we can directly use the 'Regressor' class to do regression in this part.

2.1 Task a

We set feature $\phi(x) = x$ and do regression using the above 5 models. Then test the regression function on test set, we compare the prediction values and true values in Figure 9. MSE and MAE of each algorithm are also given in the figure. From MSE and MAE result, we can see that regularized SE works best among the 5 methods. BR is a little worse than regularized SE. LSE and RR's accuracies are at the same level. Lasso's accuracy is the lowest one.

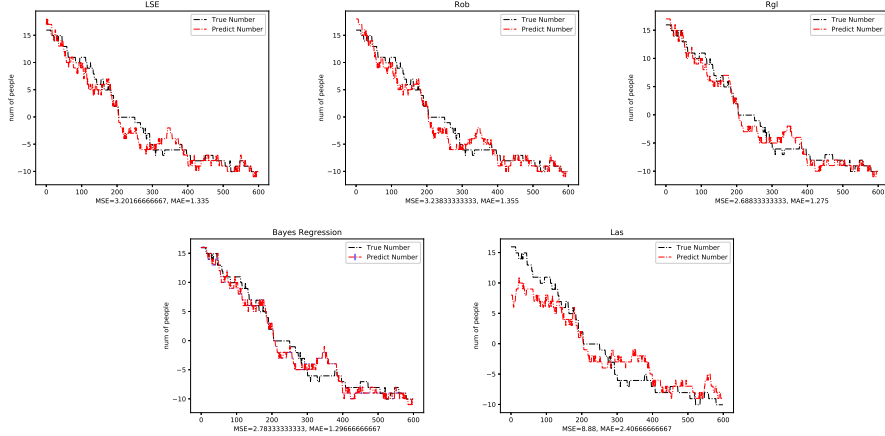


Figure 9: Prediction and True Counting Using x as Feature

2.2 Task b

In this task, we use different features to do regression. Specifically, we use 2nd and 3rd order polynomial as features, i.e. $\phi(x) = [x_1, \dots, x_9, x_1^2, \dots, x_9^2]^T$ and $\phi(x) = [x_1, \dots, x_9, x_1^2, \dots, x_9^2, x_1^3, \dots, x_9^3]^T$. Results of the two features are shown in Figure 10 and 11. We also try add some cross-terms $x_i x_j$ but the result is not improved but degenerates, so we do not give cross-term feature's result. 2nd order polynomial feature improves all methods' accuracy. But 3rd order polynomial feature does not improve accuracy of

2nd order in terms of RR and BR. Lasso's accuracy is improved largely when using 3rd order polynomial. Regularized SE is always the best approach in this task.

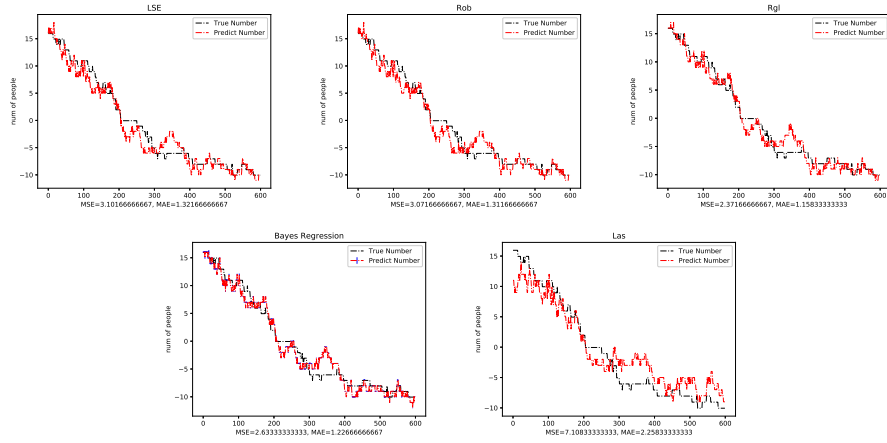


Figure 10: Prediction and True Counting Using 2nd Feature

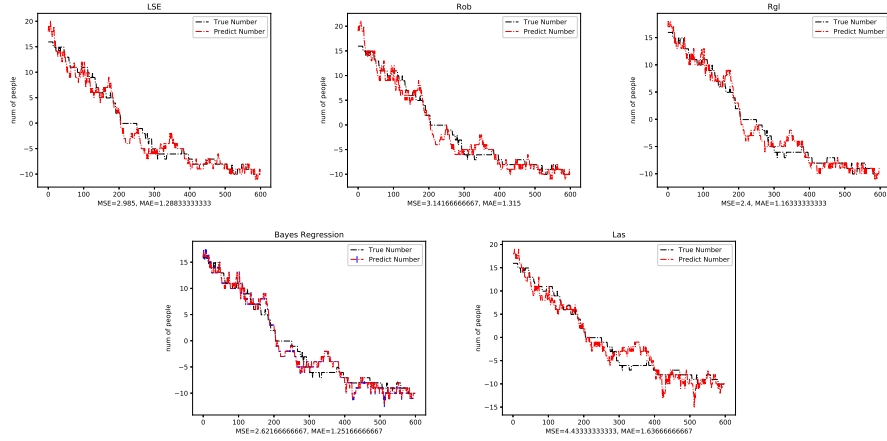


Figure 11: Prediction and True Counting Using 3rd Feature