Assignment 1

Matish Singh Tanwar AI20MTECH11005
Adarsh Srivastava AI20MTECH14008

October 26, 2020

1 Question 1

Consider a linear model of the form

$$y(x, w) = w_0 + \sum w_i x_i \tag{1.0.1}$$

together with a sum-squares error function of the form

$$E_D(w) = \frac{1}{2} \sum \{y(x_n, w) - t_n\}^2 \tag{1.0.2}$$

Now suppose that Gaussian noise ϵ_i with zero mean and variance σ^2 is added independently to each of the input variables x_i. By making use of $\mathrm{E}[\epsilon_i] = 0$ and $\mathrm{E}[\epsilon_i \epsilon_j] = \delta_i j \sigma^2$, show that minimizing E_D averaged over the noise distribution is equivalent to minimizing the sum-of-squares error for noise-free input variables with the addition of a weight-decay regularization term, in which the bias parameter w_0 is omitted from the regularizer.

1.1 Solution

According to the question, noise is added to each input variable individually, So,

$$y'(x, w) = w_0 + \sum_{i=1}^{D} (w_i x_i + \epsilon_i) \tag{1.1.1}$$

becomes our new model where ϵ_i refers to the individual errors. Now, using (1.1.1)

$$y'(x, w) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{1}^{D} \epsilon_i \tag{1.1.2}$$

$$y'(x, w) = y(x, w) + \sum_{i=1}^{D} \epsilon_i \tag{1.1.3}$$

Now, let's consider the effect of noise on the sum square error function,

$$E'_D(w) = \frac{1}{2}\sum\{y'(x_n, w) - t_n\}^2 \tag{1.1.4}$$

Substituting (1.1.3) in (1.1.4),

$$E'_D(w) = \frac{1}{2}\sum_{1}^{N}\{y(x, w) + \sum_{i=1}^{D} w_i \in_i - t_n\}^2 \tag{1.1.5}$$

$$= \frac{1}{2}\sum_{1}^{N}\{(y(x, w))^2 + (\sum_{i=1}^{D} w_i \in_i)^2 + t_n^2 + 2y(x, w)\sum_{i=1}^{D} w_i \in_i - 2\sum_{i=1}^{D} w_i \in_i t_n - 2t_n y(x, w)\} \tag{1.1.6}$$

Now, taking expectation of the new sum-of-squares

$$E[E'_D(w)] = \frac{1}{2}\sum_{1}^{N}\{(y(x, w))^2 + E[(\sum_{i=1}^{D} w_i \in_i)^2] + t_n^2 + 2y(x, w)\sum_{i=1}^{D} w_i E[\in_i] - 2\sum_{i=1}^{D} w_i E[\in_i] t_n - 2t_n y(x, w)\} \tag{1.1.7}$$

Since E[$\in_i$]=0,

$$E[E'_D(w)] = \frac{1}{2}\sum_{1}^{N}\{(y(x, w))^2 + E[(\sum_{i=1}^{D} \in_i)^2] + t_n^2 - 2t_n y(x, w)\} \tag{1.1.8}$$

$$= \frac{1}{2}\sum_{1}^{N}\{(y(x, w))^2 + \sum_{i=1}^{D}\sum_{i=1}^{D} w_i w'_i \in_i \in'_i + t_n^2 - 2t_n y(x, w)\} \tag{1.1.9}$$

$$= \frac{1}{2}\sum_{1}^{N}\{(y(x, w))^2 + \sum_{i=1}^{D}\sum_{i=1}^{D} w_i w'_i E[\in_i \in'_i] + t_n^2 - 2t_n y(x, w)\} \tag{1.1.10}$$

Since $E[i_i i_i] = 1$,

$$= \frac{1}{2}\sum_{1}^{N}\{(y(x, w))^2 + \sum_{i=1}^{D} w_i^2 + t_n^2 - 2t_n y(x, w)\} \tag{1.1.11}$$

$$= \frac{1}{2}\sum_{1}^{N}\{(y(x, w) - t_n)^2 + \sum_{i=1}^{D} w_i^2\} \tag{1.1.12}$$

$$= E_D(w) + \frac{N}{2}\sum_{i=1}^{D} w_i^2 \tag{1.1.13}$$

Hence we get a Ridge Regression term without the bias parameter.

2 Question 2

Consider the problem where inputs are associated with multiple real valued outputs (K ¿ 1) known as multi output regression (For e.g. predicting student score across different courses).

$$y(x, w) = W^T(x)\phi(x)$$

here y is a K-dimensional column vector, W is an M * K matrix of parameters,and ϕ(x) is an M-dimensional column vector with elements ϕj(x), with ϕ0(x) = 1.

1. Provide the expression for the likelihood, and derive ML and MAP estimates of W in the multi output regression case.

2. Consider a multi-output regression problem where we have multiple independent outputs in linear regression. Let's consider a 2 dimensional output vector $y_i \in R^2$. Suppose we have some binary input data, $x_i \in 0, 1$. The training data is as given in the right side. Let us embed each x_i into 2d using the following basis function: ϕ (0) = $(1, 0)^T$, $\phi(1) = (0, 1)^T$. The model becomes y =$W^T(x)$ where W = [w_1, w_2] is a 2 x 2 matrix, with both w_1 and w_2 column vectors. Find the MLE for w_1 and w_2

x	y
0	(-1,-1)^T
0	(-1,-2)^T
0	(-2,-1)^T
1	(1,1)^T
1	(1,2)^T
1	(2,1)^T

2.1 Solution

2.2 Estimation of MLE

$$t = y(x, w) + \epsilon \tag{2.2.1}$$

where ϵ is a zero mean gaussian random variable with variance σ^2

$$\epsilon \sim N(0, \sigma^2) \tag{2.2.2}$$

And we know that t labels are normally distributed.Our aim is to estimate the best values for mean and variance of normal distribution t. Let's get the mean

and variance of t in terms of y and ϵ normal distribution.

$$\mathbf{E}(t) = \mathbf{E}(y + \epsilon) \tag{2.2.3}$$
$$\mathbf{E}(t) = \mathbf{E}(y) + \mathbf{E}(\epsilon) \tag{2.2.4}$$
$$\mathbf{E}(t) = W^T\phi(x) + 0 \tag{2.2.5}$$
$$\mathbf{E}(t) = W^T\phi(x) \tag{2.2.6}$$

Similarly for variance,

$$\mathbf{Var}(t) = \mathbf{Var}(y + \epsilon) \tag{2.2.7}$$
$$\mathbf{Var}(t) = \mathbf{Var}(y) + \mathbf{Var}(\epsilon) \tag{2.2.8}$$
$$\mathbf{Var}(t) = 0 + \sigma^2 \tag{2.2.9}$$
$$\mathbf{Var}(t) = \sigma^2 \tag{2.2.10}$$

So,

$$t \sim \mathcal{N}(W^T\phi(x), \sigma^2) \tag{2.2.11}$$
$$p(t|x, w, \sigma^2) = \mathcal{N}(t|W^T\phi(x), \sigma^2) \tag{2.2.12}$$

If we have a set of observations $t_1, ..., t_n$ we can combine these into a matrix T of size NxK such that n^{th} row is given by t_n^T.Similarly we can combine the input vectors $X1, ...X_n$ into a matrix X. The log likelihood function is then given by:-

$$\ln(p(t|x, w, \sigma^2)) = \sum_{n=1}^{N} \ln(\mathcal{N}(t_n|W^T\phi(x_n), \sigma^2)) \tag{2.2.13}$$
$$\implies -\frac{NK}{2}\ln(2\pi) - \frac{NK}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left|t_n - W^T\phi(x_n)\right|^2 \tag{2.2.14}$$

Differentiating above equation with respect to w

$$\frac{\partial(\ln(p(t|x, w, \sigma^2)))}{\partial w} = \frac{1}{2\sigma^2}\frac{\partial(T^2 - 2\phi(x_n)^T WT + \phi(x_n)^T\phi(x_n)W^2)}{\partial w} \tag{2.2.15}$$
$$\implies \frac{1}{2\sigma^2}(0 - 2\phi(x_n)^T + 2\phi(x_n)^T\phi(x_n)W) \tag{2.2.16}$$

Optimal values for W is when

$$\frac{\partial(\ln(p(t|x, w, \sigma^2)))}{\partial w} = 0 \tag{2.2.17}$$
$$\implies 2\phi(x_n)^T T = 2\phi(x_n)^T\phi(x_n)W \tag{2.2.18}$$
$$\implies \boxed{W = (\phi(x_n)^T\phi(x_n))^{-1}\phi(x_n)^T T} \tag{2.2.19}$$

2.3 Estimation of MAP

Conjugate pair of Normal Distribution is Normal Distribution. p(w|x,t,α,β) is the posterior.We have to calculate this.In this Prior probability will also be taken into consideration. where β is inverse variance used for Likelihood Normal Distribution.

$$p(w|x,t,\alpha,\beta) \propto p(t|x,w,\beta)p(w|\alpha) \tag{2.3.1}$$

$$p(w|\alpha) = \mathcal{N}(w|0,\alpha^{-1}I) \tag{2.3.2}$$

$$\implies (\frac{\alpha}{2\pi})^{(M+1)/2} \exp(-\frac{\alpha}{2}w^T w) \tag{2.3.3}$$

$$\tag{2.3.4}$$

Taking log on both sides

$$\ln(p(w|\alpha)) = \frac{M+1}{2}\ln\alpha - \frac{M+1}{2}\ln 2\pi - \frac{\alpha}{2}w^T w \tag{2.3.5}$$

We got both prior as well as likelihood.Now combining them and taking the log of posterior also we get equation as

$$-\frac{\beta}{2}\sum_{n=1}^{N}\left|t_n - W^T\phi(x_n)\right|^2 - \frac{\alpha}{2}w^T w \tag{2.3.6}$$

Now differentiating this with respect to w we get

$$-\frac{\beta}{2}(0 - 2\phi(x_n)^T T + 2\phi(x_n)^T\phi(x_n)W) - \alpha W \tag{2.3.7}$$

Making it equal to 0 for optimal values,we get

$$\phi(x_n)^T T = (\phi(x_n)^T\phi(x_n) + \frac{\alpha}{\beta})W \tag{2.3.8}$$

$$\implies \boxed{W = (\frac{\alpha}{\beta}I + \phi(x_n)^T\phi(x_n))^{-1}\phi(x_n)^T T} \tag{2.3.9}$$

For each target variable t_k we get w_k as

$$w_k = (\frac{\alpha}{\beta}I + \phi(x_n)^T\phi(x_n))^{-1}\phi(x_n)^T t_k \tag{2.3.10}$$

2.4 As per the given question we can write,

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} — & w_1^T & — \\ — & w_2^T & — \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \tag{2.4.1}$$

Now, From the table provided in the question,

$$X = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, y_1 = \begin{bmatrix} -1 \\ -1 \\ -2 \\ 1 \\ 1 \\ 2 \end{bmatrix}, y_2 = \begin{bmatrix} -1 \\ -2 \\ -1 \\ 1 \\ 2 \\ 1 \end{bmatrix} \tag{2.4.2}$$

From equation one it is evident that multi-output problem is composed of following two single output linear regression.

$$y_1 = \phi(x)\hat{w}_1 \tag{2.4.3}$$

$$\hat{W}_1 = ((X^T X)^{-1} X^T) y_1 \tag{2.4.4}$$

$$y_2 = \phi(x)\hat{w}_2 \tag{2.4.5}$$

$$\hat{W}_2 = ((X^T X)^{-1} X^T) y_2 \tag{2.4.6}$$

Calculating the value of $(X^T X)^{-1} X^T$,

$$(X^T X)^{-1} X^T = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \tag{2.4.7}$$

Solving (2.4.4) and (2.4.6),

$$\hat{w}_1 = ((X^T X)^{-1} X^T) y_1 \tag{2.4.8}$$

$$= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -2 \\ 1 \\ 1 \\ 2 \end{bmatrix} \tag{2.4.9}$$

$$= \begin{bmatrix} -\frac{4}{3} \\ \frac{4}{3} \end{bmatrix} \tag{2.4.10}$$

$$\hat{w}_2 = ((X^T X)^{-1} X^T) y_2 \tag{2.4.11}$$

$$= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} -1 \\ -2 \\ -1 \\ 1 \\ 2 \\ 1 \end{bmatrix} \tag{2.4.12}$$

$$= \begin{bmatrix} -\frac{4}{3} \\ \frac{4}{3} \end{bmatrix} \tag{2.4.13}$$

As we can see that $\hat{w}_1$ and $\hat{w}_2$ are equal to $\begin{bmatrix} -\frac{4}{3} \\ \frac{4}{3} \end{bmatrix}$ Hence, $\hat{W}$=$\begin{bmatrix} -\frac{4}{3} & -\frac{4}{3} \\ \frac{4}{3} & \frac{4}{3} \end{bmatrix}$

3 Question 3

Model the horse kick deaths using the Poisson distribution with different parameters for each of the corps. Learn Poisson distribution parameters for each of the corps using first 13 years of data and make predictions on remaining 7 years and compute the RMSE of predictions for each of the corps.

2. (a) Use maximum likelihood estimation to learn the parameters.
   (b) Use maximum aposteriori estimation to learn the parameters
      i. Assume appropriate prior distribution over parameters and justify your assumption
      ii. Plot prior, likelihood and posterior and provide your observations in terms of mode of the distributions for corps 2, 4 and 6.

3.1 Solution

(a) In this part we have to estimate the parameters using MLE and predict the values for the next 7 years. So, first we are calculating MLE for each corps as shown in the following code.

Listing 1: Parameter Through MLE

```
import numpy
import matplotlib.pyplot as plt
import math
'Creating_training_and_test_datasets'
train=numpy.array
    ([[0,2,2,1,0,0,1,1,0,3,0,2,1],[0,0,0,1,0,3,0,2,0,0,0,1,1],[0,0,0,2,0,2,0,0,1,1,0,0,2],[0,0,0,1,1,1,2,0,2,0,0,0,1],

[0,1,0,1,1,1,1,0,0,0,0,1,0],[0,0,0,0,2,1,0,0,1,0,0,1,0],[0,0,1,0,2,0,0,1,2,0,1,1,3],[1,0,1,0,0,0,1,0,1,1,0,0,2],[1,0,0,0,1,0,0,1,0,0,0,0,1],

[0,0,0,0,0,2,1,1,1,0,2,1,1],[0,0,1,1,0,1,0,2,0,2,0,0,0],[0,0,0,0,0,2,4,0,1,3,0,1,1,1],[1,1,2,1,1,3,0,4,0,1,0,3,2],[0,1,0,0,0,0,0,0,1,0,1,1,0,0]])

test=numpy.array
    ([[0,0,1,0,1,0,1],[1,0,2,0,3,1,0],[1,1,0,0,2,0,0],[0,1,2,1,0,0,0],[0,0,0,1,1,0,0],[1,1,1,1,1,1,0],

        [1,1,1,0,3,0,0],[0,0,2,1,0,2,0],[0,0,0,1,1,0,1],[0,1,2,0,1,0,0],[0,2,1,3,0,1,1],[1,2,1,3,1,3,1],

        [1,0,2,1,1,0,0],[0,2,2,0,0,0,0]])
'Using_MLE_to_learn_the_parameters'
MLE_parameter=[sum(i) for i in train]
MLE_parameter=[i/13 for i in MLE_parameter]
print("MLE_Parameters_for_Corps:")
print(MLE_parameter)
plt.plot(MLE_parameter)
plt.title("MLE_Parameter")
plt.show()
```

In this portion we are predicting the values of next 7 years by using MLE of the previous 13 years. MLE is depecting the average number of deaths per year and that average death rounded off can be considered as the estimation for death per year.

Listing 2: Predicted Values

```
"Now predicting values for last 7 years'
prediction=[round(i) for i in MLE_parameter]
print("Predicted Value for next 7 years for each corp")
predictedval=[i*7 for i in prediction]
print(predictedval)
```

Finally, We are calculating the Root Mean Square Error by using the formula

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2} \tag{3.1.1}$$

Listing 3: Root Mean Square Error

```
'Calculating RMSE'
values=[]
for i in range(0,len(test)):
    value=[]
    for j in range(0,7):
        value.append(prediction[i])
    values.append(value)
diff=[]
for i in range(0,len(test)):
    diff.append(numpy.subtract(values[i],test[i]))
sum=[]
diff=numpy.array(diff)
data=[0,0,0,0,0,0,0,0,0,0,0,0,0,0]
for i in range(0,len(test)):
        data[i]=[i*i for i in diff[i]]
for i in range(0,len(test)):
    sum.append(numpy.sum(data[i]))
sum=numpy.array(sum)
MSE=[i/7 for i in sum]
RMSE=[numpy.sqrt(i) for i in MSE]
print("RMSE for Corps")
print(RMSE)
plt.plot(RMSE)
plt.title("Root Mean Square Error")
plt.show()
print("Average RMSE for Corps")
print(numpy.sum(RMSE)/len(RMSE))
```

i. Results MLE for the corps came out to be,

$$x_1 = 1.0 \tag{3.1.2}$$
$$x_2 = 0.6153846153846154 \tag{3.1.3}$$
$$x_3 = 0.6153846153846154 \tag{3.1.4}$$
$$x_4 = 0.6153846153846154 \tag{3.1.5}$$
$$x_5 = 0.46153846153846156 \tag{3.1.6}$$
$$x_6 = 0.38461538461538464 \tag{3.1.7}$$
$$x_7 = 0.8461538461538461 \tag{3.1.8}$$
$$x_8 = 0.5384615384615384 \tag{3.1.9}$$
$$x_9 = 0.3076923076923077 \tag{3.1.10}$$
$$x_{10} = 0.6923076923076923 \tag{3.1.11}$$
$$x_{11} = 0.5384615384615384 \tag{3.1.12}$$
$$x_{11} = 1.0 \tag{3.1.13}$$
$$x_{12} = 1.4615384615384615 \tag{3.1.14}$$
$$x_{13} = 0.3076923076923077 \tag{3.1.15}$$

Prediction per year is the rounded off value of these MLE. RMSE for the corps came out to be,

$$x_1 = 0.7559289460184544 \tag{3.1.16}$$
$$x_2 = 1.0690449676496976 \tag{3.1.17}$$
$$x_3 = 0.8451542547285166 \tag{3.1.18}$$
$$x_4 = 0.8451542547285166 \tag{3.1.19}$$
$$x_5 = 0.5345224838248488 \tag{3.1.20}$$
$$x_6 = 0.9258200997725514 \tag{3.1.21}$$
$$x_7 = 1.0 \tag{3.1.22}$$
$$x_8 = 0.9258200997725514 \tag{3.1.23}$$
$$x_9 = 0.6546536707079771 \tag{3.1.24}$$
$$x_{10} = 0.8451542547285166 \tag{3.1.25}$$
$$x_{11} = 1.0 \tag{3.1.26}$$
$$x_{11} = 1.1338934190276817 \tag{3.1.27}$$
$$x_{12} = 0.7559289460184544 \tag{3.1.28}$$
$$x_{13} = 1.0690449676496976 \tag{3.1.29}$$
$$Avg_{RMSE} = 0.8828657403305332 \tag{3.1.30}$$

ii. Plots
This portion consist of visual representation of MLE parameters and RMSE of each individual corps.

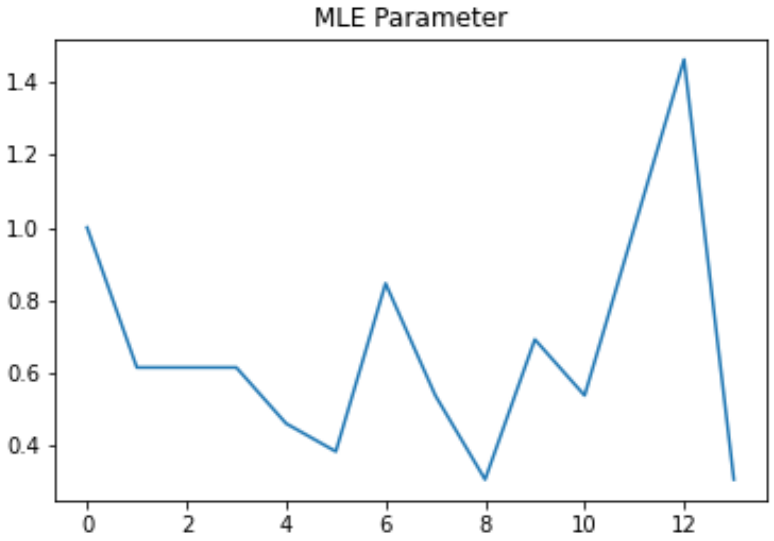

Figure 0

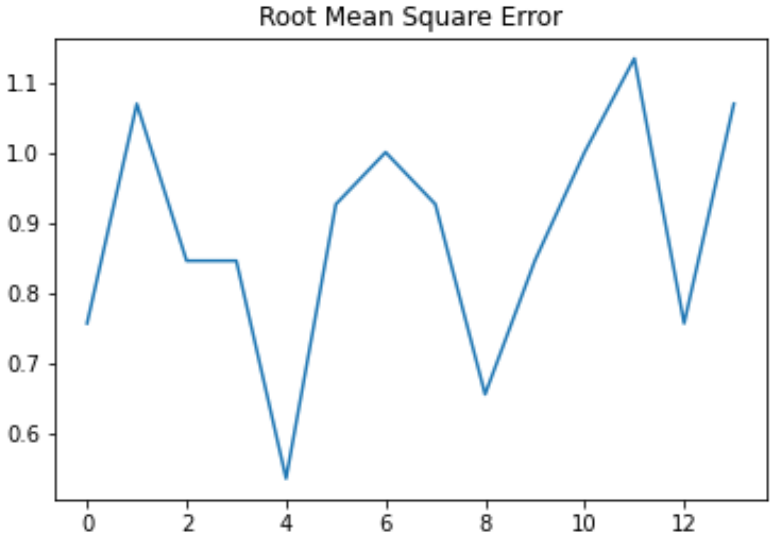

Figure 0

(b) In this section we are using MAP to estimate the parameters. We are taking Gamma as the prior in the following code.An explanation of why gamma is a suitable prior will be present at the end of the question.

Listing 4: Plotting Prior,Likelihood and Posterior

```
def mode(values):
    values.sort()
    temp=[]
    i = 0
    while i < len(values) :
        temp.append(values.count(values[i]))
        i += 1
    temp1 = dict(zip(values, temp))
    temp2={k for (k,v) in temp1.items() if v == max(temp) }
    for x in temp2:
        Mode=x
    return Mode
alpha = [1, 0.9, 1]
scale = [0.75, 2, 4]
dataset=[[0,0,0,2,0,2,0,0,1,1,0,0,2],[0,1,0,1,1,1,1,0,0,0,0,1,0],[0,0,1,0,2,0,0,1,2,0,1,1,3]]
for i in range(0,3):
  Mode=mode(dataset[i])
  mean=np.mean(dataset[i])
  var=np.var(dataset[i])
  x = Mode
  x=np.linspace(0,4,100)
  y1 = gamma.pdf(x, a=alpha[i], scale=scale[i])
  plt.plot(x, y1)
  plt.title("Corp prior vs Mode")
  plt.show()
  MAP=(alpha[i]-1+mean)
  MAP=MAP/((1/scale[i])+1)
  plt.axvline(MAP, ymin = 0, ymax=1, linewidth = 2)
  plt.title("MAE")
  plt.show()
  plt.plot(x, MAP*y1, "r-", )
  plt.title("Corp posterior vs Mode")
  plt.show()
```

Gamma prior is conjugate to Poisson

$$X_i \sim P(\lambda) \tag{3.1.31}$$

X follows poisson distribution with mean λ

$$P(X_i|\lambda) = \frac{e^{-\lambda}\lambda^{X_i}}{X_i!} \tag{3.1.32}$$

$$\lambda \sim gamma(\alpha, \beta) \tag{3.1.33}$$

$$P(\lambda) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda} \tag{3.1.34}$$

$$\propto \lambda^{\alpha-1}e^{-\beta\lambda} \tag{3.1.35}$$

$$P(\lambda|X) = \frac{P(X|\lambda)P(\lambda)}{P(X)} \tag{3.1.36}$$

$$\implies \propto P(X|\lambda)P(\lambda) \tag{3.1.37}$$

$$P(X|\lambda) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{X_i}}{X_i!} \tag{3.1.38}$$

$$\implies \frac{e^{-N\lambda}\lambda^{X_1+X_2+...+X_N}}{\prod_{i=1}^{n} X_i!} \tag{3.1.39}$$

$$\implies \propto e^{-N\lambda}\lambda^{X_1+X_2+...+X_N} \tag{3.1.40}$$

$$\propto e^{-N\lambda}\lambda^{N\overline{X}} \tag{3.1.41}$$

Substituting (3.1.41),(3.1.35) in (3.1.37) we get,

$$P(\lambda|X) \propto e^{-N\lambda}\lambda^{N\overline{X}}\lambda^{\alpha-1}e^{-\beta\lambda} \tag{3.1.42}$$

$$\implies \lambda^{N\overline{X}+\alpha-1}e^{-(N+\beta)\lambda} \tag{3.1.43}$$

$$\sim gamma(N\overline{X} + \alpha, \beta + N) \tag{3.1.44}$$

From (3.1.44) it can be seen that we have prooved that Gamma Prior is conjugate to Poisson

4 Question 4

You are provided hourly rental data spanning two years (Data (training and test) available here). The training set is comprised of the first 19 days of each month, while the test set is the 20th to the end of the month. You must predict the total count of bikes rented during each hour covered by the test set, using only information available prior to the rental period. Fit a Poisson regression model to the count data (output). Treat year, month, weekday, hour, holiday, weather, atemp, humidity, windspeed etc. as input features that are combined linearly to determine the rate parameter of the Poisson distribution. Create a 80-20 split of the train data into training, and validation.

(a) Explain maximum likelihood estimation in poisson regression and derive the loss function which is used to estimate the parameters.

(b) Find statistics of the dataset like mean count per year, month etc.

(c) Plot count against any 5 features.

(d) Apply L1 and L2 norm regularization over weight vectors, and find the best hyper-parameter settings for the mentioned problem using validation data and report the accuracy on test data for no regularization, L1 norm regularization and L2 norm regularization.

(e) Determine most important features determining count of bikes rented.

4.1 Solution

(a) Maximum Likelihood Estimation
Poisson regression model is a generalized linear model with poisson error and a log link function. In poisson regression the parameters can be estimated by using Maximum Likelihood Estimate which is of the form,

$$L(\lambda, y) = \Pi_{i=1}^{n} \left[e^{-\lambda_i} \frac{{\lambda_i}^{y_i}}{y_i!} \right] \tag{4.1.1}$$

where,

$$\lambda = (\lambda_1, \lambda_2, \cdots)^T \tag{4.1.2}$$

$$y = (y_1, y_2, \cdots)^T \tag{4.1.3}$$

Now taking log of (4.1.1) and using $\lambda_i = e^{x_i^T \beta}$,

$$l(\lambda, y) = -\sum_{i}^{n} exp(e^{x_i^T \beta}) + \sum_{i}^{n} y_i ln(exp(e^{x_i^T \beta})) - \sum_{i}^{n} ln(y!) \tag{4.1.4}$$

The above equation represents the log likelihood of poisson regression. Differentiating it respect to β and using the iterative weighted least square estimation,

$$\beta = (X^T W X)^{-1} X^T W q \tag{4.1.5}$$

where W=$diag(\lambda_1, \lambda_2, \lambda_3...)$

(b) Statistics of the dataset