Progress Report

2021/04/19

Student: Ho Chi En

Bayesian regression

My perspective:

Symptoms to let the doctor make a precise decision

The more effective symptoms, the more probability to find the actual diseases

Given data

1_data.mat

Workspace

Name	Value
t	*100x1 double*
x	*100x1 double*

$\mathbf{x} = \{x_1, x_2, ..., x_{100} | 0 \leq x_i \leq 2\}$ and $\mathbf{t} = \{t_1, t_2, ..., t_{100}\}$

Data Preprocessing

Step 1: similar to normalize to append the dataset x into 3 features

Step 2: data pass the linear function (Sigmoidal)

Data Preprocessing

Step 1

	X
1	0.51
2	1.2
3	1.7
4	1.81
5	1.85
6	1.17
7	1.14
8	1.62
9	1.61
10	1.55
11	1.91
12	0.71
13	1.73
14	0.17
15	0.69

$$\frac{x-\mu_j}{s}, \mu_j = \frac{2j}{M}$$

```
[[  5.1         -1.56666667  -8.23333333]
 [ 12.           5.33333333  -1.33333333]
 [ 17.          10.33333333   3.66666667]
 [ 18.1         11.43333333   4.76666667]
 [ 18.5         11.83333333   5.16666667]
 [ 11.7          5.03333333  -1.63333333]
 [ 11.4          4.73333333  -1.93333333]
 [ 16.2          9.53333333   2.86666667]
 [ 16.1          9.43333333   2.76666667]
 [ 15.5          8.83333333   2.16666667]
 [ 19.1         12.43333333   5.76666667]
 [  7.1          0.43333333  -6.23333333]
 [ 17.3         10.63333333   3.96666667]
 [  1.7         -4.96666667 -11.63333333]
 [  6.9          0.23333333  -6.43333333]]
```

$$\begin{cases} x_0 = \dfrac{x - \left(\frac{2\cdot 0}{M}\right)}{s} \\ x_1 = \dfrac{x - \left(\frac{2\cdot 1}{M}\right)}{s} \\ x_2 = \dfrac{x - \left(\frac{2\cdot 2}{M}\right)}{s} \end{cases}$$

$$(M, s) = (3, 0.1)$$

Data Preprocessing

Step 2 $Sigmoidal(x) = \dfrac{1}{1 + e^{-x}}$

```
[[  5.1          -1.56666667  -8.23333333]
 [ 12.            5.33333333  -1.33333333]
 [ 17.           10.33333333   3.66666667]
 [ 18.1          11.43333333   4.76666667]
 [ 18.5          11.83333333   5.16666667]
 [ 11.7           5.03333333  -1.63333333]
 [ 11.4           4.73333333  -1.93333333]
 [ 16.2           9.53333333   2.86666667]
 [ 16.1           9.43333333   2.76666667]
 [ 15.5           8.83333333   2.16666667]
 [ 19.1          12.43333333   5.76666667]
 [  7.1           0.43333333  -6.23333333]
 [ 17.3          10.63333333   3.96666667]
 [  1.7          -4.96666667 -11.63333333]
 [  6.9           0.23333333  -6.43333333]]
```

$\dfrac{1}{1 + e^{-x}}$

```
[[9.93940199e-01 1.72692104e-01 2.65578804e-04]
 [9.99993856e-01 9.95195247e-01 2.08608527e-01]
 [9.99999959e-01 9.99967471e-01 9.75075573e-01]
 [9.99999986e-01 9.99989172e-01 9.91563092e-01]
 [9.99999991e-01 9.99992742e-01 9.94328797e-01]
 [9.99991706e-01 9.93525146e-01 1.63374240e-01]
 [9.99988805e-01 9.91279616e-01 1.26382089e-01]
 [9.99999908e-01 9.99927607e-01 9.46173837e-01]
 [9.99999898e-01 9.99919994e-01 9.40847748e-01]
 [9.99999814e-01 9.99854230e-01 8.97215975e-01]
 [9.99999995e-01 9.99996016e-01 9.96879593e-01]
 [9.99175575e-01 6.06669356e-01 1.95905257e-03]
 [9.99999969e-01 9.99975901e-01 9.81415475e-01]
 [8.45534735e-01 6.91813645e-03 8.86550904e-06]
 [9.98993229e-01 5.58070106e-01 1.60450637e-03]]
```

Bayesian regression

$${S_0^{-1}}_{(3,3)} = 10^{-6.} \cdot I_{(3,3)}$$

$${m_0}_{(3,1)} = 0_{(3,1)}$$

$${S_N^{-1}}_{(3,3)} = {S_0^{-1}}_{(3,3)} + \beta \cdot \left({X^T}_{(3,N)} X_{(N,3)}\right)_{(3,3)}$$

$${S_N}_{(3,3)} = \left({S_N^{-1}}_{(3,3)}\right)^{-1}$$

$${m_N}_{(3,1)} = {S_N}_{(3,3)} \left[{S_0^{-1}}_{(3,3)} {m_0}_{(3,1)} + \beta \cdot \left({X^T}_{(3,N)} T_{(N,1)}\right)_{(3,3)}\right]_{(3,1)}$$

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}\right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}. \end{aligned}$$

mind the shape of the matrix

Generate 5 sample curves

Step 1: generate enough points from 0~2

Step 2: doing the same normalization

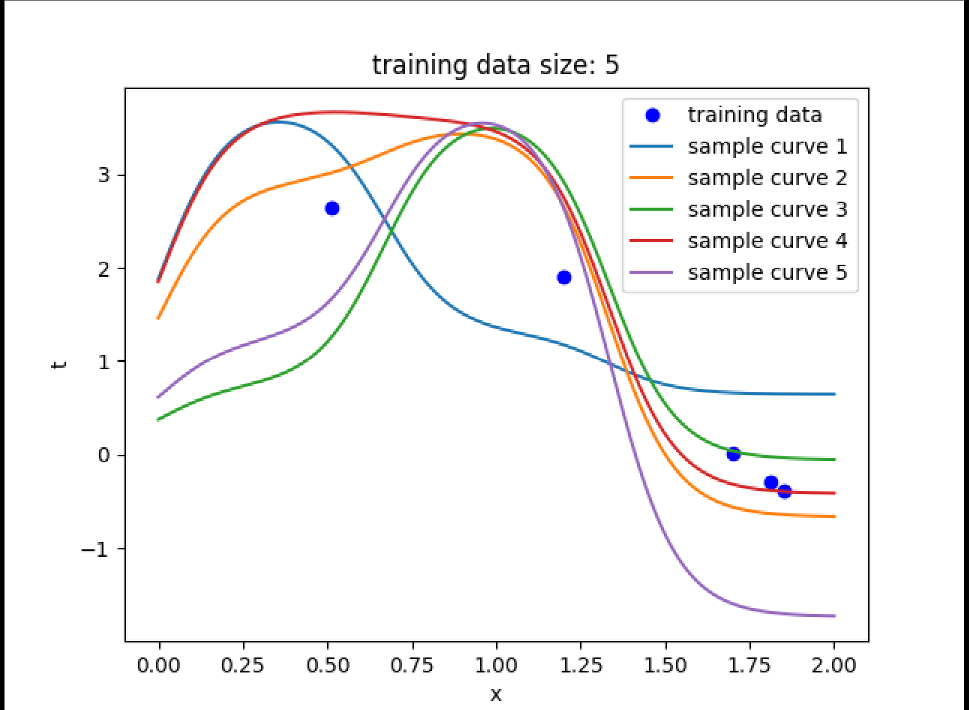
Step 3: sigmoidal again

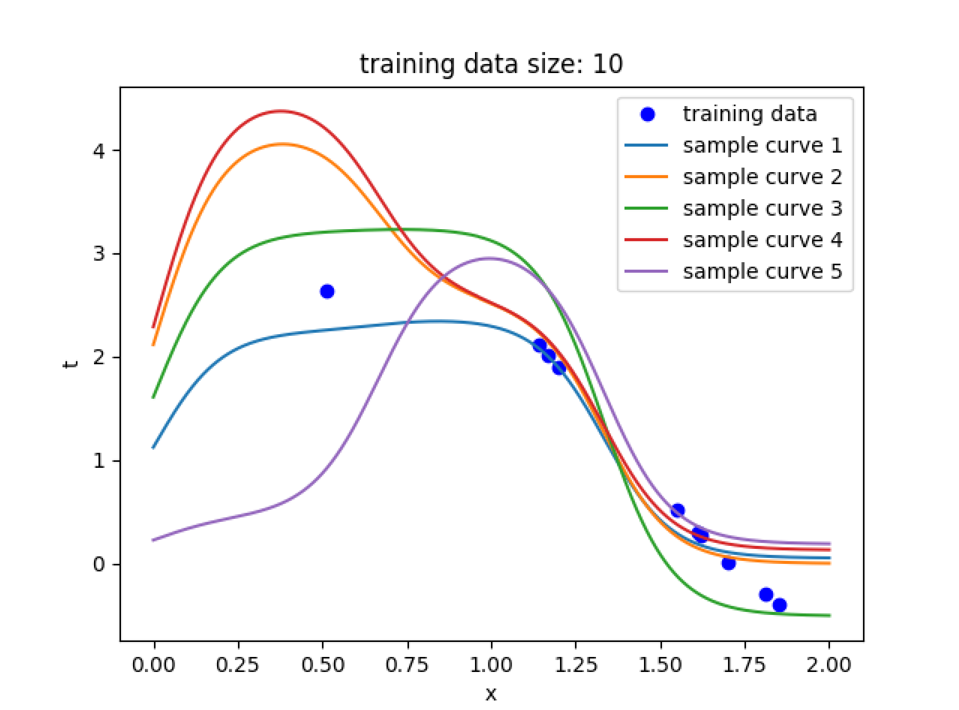
Step 4: generate weight by np.random.multivariate_normal(mean,cov)

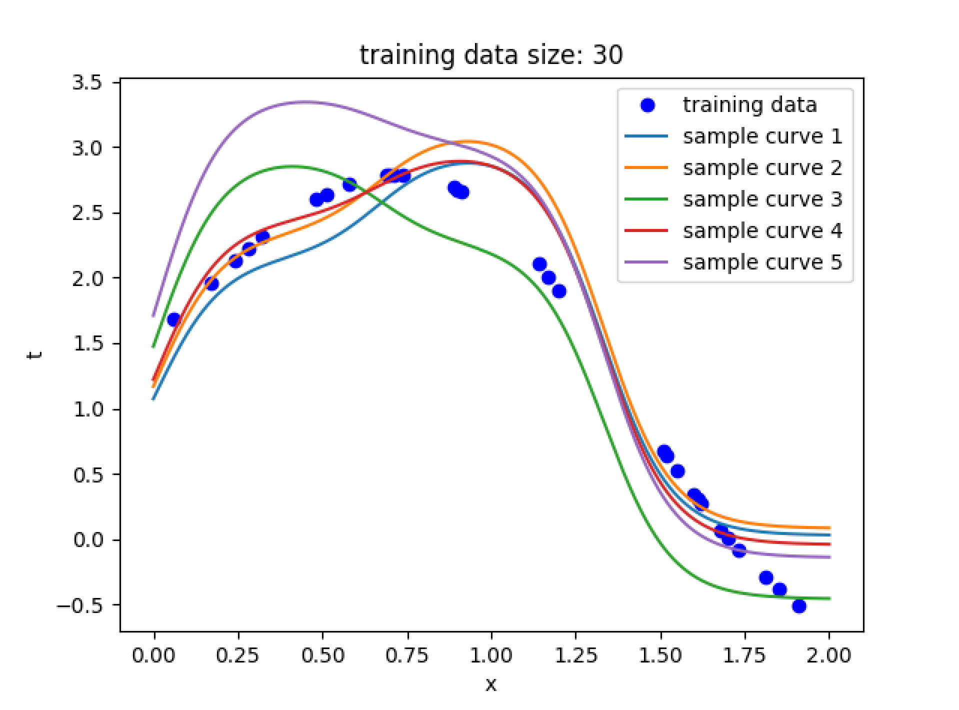
Step 5: $y_{(N',1)} = x_{preprocess_{(N',3)}} \cdot weight_{(3,1)}$

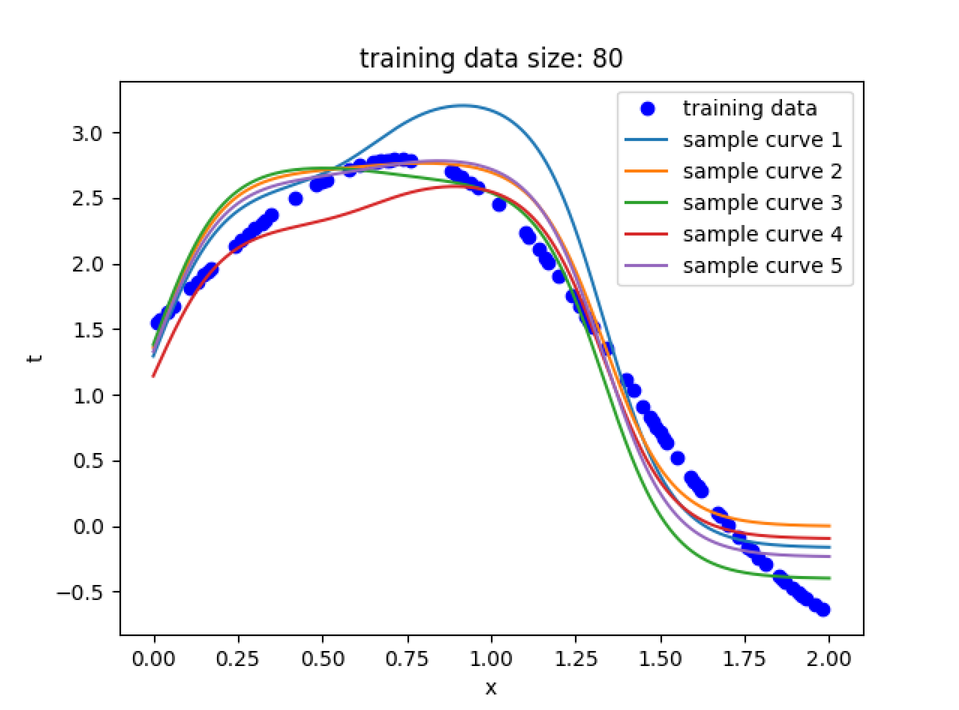
$m_{N\,(3,1)}.reshape(3,)$ $\quad S_{N\,(3,3)}$

Results

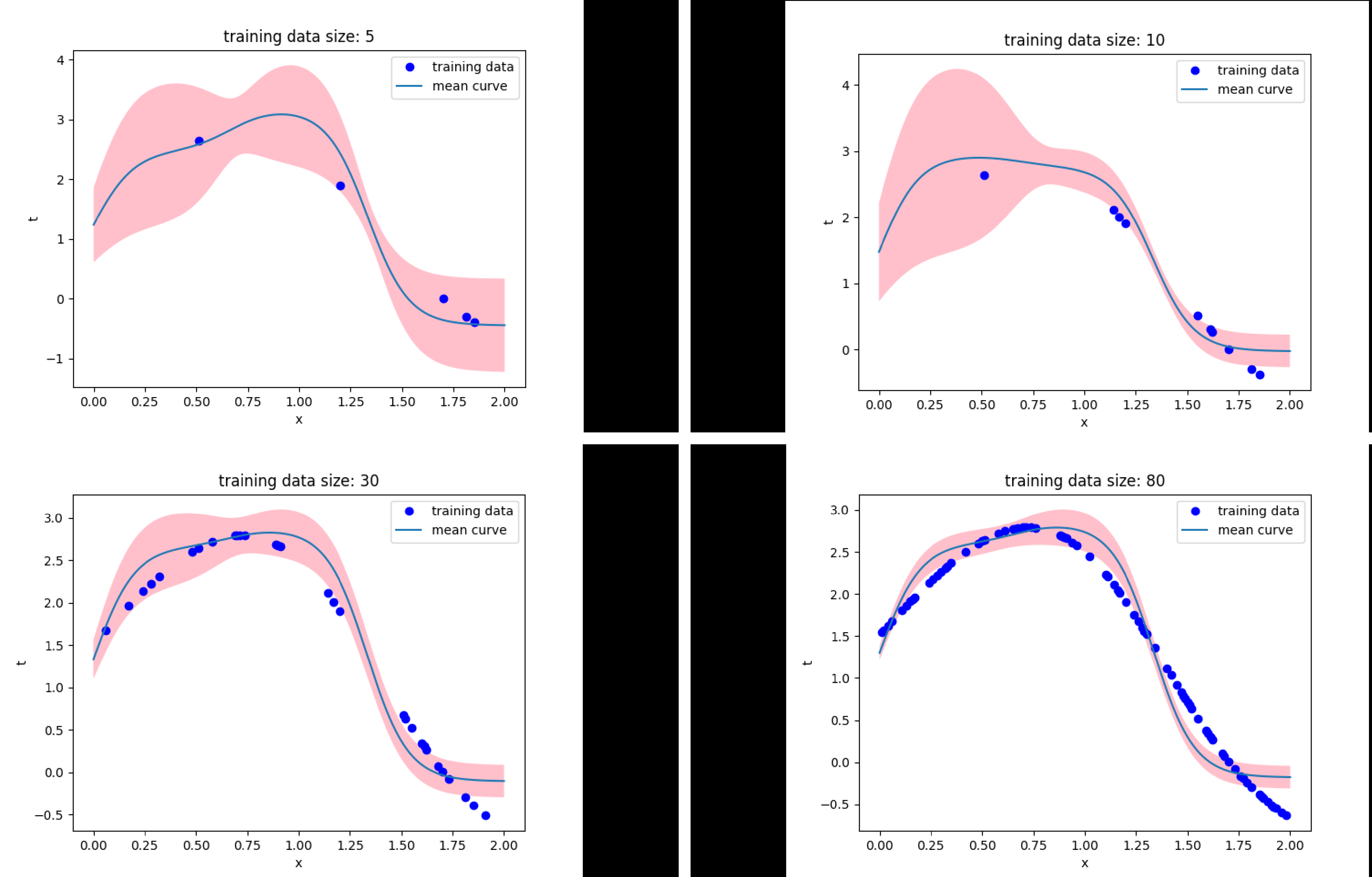

training data size: 5
training data
sample curve 1
sample curve 2
sample curve 3
sample curve 4
sample curve 5
t
x


training data size: 10
training data
sample curve 1
sample curve 2
sample curve 3
sample curve 4
sample curve 5
t
x


training data size: 30
training data
sample curve 1
sample curve 2
sample curve 3
sample curve 4
sample curve 5
t
x

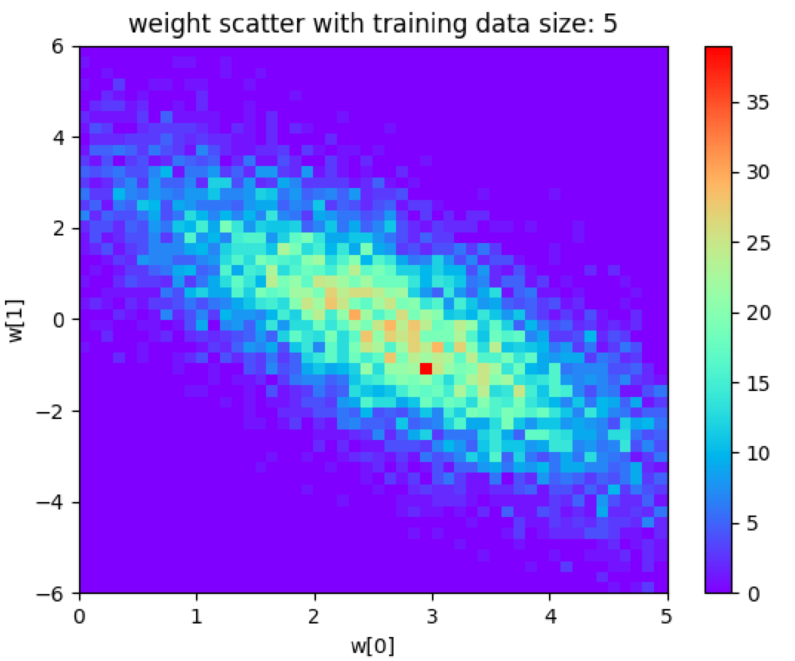

training data size: 80
training data
sample curve 1
sample curve 2
sample curve 3
sample curve 4
sample curve 5
t
x

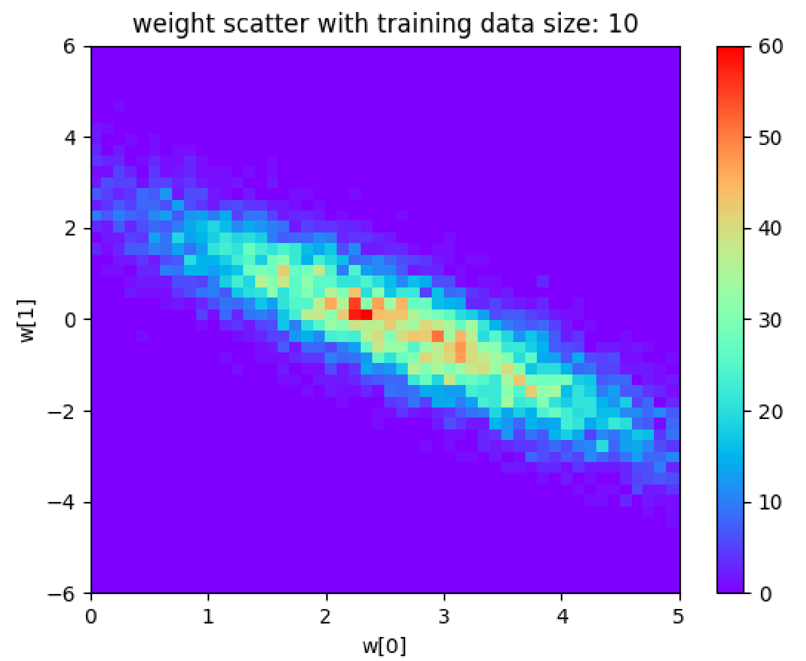
On page ten, I generate 5 sample curves to fit the training data, as we can see, the more dataset in training, the more probability to fit the data. Besides, we can observe one more special phenomenon is that if there has a training data in the graph, the sample curve will get together more often. If there is no training point at there, the predict sample curve will not become such dense.

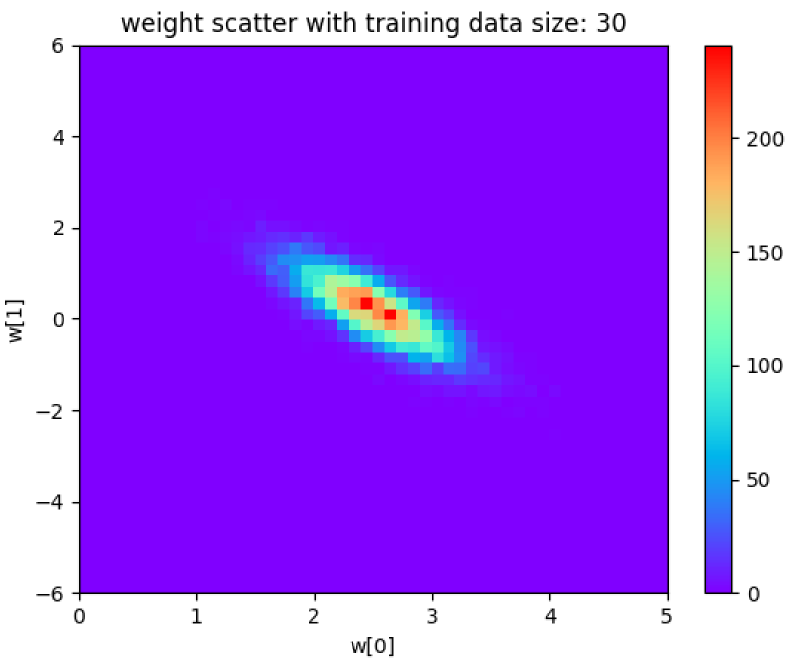


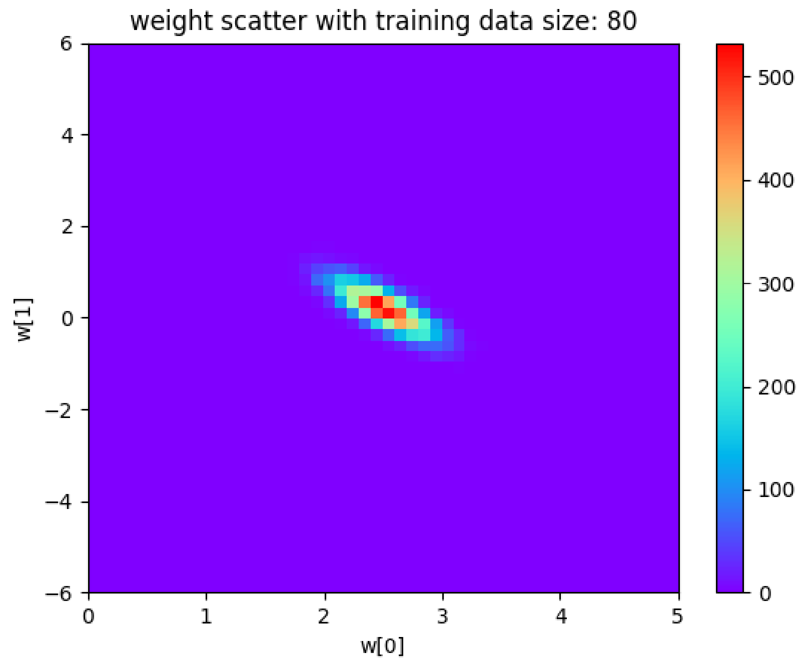
training data size: 5
training data
mean curve
t
x
training data size: 10
training data
mean curve
t
x
training data size: 30
training data
mean curve
t
x
training data size: 80
training data
mean curve
t
x

On page 12, we can see that the predict uncertainty depends on x and is smallest in the neighborhood of the data points. Also note that the level of the uncertainty decreases as more data points are observed.


weight scatter with training data size: 5
w[0]
w[1]
0 1 2 3 4 5
6 4 2 0 −2 −4 −6
0 5 10 15 20 25 30 35


weight scatter with training data size: 10
w[0]
w[1]
0 1 2 3 4 5
6 4 2 0 −2 −4 −6
0 10 20 30 40 50 60


weight scatter with training data size: 30
w[0]
w[1]
0 1 2 3 4 5
6 4 2 0 −2 −4 −6
0 50 100 150 200


weight scatter with training data size: 80
w[0]
w[1]
0 1 2 3 4 5
6 4 2 0 −2 −4 −6
0 100 200 300 400 500

On page 14, we can see that the posterior distribution will become a delta function centered on the true parameter values, and as the training data increase, the posterior distribution decrease.

Conclusion

From the result in the last few pages, we can give a conclusion that the level of uncertainty decreases as more data points are used to training and the distribution of the weight will become more dense. Furthermore, the data augmentation is also important for the machine learning. Once if we have enough data, we can train our model sequentially, so that it might be more efficient and accurate.