Name: Zihao Zhang
UNI: zz2763

Homework 2

1. **Solution to problem 1**

   (a) According to maximum likelihood estimation,

$$\begin{aligned}\hat{\pi} &= arg\max_{\pi} \sum_{i=1}^{n} \ln p(y_i|\pi) \\ &= arg\max_{\pi} \sum_{i=1}^{n} y_i \ln \pi + (1-y_i)\ln(1-\pi)\end{aligned}$$

   Let the derivative equals 0, we can get:

$$\sum_{i=1}^{n} (\frac{y_i}{\pi} - \frac{1-y_i}{1-\pi}) = 0$$

   Hence,

$$\hat{\pi} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

   (b) According to maximum likelihood estimation,

$$\hat{\lambda}_{y,d} = arg\max_{\lambda_{y,d}} (\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{y_i,d}))$$

   Since y can be either 1 or 0, we use an indicator $\mathbb{1}$ to denote it:

$$\sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{y_i,d}) = \sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{y,d})\mathbb{1}(y_i = y)$$

   Ignore the prior, since $x_{i,d}|y_i \sim \text{Pois}(\lambda_{y_i,d}), d = 1, ..., D$

$$\hat{\lambda}_{y,d} = arg\max_{\lambda_{y,d}} \sum_{i=1}^{n} -\lambda_{y,d} - \ln x_{i,d}! + x_{i,d}\ln\lambda_{y,d}\mathbb{1}(y_i = y)$$

   Let the derivative equal 0, we have:

$$\hat{\lambda}_{y,d} = \frac{\sum_{i=1}^{n} x_{i,d}\mathbb{1}(y_i = y)}{\sum_{i=1}^{n} \mathbb{1}(y_i = y)}$$

2. **Solution to problem 2**

(a) The 2×2 predicted table is shown as follows.

Table 1: Naive Bayes classifier prediction outcome

		predict	
		$y' = 1$	$y' = 0$
actual	$y = 1$	1702	111
	$y = 0$	492	2295

And the predicted accuracy is 86.9%.

(b) The stem plot of 54 Poisson parameters for each class averaged across the 10 runs.

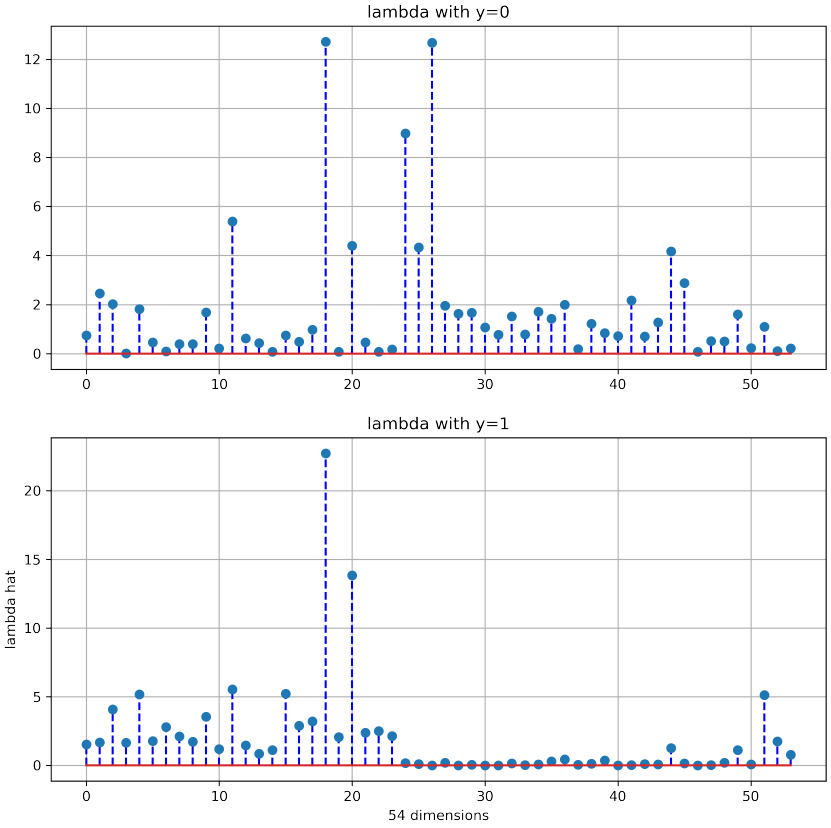

Figure 1: Stem plot of Poisson parameters

The dimension 16 is word 'free' and dimension 52 is an punctuation '!'. I think that 'free' might be emails inducing purchase since it is free and would appeal to people. And '!' means emotional expression which may also induce people. However, normal emails do not contain these usually.

(c) After running the steepest ascent logistics algorithm 10 times, the function trend is shown as follows.

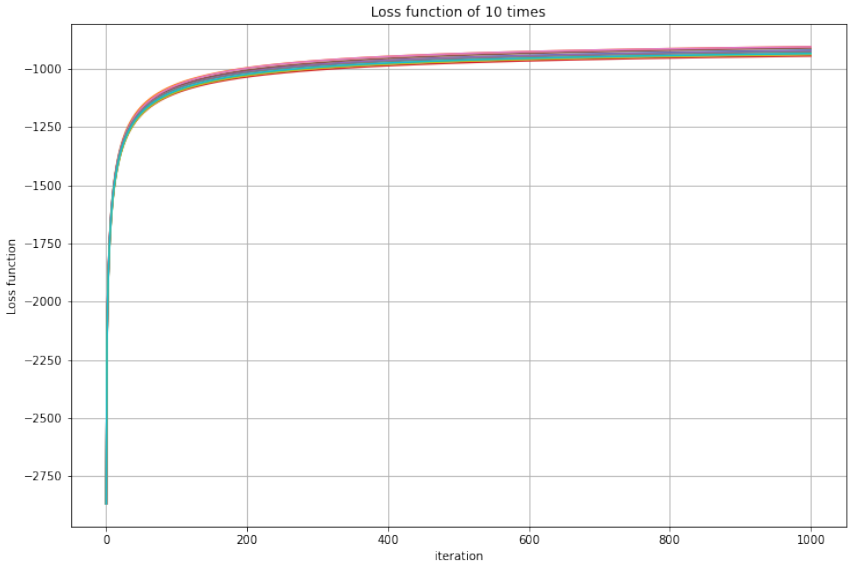

Figure 2: loss function

(d) Firstly we derive w_{t+1} according to the given function:

$$\mathcal{L}(w) \approx \mathcal{L}(w_t) + (w - w_t)^T \nabla \mathcal{L}(w_t) + \frac{1}{2}(w - w_t)^T \nabla^2 \mathcal{L}(w_t)(w - w_t)$$

Take gradient ∇_w at both side,

$$\begin{aligned} \nabla \mathcal{L}(w) &= \nabla \mathcal{L}(w_t) + \frac{1}{2}(w - w_t)^T \nabla^2 \mathcal{L}(w_t)^T + \frac{1}{2}\nabla^2 \mathcal{L}(w_t)(w - w_t) \\ &= \nabla \mathcal{L}(w_t) + \nabla^2 \mathcal{L}(w_t)(w - w_t) \end{aligned}$$

Setting the left as 0, we have:

$$0 = \nabla \mathcal{L}(w_t) + \nabla^2 \mathcal{L}(w_t)(w - w_t)$$

If $\nabla^2 \mathcal{L}(w_t)$ is non singular and replace w with w_{t+1}, we get:

$$w_{t+1} = w_t - (\nabla^2 \mathcal{L}(w_t))^{-1} \nabla \mathcal{L}(w_t)$$

After running the algorithm 10 times, the loss function is plotted as below.

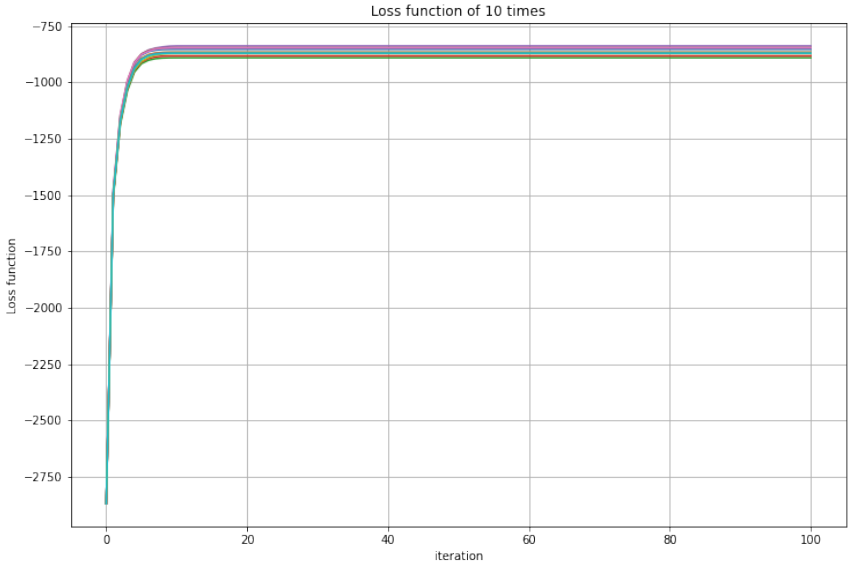

Figure 3: loss function

(e) The 2×2 predicted table is shown as follows. 1 means is spam and -1 means is not spam.

Table 2: Newton's method prediction outcome

	predict		
actual		$y' = 1$	$y' = -1$
	$y = 1$	1604	209
	$y = -1$	143	2644

And the predicted accuracy is 92.3%.

3. **Solution to problem 3**

(a) After running Gaussian process with 60 pairs of hyperparameter. The RMSE table is shown as below.

Table 3: RMSE table

RMSE		σ^2									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
b	5	1.966	1.933	1.923	1.922	1.925	1.929	1.935	1.941	1.947	1.953
	7	1.920	1.905	1.908	1.916	1.925	1.934	1.942	1.950	1.958	1.965
	9	1.898	1.903	1.918	1.933	1.946	1.957	1.967	1.976	1.985	1.992
	11	1.891	1.915	1.939	1.958	1.973	1.986	1.996	2.006	2.014	2.021
	13	1.896	1.936	1.965	1.986	2.001	2.014	2.024	2.033	2.041	2.049
	15	1.910	1.960	1.991	2.012	2.027	2.039	2.049	2.058	2.066	2.073

(b) The best parameter is $b = 11$ and $\sigma^2 = 0.1$. The RMSE is 1.891. In homework 1, the smallest RMSE appears to be approximately 2.08 when applying 3rd polynomial regression with $\lambda = 50$.

One of the drawback of Gaussian process is the intuition about the value of b and λ. Although the problem has given an approximate range for selection but the program still run 60 times. In every calculation, the program need solve the inversion of a huge matrix and will cost a lot of time. However, in homework 1, ridge regression only requires one hyperparameter λ and it can be 1 to 1000 in order. And the computation of ridge regression does not cost much time.

(c) After running the algorithms with $x[4]$ only (both training X and testing X), the comparison graph is shown as below.

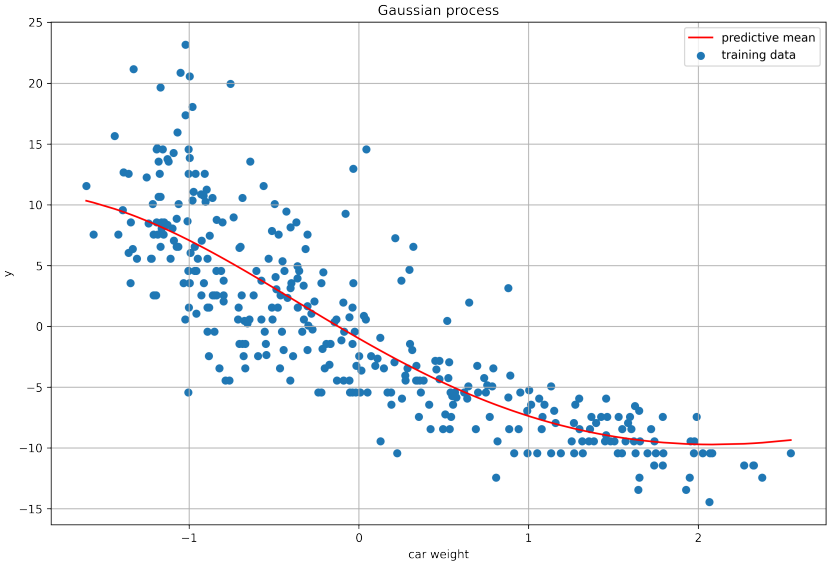

Figure 4: Gaussian process with $x[4]$