Junyi CHEN

CS5487

2017/11/04

CS5487 Programming Assignment 2: clustering

# PART 1 pOLYNOMINAL FUNCTION

## implementation

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

## Run the algorithms on the three synthetic datasets

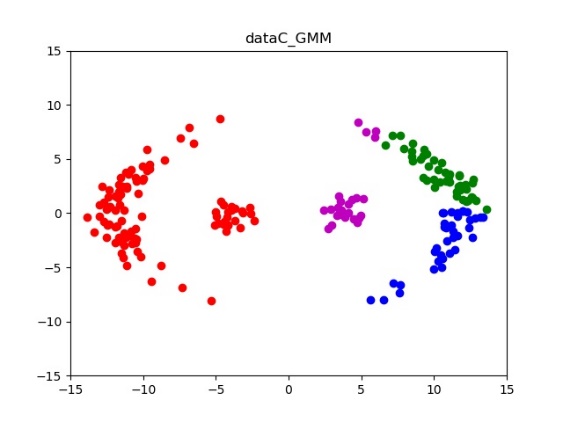
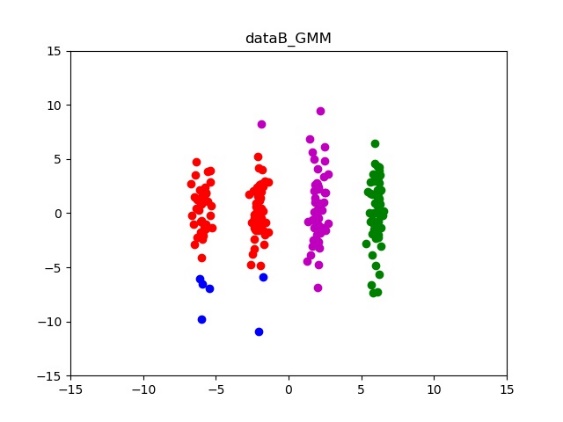
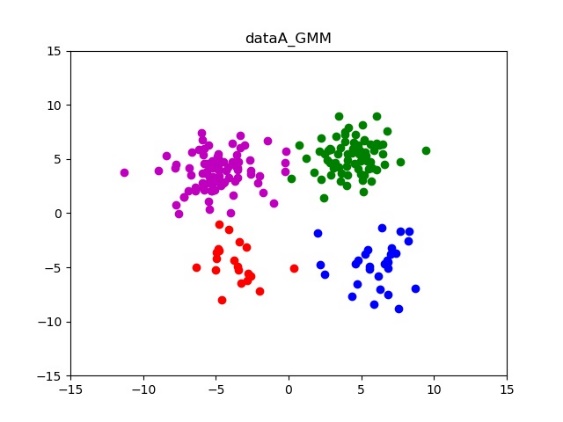
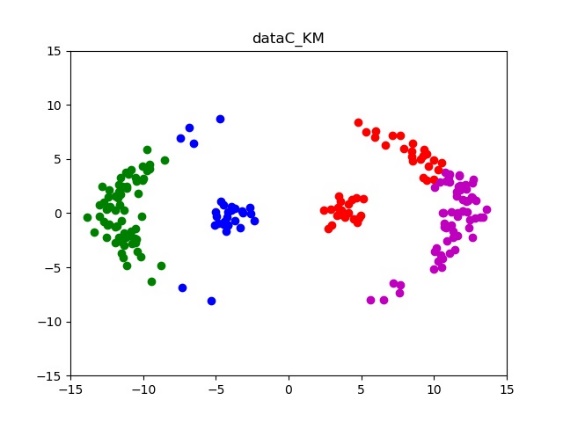
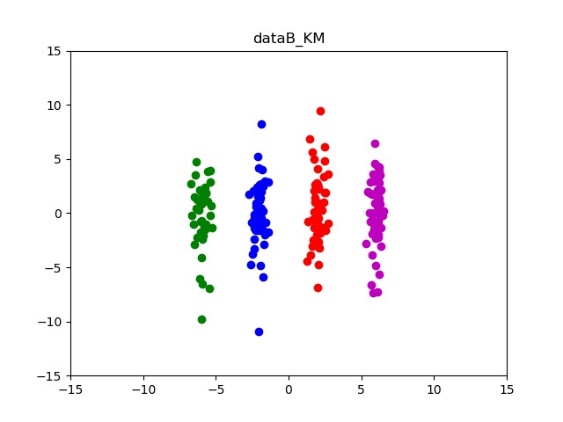
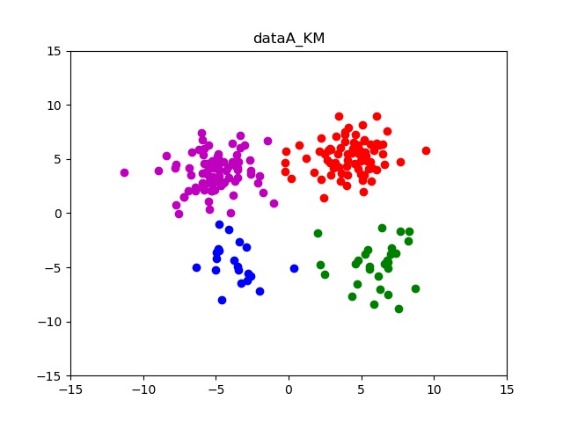


Figure 1: Predict of different clusters, the first 3 figures are K-means while the latter 3 figures are Gaussian mixture model

In terms of K-means (KM) and Gaussian mixture model (GMM) the performance of data A is similar, both of them can correctly cluster four sets of data points. In terms of data B, K-means out puts better performance than Gaussian mixture model because GMM fail to discriminate the first and the second column wise data points. However, both KM and GMM do not work well in data C, KM cannot handle with points stay in the margin from he leftmost cluster. While GMM fail to separate the first and the second from left hand side.

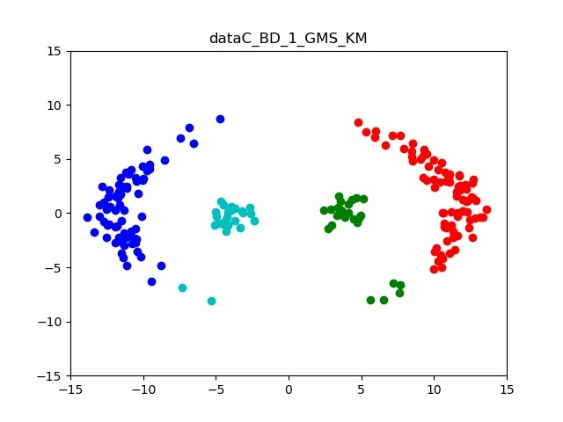
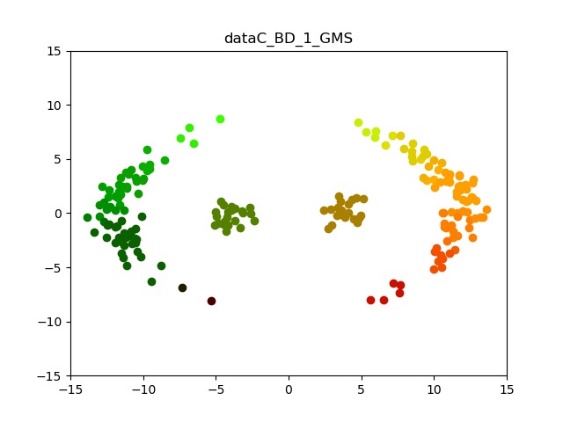
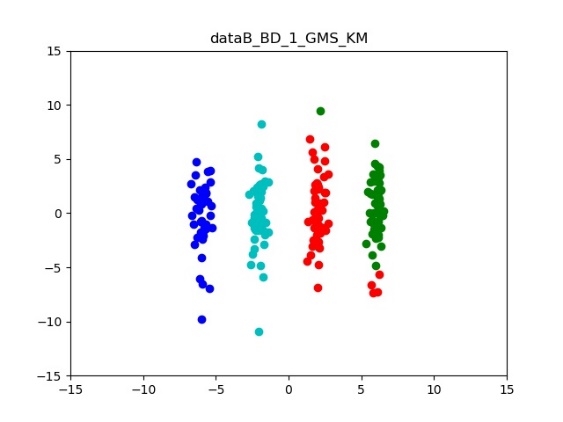
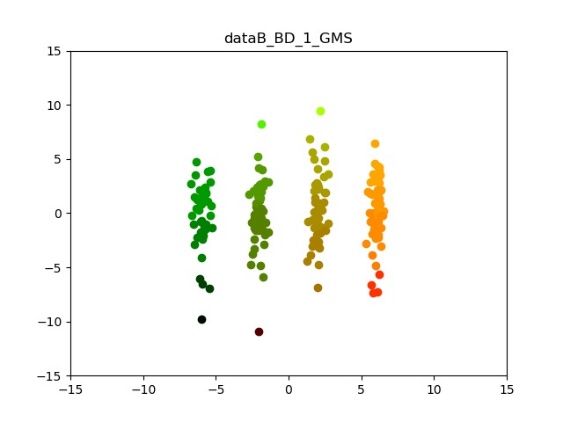
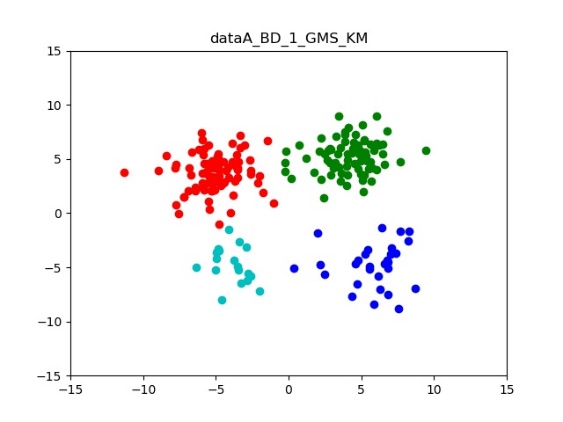
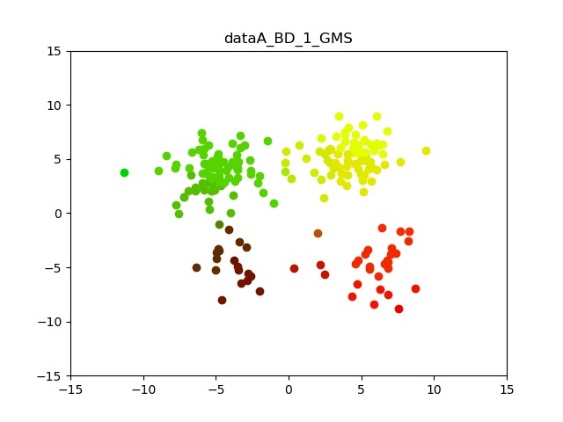


Figure Predict of mean shift clustering with bandwidth equals to 1, color is displayed according to the value of local peak (left) and plots of local peak values clustered by K-means(right)

Figure 2 are cluster result of mean shift clustering (MS). Because MS calculate the value of local peak, not represented by cluster center. To visualize the clustering result, the data point in figure 2 are colored by the value of local peak. Hence, points converge to the same peak will have similar color in figure 2. We can see that MS is working good on data A for the colors in 4 clusters are well separated. In terms of data B and C, colors of the 4 clusters can basically be discriminated. Anyway, from the plot of data B of MS, data points from the bottom seems to form their own cluster like GMM. Plots form right hand side is picture we apply KM on the output local peaks of MS. The reason why we do the second clustering is to limit the number of clusters. Qualitatively, MS is not as good as KM in data A and B but it works well in data C.

In summary, distance based KM works good in data A and B, and it is easy to implement. But the performance is not good using KM when the densities between clusters are not equal like data C. Also, cluster number is fixed and affiliation of data points is hard. One point belongs to one clusters only. GMM can output probability of cluster affiliation, but the cluster may not be as precise as KM and the initialization of estimated values potentially affect clustering result. Cluster number of GMM is also fixed. MS works well in our data A, B and C, and the representation is not by cluster center so it is flexible. But the limitation is that the stagey of cluster prototyping will affect the performance of cluster and the speed of GMM is much slower than KM and GMM.

## SENSITIVITY OF MEAN SHIFT BANDWIDTH

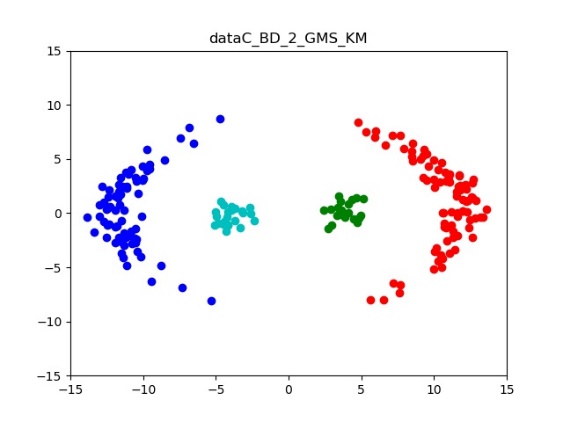
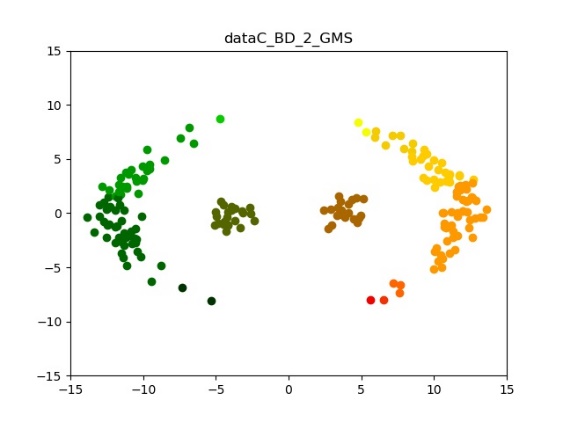
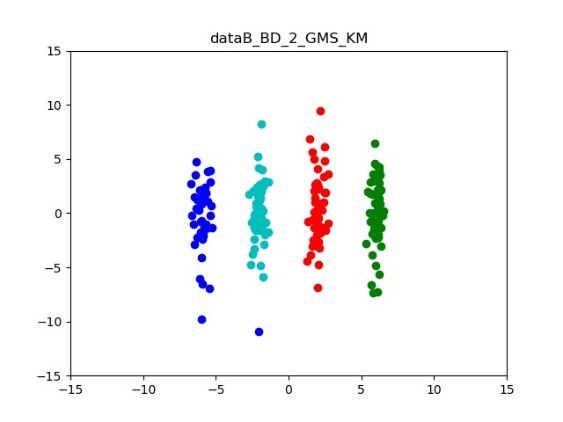
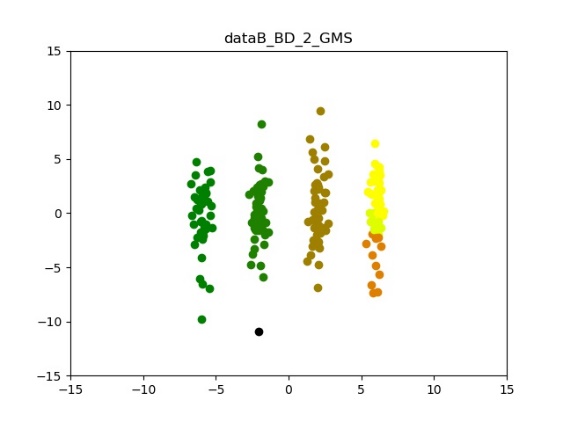
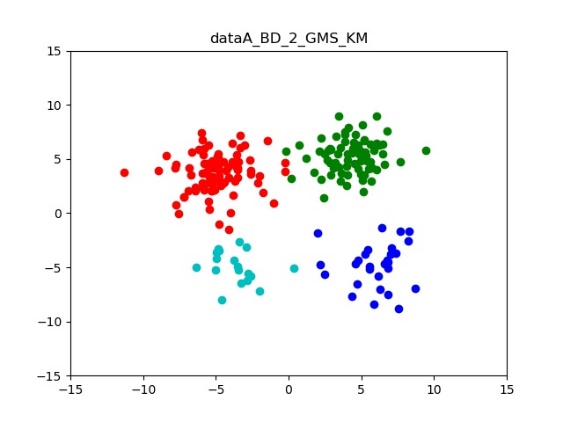
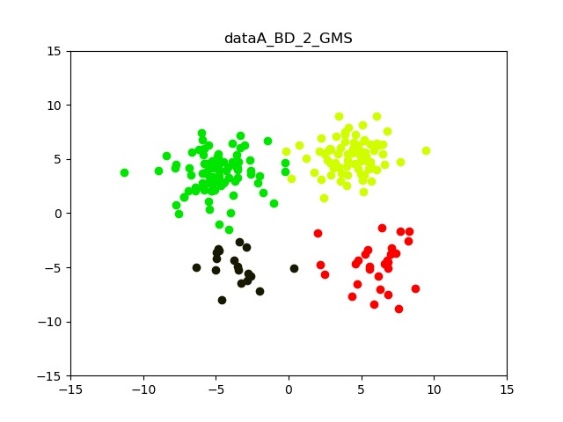


Figure Predict of mean shift clustering with bandwidth equals to 2, color is displayed according to the value of local peak (left) and plots of local peak values clustered by K-means(right)

To analysis the sensitivity of MS bandwidth. We tested different bandwidth values on three datasets. By increasing the bandwidth to 2, the clustering performance is better than 1.

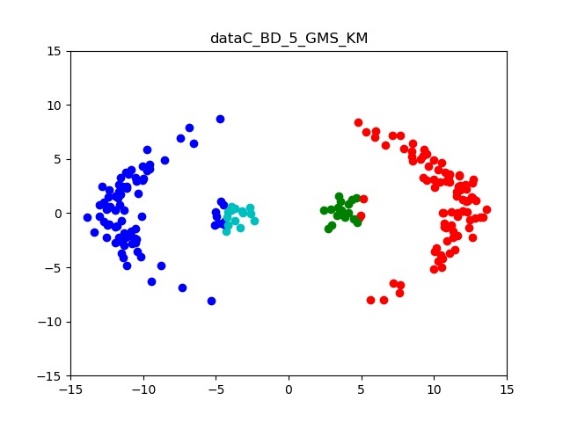
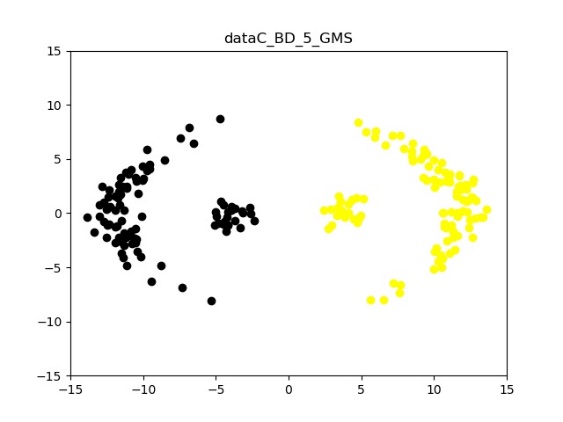
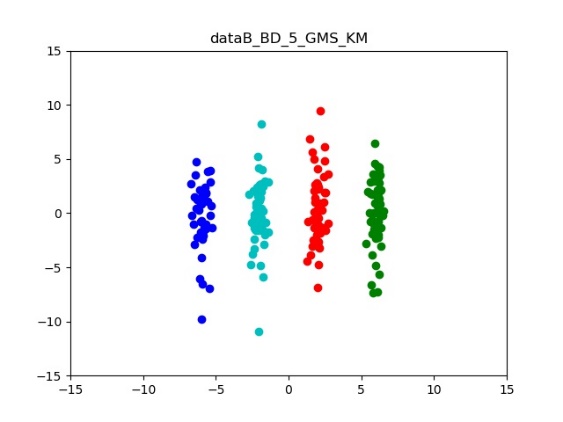
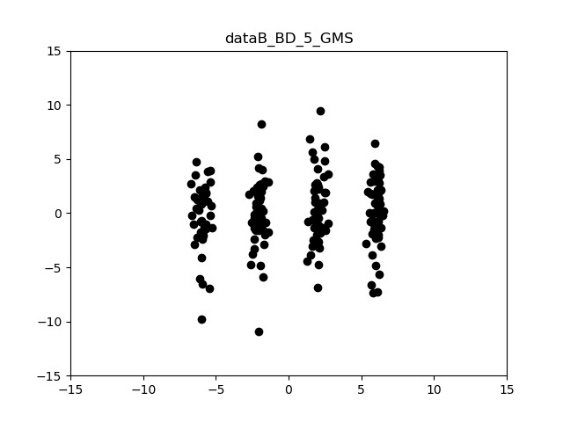
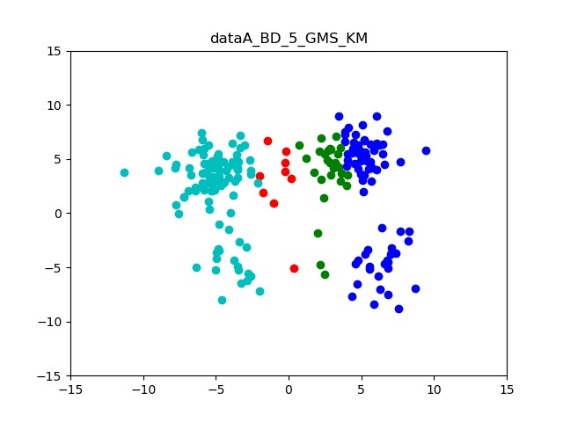
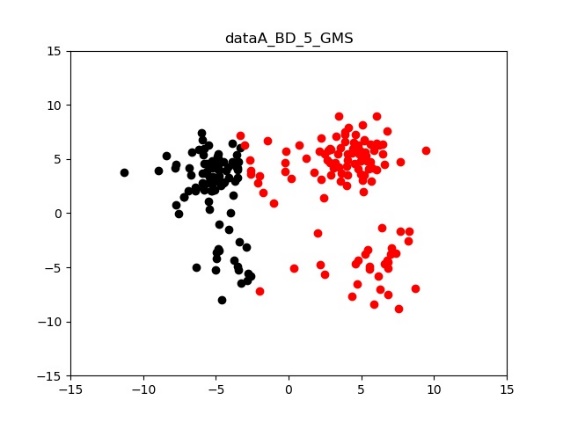


Figure Predict of mean shift clustering with bandwidth equals to 5, color is displayed according to the value of local peak (left) and plots of local peak values clustered by K-means(right)

## outlies and robustity analysis

In this section, 4 outliers are added to the training set: (outliers\_x = [-1.3,0.5,0.7,1]; outliers\_y = [80,30,50,-30]). Hyper-parameters are set as we examined before. Table 2 and Figure 4 show some statistic of outliers experiments.

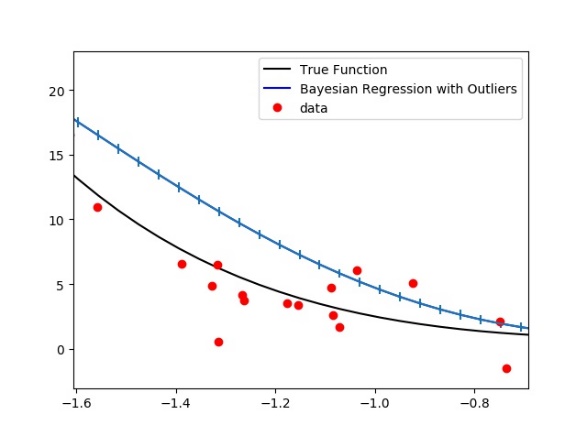
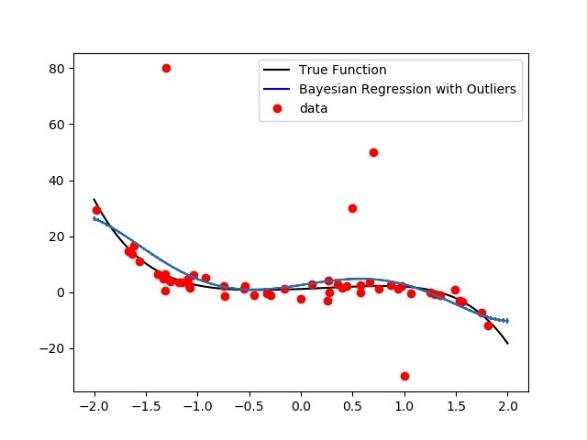
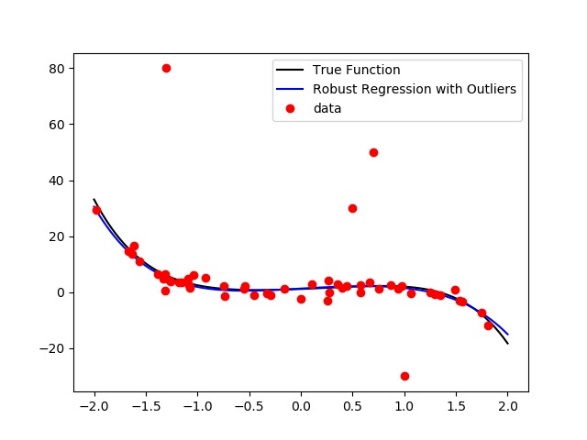
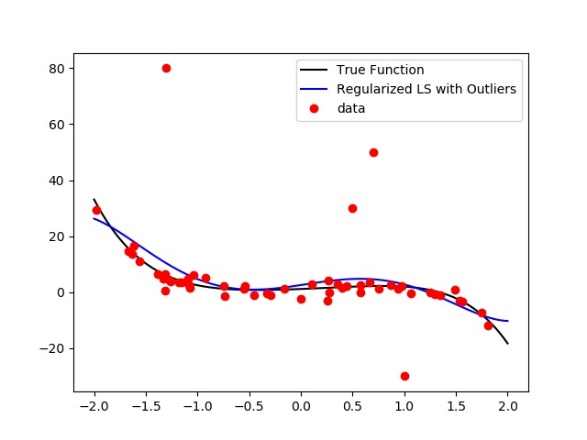
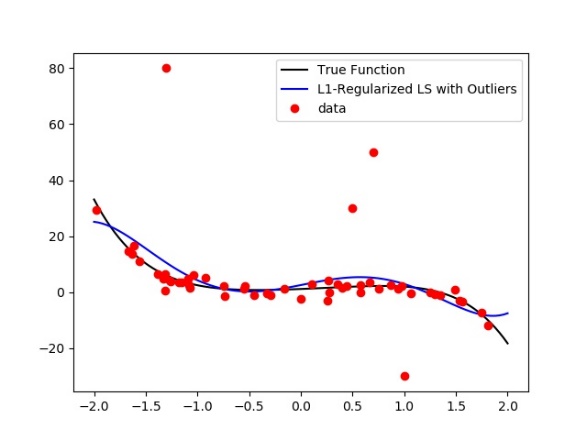
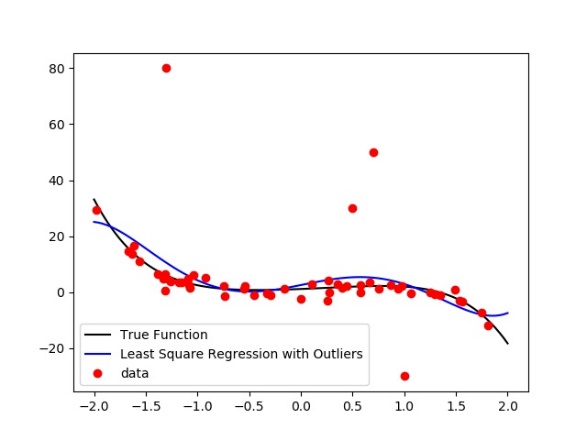


Figure 5 Predict function with outliers of different regression methods (figure in lower-right corner is the zoomed plot to show deviation of Bayesian Regression)

From figure 4, we can see that RR have the best performance to resist outliers for it has a similar curve with normal experiment. Also, MSE of RR stays at around 0.8 this is another evidence that RR is robust when some outliers are added to the dataset. RLS and BR have similar MSE around 7 while LASSO and LS have MSE around 9.5. In conclusion, RR is the robust to outliers LS is the most sensitive one.

One possible reason is that the objective function of LS is square formed. Outliers with unexpected large values will amplify the error function a lot so the regression method tend to be sensitive to the outliers.

## HIGHER ORDER ESTIMATION

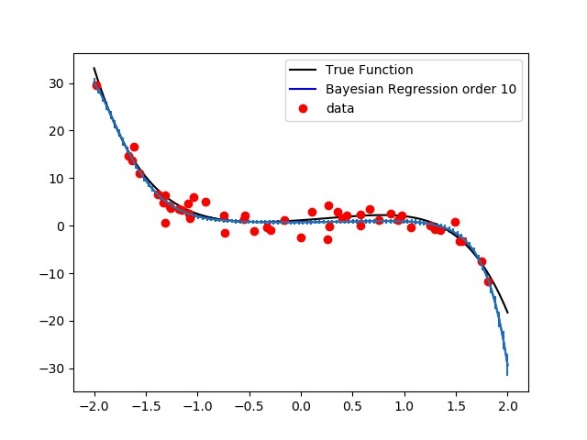
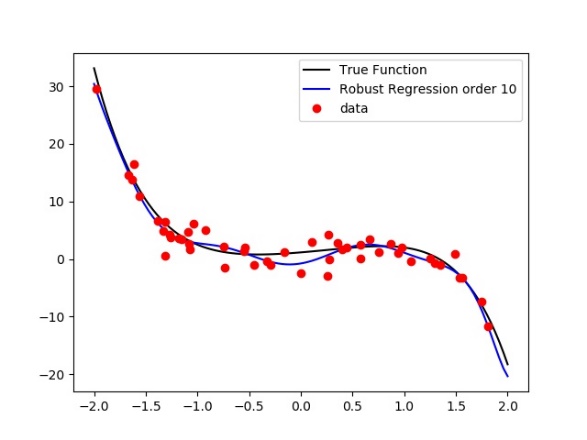
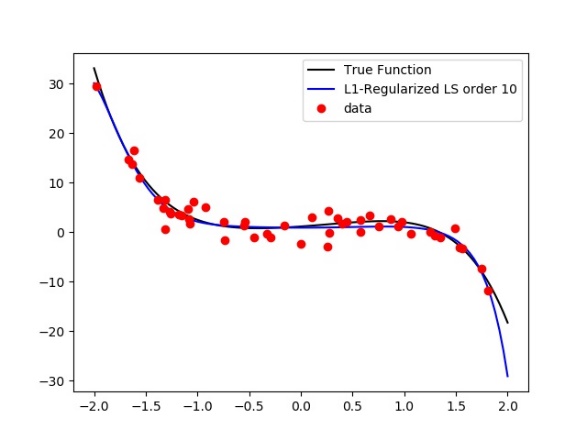
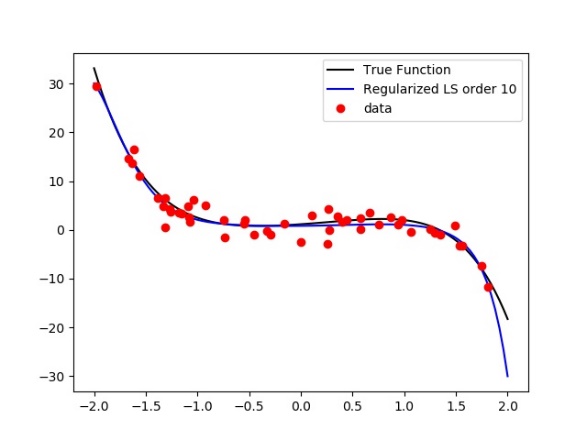
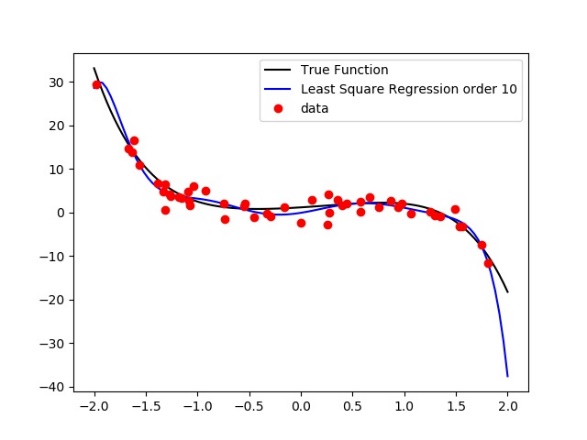


Figure 6 Predict function of different regression methods (order 10)

From Table 3, we can conclude that RR have the lowest MSE while LS outputs the largest MSE. RLS, LASSO and BR share similar numbers. But when we refer to Figure 5, the RR curve is twisted and not ‘close’ (visually) to the true function. 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.

Table 4 is a list of predicted theta values of different. The yellow part of the table theta of order higher than 5. 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 thetas should have small theta when the order is higher than Phenomena on LS and RR, infer that two estimators overfit the dataset.

Table 3 Experiment mean-square errors of different hyper-parameters (order 10)

|  |  |  |
| --- | --- | --- |
|  | MSE of LS | MSE of RR |
| NA | 7.983107 | 1.289857 |

|  |  |
| --- | --- |
| Alpha and sigma | MSE of BR |
| 'alpha': 0.1, 'sigma': 0.1 | 17.82521114 |
| 'alpha': 0.1, 'sigma': 0.5 | 3.890454631 |
| 'alpha': 0.1, 'sigma': 1 | **2.876626237** |
| 'alpha': 0.1, 'sigma': 5 | 7.669525294 |
| 'alpha': 0.5, 'sigma': 0.1 | 15.91938565 |
| 'alpha': 0.5, 'sigma': 0.5 | 9.98381222 |
| 'alpha': 0.5, 'sigma': 1 | 4.351475446 |
| 'alpha': 0.5, 'sigma': 5 | 4.518810872 |
| 'alpha': 1, 'sigma': 0.1 | 13.49094405 |
| 'alpha': 1, 'sigma': 0.5 | 14.03001502 |
| 'alpha': 1, 'sigma': 1 | 6.548096165 |
| 'alpha': 1, 'sigma': 5 | 3.49799502 |
| 'alpha': 5, 'sigma': 0.1 | 9.573704116 |
| 'alpha': 5, 'sigma': 0.5 | 18.18799337 |
| 'alpha': 5, 'sigma': 1 | 15.22605129 |
| 'alpha': 5, 'sigma': 5 | 3.043253601 |

|  |  |  |
| --- | --- | --- |
| Lambda | MSE of RLS | MSE of LASSO |
| 0.1 | 17.82521 | 16.64525 |
| 0.25 | 14.03002 | 19.16953 |
| 0.5 | 9.983812 | 13.3456 |
| 1 | 6.548096 | 6.142715 |
| 2 | 4.351475 | **2.686592** |
| 5 | **3.043254** | 3.493389 |

Table 4 Theta values of different methods (some rows of LASSO is missing, possibly due to precision issues)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | LS | RLS | LASSO | RR | BR | True |
| 0 | -0.09938 | 0.819574 | 0.912094 | -0.77214 | 0.73481 | 1.15243 |
| 1 | 3.793512 | 0.219007 | 5.23E-08 | 3.025203 | 0.115234 | 1.48629 |
| 2 | 6.49122 | 0.601251 | 0.550202 | 13.14885 | 0.448381 | 0.92950 |
| 3 | -10.7448 | -0.29162 | NA | -7.57593 | -0.20169 | -1.11344 |
| 4 | -5.52168 | 0.067251 | NA | -19.78 | 0.13653 | 0.15980 |
| 5 | 8.632264 | -0.26496 | -0.35734 | 5.642387 | -0.23723 | -0.61788 |
| 6 | 0.653714 | -0.13463 | 2.48E-10 | 12.19446 | 0.00411 | 0 |
| 7 | -3.0458 | -0.19206 | -0.19844 | -2.1642 | -0.22735 | 0 |
| 8 | 0.763549 | 0.240247 | 0.171415 | -3.15724 | 0.15408 | 0 |
| 9 | 0.310281 | 0.00966 | 0.014124 | 0.245386 | 0.016373 | 0 |
| 10 | -0.17514 | -0.05584 | -0.04541 | 0.290522 | -0.04301 | 0 |

# Part 2 A real world regression problem – counting people

## ORIGINAL FEATURE SET ANALYSIS

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

Table 5 Errors of different methods using original feature set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | BR | LASSO | LS | RLS | RR |
| MSE | 2.618734 | **2.464461** | 3.102838 | 2.618734 | 3.118997 |
| MAE | 1.282433 | **1.256468** | 1.358444 | 1.282433 | 1.364567 |

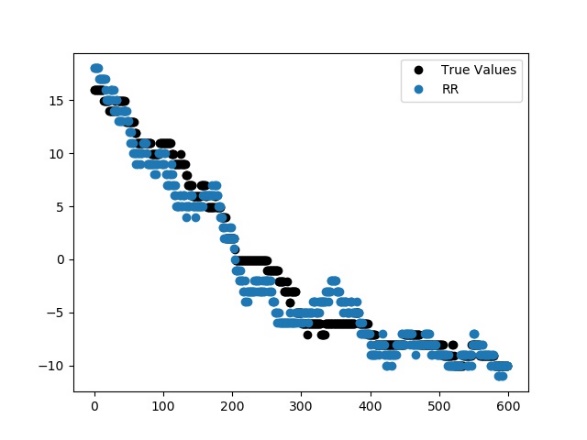
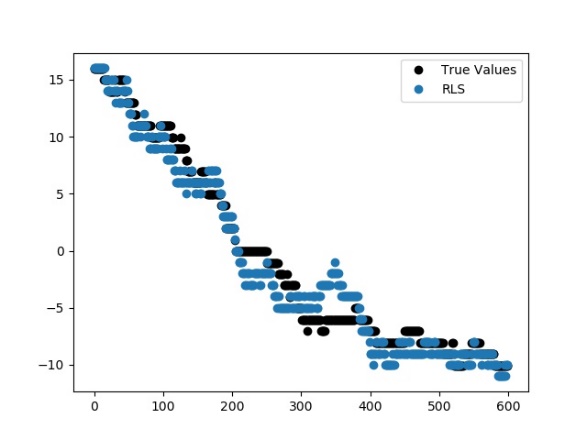
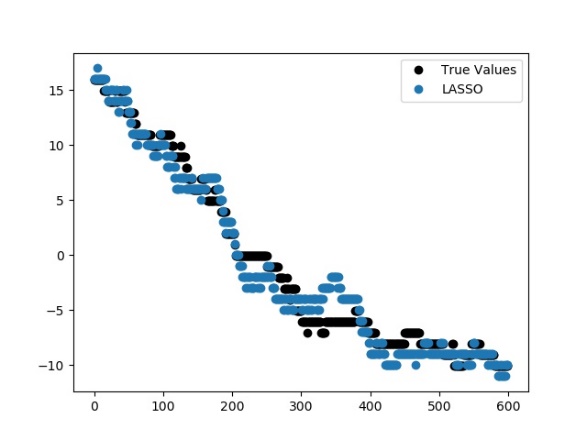
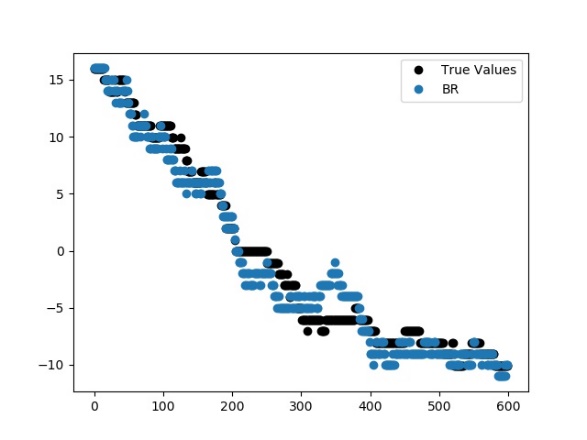
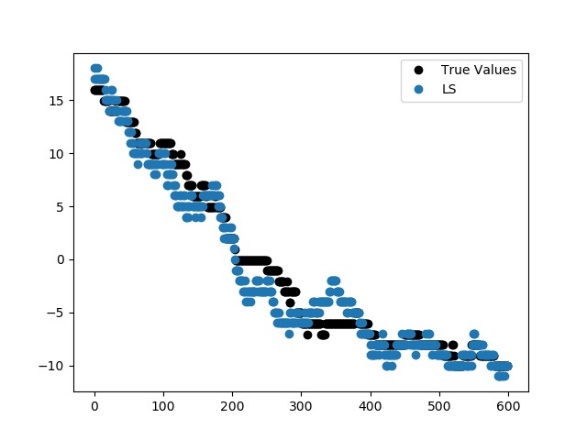
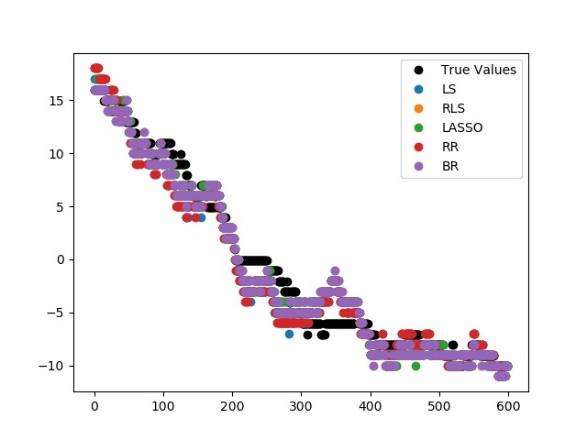


Figure 7 Plot of predict data and true number (figure on the upper left represent different colors represent different methods)

From Table 5, we can see that LASSO contribute the smallest MSE and MAE using the original feature. From the plots, we can see that shape of prediction plots are similar across all methods, the difference between different methods is not significant. On the other hand, most wrong predictions locate in the region between -5 to 0 (sample 200-400). If we can improve the performance in this region, the performance may increase.

## Other feature transformation