NAME - PRASHAM D. SHETH

UNI - pds2136

Problem 1 :—

→ We have labeled set of data

$$(y_1, x_1), \ldots, (y_n, x_n)$$

$y \in \{0, 1\}$

x is D-dimensional vector

We predict y_0 for new x_0 as

$$y_0 = \arg\max_y p(y_0 = y \mid \pi) \prod_{d=1}^{D} p(x_{0,d} \mid \lambda_{y,d})$$

$$p(y_0 = y \mid \pi) = \text{Bernoulli}(y \mid \pi)$$

$\therefore$ Data: $y_i \overset{iid}{\sim} \text{Bern}(\pi)$

$$x_{i,d} \mid y_i \sim \text{Pois}(\lambda_{y_i,d}) \qquad d = 1 \ldots D$$

Prior : $\lambda_{y,d} \overset{iid}{\sim} \text{Gamma}(2, 1)$

$$\hat{\pi}, \hat{\lambda}_{0,1:D}, \hat{\lambda}_{1,1:D} = \arg\max_{\pi, \lambda_{0,1:D}, \lambda_{1,1:D}} [L]$$

$$L = \left[\sum_{i=1}^{n} \ln p(y_i \mid \pi) + \sum_{d=1}^{D} (\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \sum_{i=1}^{n} \ln p(x_{i,d} \mid \lambda_{y_i,d}) \right.$$

(a) for $\hat{\pi}$, By using $p(y_i|\pi) = \pi^{y_i}(1-\pi)^{1-y_i}$ using first order characteristic,

$$\frac{\partial L}{\partial \hat{\pi}} = \frac{\partial}{\partial \hat{\pi}}\left[\sum_{i=1}^{n}(y_i \log \pi + (1-y_i)\log(1-\pi)) + c\right]$$

(here c represents the 2nd term in the objective which is a constant w.r.t π)

$$\therefore \frac{\partial L}{\partial \hat{\pi}} = \frac{\sum_{i=1}^{n} y_i}{\pi} + (-1)\frac{\sum_{i=1}^{n}(1-y_i)}{1-\pi} = 0$$

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

$$\therefore \frac{\sum_{i=1}^{N} y_i}{\pi} - \frac{n}{1-\pi} + \frac{\sum_{i=1}^{N} y_i}{1-\pi} = 0$$

$$\therefore \sum_{i=1}^{N} y_i \left[\frac{1}{\pi} + \frac{1}{1-\pi}\right] = \frac{n}{1-\pi}$$

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

$$\therefore \boxed{\hat{\pi} = \sum_{i=1}^{N} y_i / N}$$

(b) $\lambda_{y,d} \sim \text{Gamma}(2,1)$

$$p(\lambda_{y,d}) = \frac{(1)^2}{\Gamma(2)} \cdot (\lambda_{y,d})^1 \cdot e^{-\lambda_{y,d}}$$

as $\Gamma(2) = 1$

$$p(\lambda_{y,d}) = (\lambda_{y,d}) \cdot e^{-\lambda_{y,d}}$$

$x_{i,d} | y_i \sim \text{Pois}(\lambda_{y_i,d})$

$$p(x_{i,d} | y_i) = \frac{(\lambda_{y_i,d})^{x_{i,d}} \cdot e^{-\lambda_{y_i,d}}}{(x_{i,d})!}$$

We can write L as

$$L = c' + \sum_{d=1}^{D} \left(\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \sum_{i=1}^{n} \ln p(x_{i,d} | \lambda_{y_i,d}) \right)$$

c' is first term of L which is a constant w.r.t $\lambda_{0,1:D}$ & $\lambda_{1,1:D}$

Also, $\sum_{i=1}^{N} \ln p(x_{i,d} | \lambda_{y_i,d}) = \sum_{i=1}^{N} \left(y_i \cdot \ln p(x_{i,d} | \lambda_{1,d}) + (1-y_i) \cdot \ln p(x_{i,d} | \lambda_{0,d}) \right)$

(here $y_i = 1$ can be used as indicator for $\lambda_{1,d}$

and $(1-y_i) = 1 \Rightarrow y_i = 0$ can be used as indicator for $\lambda_{0,d}$.

$$\therefore L = c' + \sum_{d=1}^{D} \Big(\ln p(\lambda_{0,d}) + \ln p(\lambda_{1,d}) + \sum_{i=1}^{n} \big(y_i \cdot \ln p(x_{i,d} | \lambda_{1,d}) + (1-y_i) \cdot \ln p(x_{i,d} | \lambda_{0,d}) \big) \Big)$$

differentiating L w.r.t $\lambda_{0,d}$, (leaving d arbitrary)

$$\frac{\partial L}{\partial \lambda_{0,d}} = \frac{\partial}{\partial \lambda_{0,d}} \Big[c' + \sum_{d=1}^{D} \ln p(\lambda_{0,d}) + \ln(\lambda_{1,d}) + \sum_{i=1}^{n} \big(y_i \cdot \ln p(x_{id} | \lambda_{1,d}) + (1-y_i) \ln p(x_{i,d} | \lambda_{0,d}) \big) \Big]$$

$$\ln p(\lambda_{0,d}) = \log(\lambda_{0,d}) - \lambda_{0,d}$$

$$\ln p(\lambda_{1,d}) = \log(\lambda_{1,d}) - \lambda_{1,d}$$

$$\ln p(x_{i,d} | \lambda_{1,d}) = x_{i,d} \log(\lambda_{1,d}) - \lambda_{1,d} - \log(x_{i,d})!$$

$$\ln p(x_{i,d} | \lambda_{0,d}) = x_{i,d} \log(\lambda_{0,d}) - \lambda_{0,d} - \log(x_{i,d})!$$

$$\therefore \frac{\partial L}{\partial \lambda_{0,d}} = 0$$

$$\therefore 0 = -1 + \frac{1}{\lambda_{0,d}} + \sum_{i=1}^{n} (1-y_i) \cdot \left[\frac{x_{i,d}}{\lambda_{0,d}} - 1 \right]$$

$$\therefore \lambda_{0,d} = \frac{\sum_{i=1}^{n} (1-y_i) \cdot x_{i,d} + 1}{\sum_{i=1}^{n} (1-y_i) + 1}$$

Similarly, for $\hat{\theta}_{1,d}$

$$\partial L / \partial \hat{\theta}_{1,d} = -1 + \frac{1}{\theta_{1,d}} + \sum_{i=1}^{n} y_i \left[\frac{x_{i,d}}{\theta_{1,d}} - 1 \right]$$

$$\therefore \quad \hat{\theta}_{1,d} = \frac{\sum_{i=1}^{n} y_i \cdot x_{i,d} + 1}{\sum_{i=1}^{n} y_i + 1}.$$

$\therefore$ from the above forms of $\hat{\theta}_{1,d}$ and $\hat{\theta}_{0,d}$

$$\hat{\theta}_{y,d} = y \left[\frac{\sum_{i=1}^{N} y_i \cdot x_{i,d} + 1}{\sum_{i=1}^{N} y_i + 1} \right] + (1-y) \left[\frac{\sum_{i=1}^{N} (1-y_i) x_{i,d} + 1}{\sum_{i=1}^{N} (1-y_i) + 1} \right]$$

Problem – 2:

(a) Implement the naive Bayes classifier described above. In a 2 * 2 table, write the number of times that you predicted a class y data point (ground truth) as a class y0 data point (model prediction) in the (y; y0)th cell of the table, where y and y0 can be either 0 or 1. There should be four values written in the table in your PDF. Next to your table, write the prediction accuracy—the sum of the diagonal divided by 4600. (The sum of all entries in the table should be 4600.)

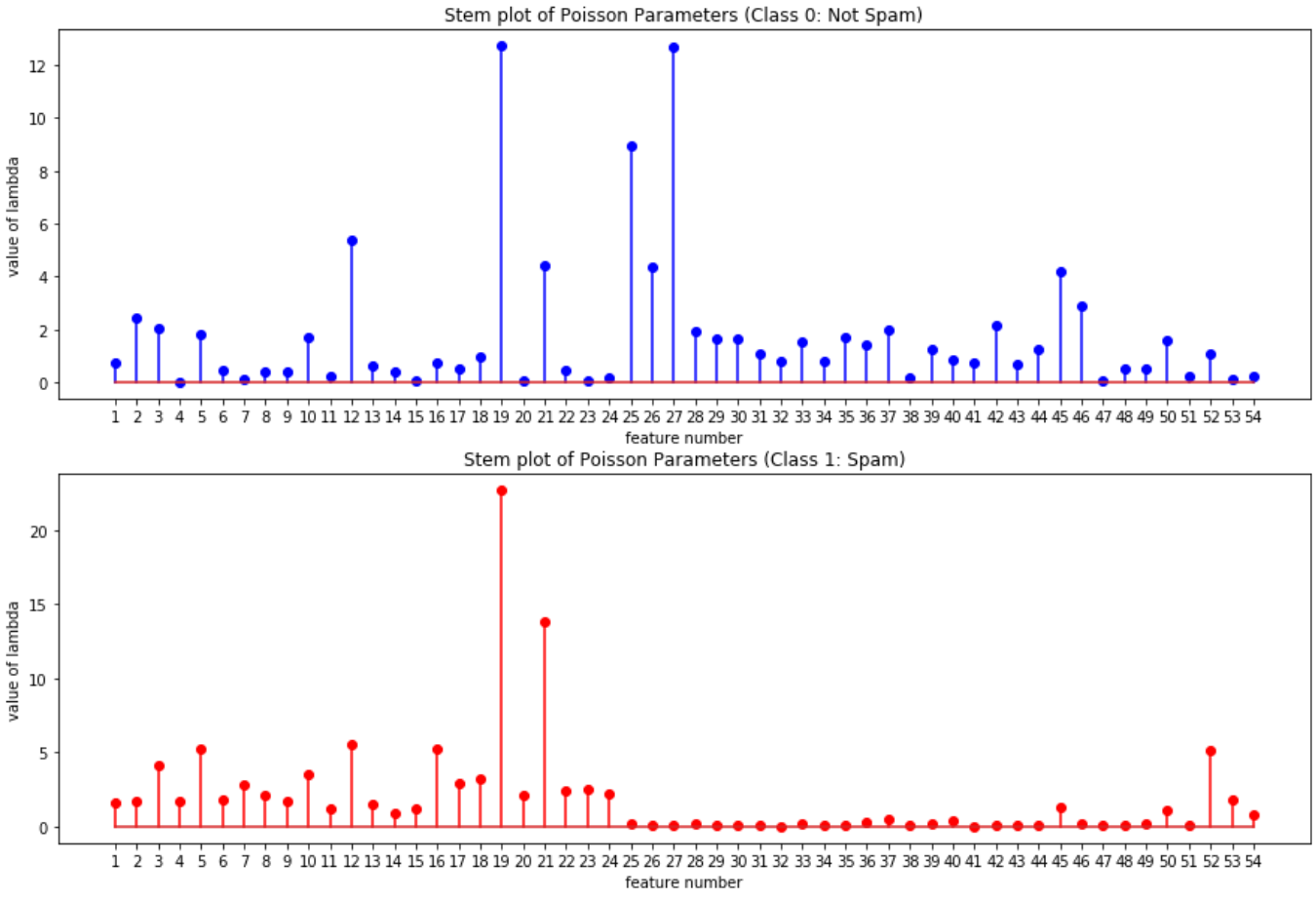
- For the Naïve Bayes implementation as done in the code, we get the following confusion matrix:

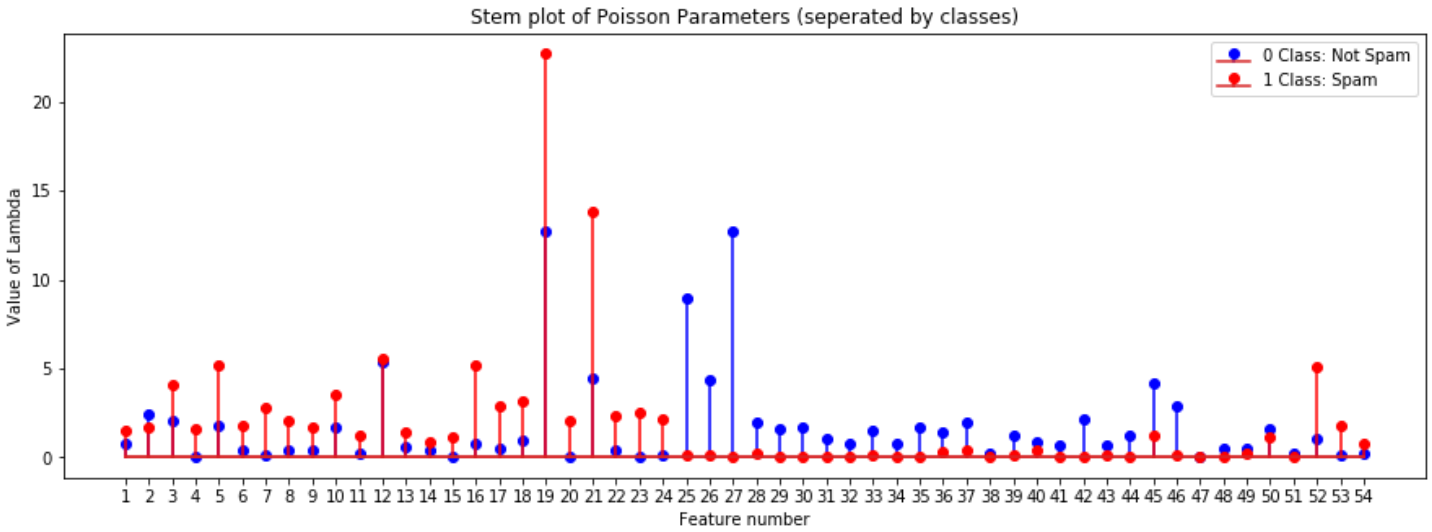
	Predicted y = 1	Predicted y = 0
Actual y = 1	1705	491
Actual y = 0	108	2296

- The predicted accuracy for the same is (1705 + 2296) / 4600 = 0.86978 ≈ 0.87

(b) In one figure, show a stem plot (stem () in MATLAB) of the 54 Poisson parameters for each class averaged across the 10 runs. (This average is only used for plotting purposes on this homework. In practice you would relearn these parameters using the entire data set to find their final values.) Use the README file to make an observation about dimensions 16 and 52.?

- The stem plot for the 54 Poisson parameters for each class averaged across the 10 runs is shown in the following figures. The first figure shows the stem plot for both the classes on two different axes while in the second one, I have plotted those over each other on the same axis to get a better view at comparing the values corresponding to the Poisson parameter of 16th and 52nd department.

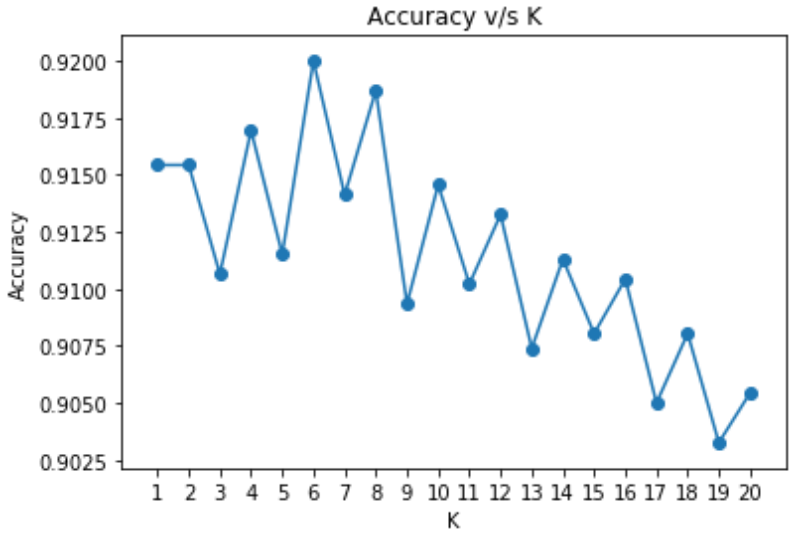




- Dimension 16 represents the word “free” in the email whereas dimension 52 represents the symbol “!” in the email.
- From the above plot (the second one in which both are plotted on the same axis) we can say that the value of lambda for both of the values is higher when the class is Spam (i.e.) 1. This tells us that they both are occurring more frequently in spam mails as compare to the non-spam ones.
- This thing aligns with an intuitive sense that most of the spam emails are for the marketing of some product/service in which both of these words very frequently.
- Moreover, the values have an almost similar scale in both of the classes, which further suggests that both of them: “free” and “!” have occurred a similar amount of times in the given data (across all the examples belonging to each class). This might be due to the fact that many times “free!” (i.e.) both the words are commonly used together in many of the spam mails especially the marketing ones.

(c) Implement the k-NN classifier for k = 1, …, 20. Use the L1 distance for this problem. Plot the prediction accuracy as a function of k.

- The following figure shows the prediction accuracy as a function of k. For the implementation of KNN, we broke the tie between the classes when k is even by giving the label of the nearest neighbour of the point.



Problem – 3:

(a) ***Write code to implement the Gaussian process and to make predictions on test data.***

- The code is written in the .py file submitted with this file. The Gaussian process class represents the code for implementing Gaussian Process. Kernel method of the class gives the value of Kernel Function by using the value of b passed as a parameter to the class.

(b) ***For b in (5; 7; 9; 11; 13; 15) and sigma^2 in (.1; .2; .3; .4; .5; .6; .7; .8; .9; 1)—so 60 total pairs (b; sigma^2)—calculate the RMSE on the 42 test points as you did in the first homework. Use the mean of the Gaussian process at the test point as your prediction. Show your results in a table.***

- The following table shows the values of RMSE for the corresponding pair of b and sigma^2. "b" values are mentioned in the rows and sigma^2 in the columns.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
5	1.966277	1.933137	1.923422	1.922199	1.924771	1.929215	1.934636	1.940585	1.946822	1.953215
7	1.920164	1.904878	1.908082	1.915904	1.924806	1.933704	1.942256	1.950382	1.958095	1.965440
9	1.897650	1.902521	1.917650	1.932517	1.945702	1.957237	1.967406	1.976494	1.984743	1.992344
11	1.890509	1.914983	1.938851	1.957939	1.973218	1.985767	1.996378	2.005606	2.013838	2.021347
13	1.895850	1.935588	1.964600	1.985504	2.001317	2.013881	2.024313	2.033309	2.041320	2.048644
15	1.909605	1.959551	1.990806	2.011918	2.027373	2.039467	2.049466	2.058107	2.065847	2.072978

(c) ***Which value was the best and how does this compare with the first homework? What might be a drawback of the approach in this homework (as given) compared with homework 1?***

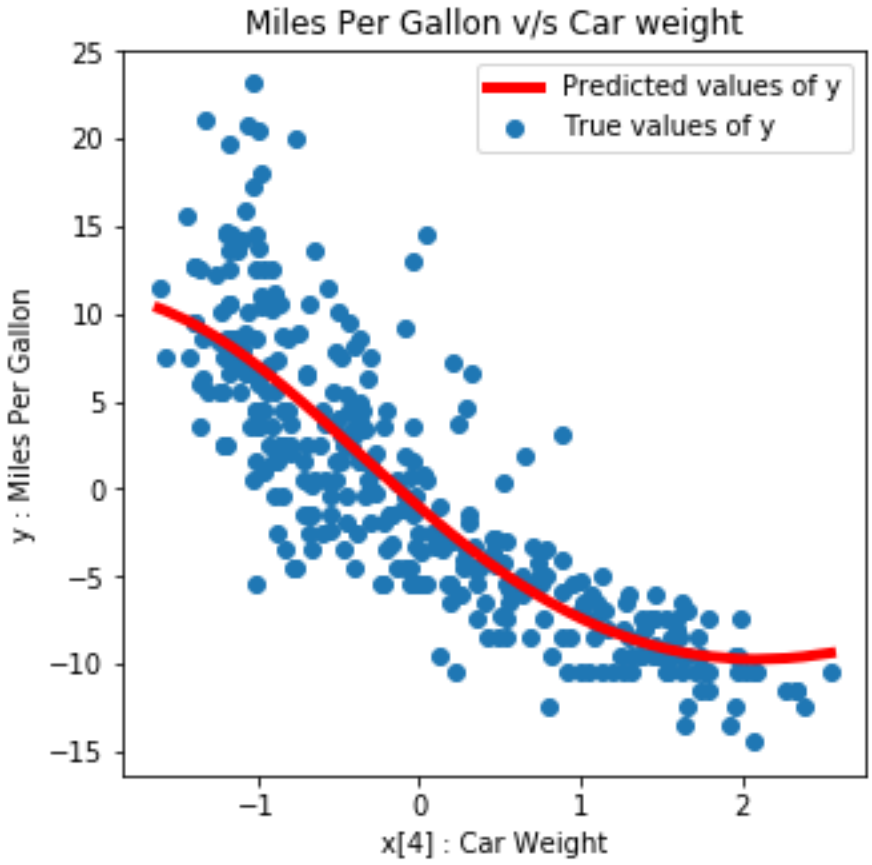
- From the above table, we can see that the RMSE value is the lowest for the following values of b and sigma^2.

b = 11 and sigma^2 = 0.1

- As can be seen here the lowest RMSE value is 1.89 whereas in the first homework it was above 2. This shows that in the particular case the Gaussian Process performs better than the p^{th} order regression as done in the first homework.
- The probable drawback of the approach in this homework as compared to the first one is that here, we are required to compute the Kernel Matrix as well as its inverse and perform the multiplication of various matrices then. This would result in more time consumption and along with an increase in the number of datapoints the time difference between Gaussian Process and Linear Regression would keep on increasing.

(d) ***To better understand what the Gaussian process is doing through visualization, re-run the algorithm by using only the 4th dimension of xi (car weight). Set b = 5 and sigma^2 = 2. Show a scatter plot of the data (x [4] versus y for each point). Also, the plot as a solid line of the predictive mean of the Gaussian process at each point in the training set. You can think of this problem as asking you to create a test set by duplicating xi [4] for each i in the training set and then to predict that test set.***

- The following figure shows the actual distribution of the data as a scatter plot. The orange line shows the predictive mean for the Gaussian process at each point in the training set.


Miles Per Gallon v/s Car weight
Predicted values of y
True values of y
y : Miles Per Gallon
x[4] : Car Weight
25
20
15
10
5
0
−5
−10
−15
−1
0
1
2