Topics in Probabilistic Modeling & Inference (CS698X), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 2

QUESTION 1

Student Name: Piyush Kumar Gaurav
Roll Number: 20104442
Date: April 18, 2021

My solution to problem 1, Part-1
Expected Complete Data log Posterior (ECDLP) is the Expectation of the log posterior distribution of θ (i.e. parameters) using the complete data (Initial data $[\mathcal{D}_0] \cup$ Query Data pool $[\mathcal{D}_P]$). Since the labels of $\mathcal{X}_P$ is not known, the expectation is taken over the PPD ($p(\mathcal{Y}_P|\mathcal{X}_P, \mathcal{D}_0)$). Now, at best we can utilize complete data ($[\mathcal{D}_0] \cup \ \mathcal{D}_P]$) to derive ECDLP. Since we are allowed to pick a subset of $\mathcal{D}_P$, we aim to pick most useful set from $\mathcal{X}_P$ which best approximates the ECDLP.

Complete Data log posterior can be represented as:

$$\mathop{\mathbb{E}}_{\mathcal{Y}_P} log(p(\theta|\mathcal{D}_O \cup (\mathcal{X}_P, \mathcal{Y}_P))) = log(p(\theta|\mathcal{D}_O)) + \sum_{m=1}^{M} \left(\underbrace{\mathop{\mathbb{E}}_{y_m} log(p(y_m|x_m, \theta)) - \mathbb{H}[log(p(y_m|x_m, \mathcal{D}_O))]}_{\mathcal{L}_m(\theta)} \right) \tag{1}$$

Hence the batch can be chosen such that the ECDLP is best approximated by the most useful set of $\mathcal{L}_m(\theta)$. If we define, $\mathcal{L}(w) = \sum_m w_m \mathcal{L}_m(\theta)$ then the elements of most useful batch will have $w_m = 1, else\ 0$. Also the no. of $(w_m = 1)$ cannot exceed the budget (i.e. b). Thus, the optimization problem is

$$\begin{aligned} w^* &= \underset{w}{minimize} \ \|\mathcal{L} - \mathcal{L}(w)\|^2 \\ &subject\ to \ \ w_m \in \{0, 1\} \ \forall \ m \end{aligned} \tag{2}$$

Hence, the $\mathcal{L}(w)$ which best approximates $\mathcal{L}$ will have least value of $\|\mathcal{L} - \mathcal{L}(w)\|$. Intuition: $\mathcal{L}$ is like a resultant vector obtained by summing M no. of "$\mathcal{L}_m$" vectors. Instead of selecting all M vectors we have to select only b vectors such that their resultant sum is as close as possible to $\mathcal{L}$

My solution to problem 1, Part-2

Here, the idea is to choose a coreset of data from the pool that best represents the complete data manifold. Intuitively, $\mathcal{L}$ is resultant sum of $\mathcal{L}_m$ vectors (where each one is a vector in function space). In every iteration an appropriate $\mathcal{L}_m$ vector is chosen such that the new vector (or $\mathcal{L}_m$) is in the direction of residual error. In order to achieve this the paper transforms the Sparse optimization to a form appropriate for Frank Wolf Optimization. The Frank Wolf Optimization solves a convex optimization problem for a function in multi dimension space in a region (polytope) with vertex as each data points. The algorithm is initialized from the vertex with min value of function. Gradient function is computed at that vertex and this function then acts as a replacement to the original function. The argmin over the gradient function is computed. A line search (convex combination) is performed between the current vertex and the argmin vertex. After several iteration the solution is thus a convex combination of all the selected vertices (data points).

So here we are trying to optimize the function $\|\mathcal{L}-\mathcal{L}(w)\|$ over w . In this process, Earlier w_m was either 0 or 1 but now it can attain any non negative value. Also the cardinality constraint of $\mathcal{L}$ (i.e. no. of elements = M) is now a polytope constraint in an M-dimensional space where each vertex of the polytope is $\sigma_m/\sigma = \|\mathcal{L}_m\|/\|\mathcal{L}\|$ i.e. the magnitude of initial objective function corresponding to each data point in Pool set. The Frank Wolf Optimization solves the objective function $(\mathbf{1}-\mathbf{w})^T\mathbf{K}(\mathbf{1}-\mathbf{w})$ which is a convex function in w. $\mathbf{K}$ is the kernel matrix representing similarity between all $\mathcal{L}_m$

My solution to problem 1, Part-3

The paper demonstrates a closed form solution for the two popular models namely Linear regression and Probit Regression. For other type of models where a close for acquisition function cannot be derived (mainly due to no conjugate models) adopts Variational Inference approximation with mean field Gaussian approximation.

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Piyush Kumar Gaurav
Roll Number: 20104442
Date: April 18, 2021

QUESTION
2

My solution to problem 2
We have N scalar observations $x_1, x_2, ..., x_N$ drawn i.i.d. from a Gaussian distribution.

$$x_i \sim \mathcal{N}(x|\mu, \beta^{-1})$$

The likelihood for X (all x's) can be defined as

$$Likelihood = p(\mathbf{X}, \mu) = \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1})$$

The mean μ has a Gaussian prior

$$\mu \sim \mathcal{N}(\mu|\mu_0, s_0)$$

The Precision β has a Inverse Gamma prior

$$\beta \sim Gamma(\beta|a, b)$$

Conditional Posterior for μ (considering β s_0, a, b as constant)

$$\begin{aligned} p(\mu|\mathbf{X}, \beta, s_0, a, b) &= \frac{p(\mu|\mathbf{X}, \beta)p(\mu)}{p(\mathbf{X})} \\ &\propto \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1})\mathcal{N}(\mu|\mu_0, s_0) \\ &= \mathcal{N}(\mu|\mu_N, \sigma_N^2) \end{aligned} \tag{3}$$

where

$$\frac{1}{\sigma_N^2} = N\beta + \frac{1}{s_0^2}$$

$$\mu_N = \frac{1}{N\beta\sigma_0^2} + \frac{N\beta\sigma_0^2}{N\beta\sigma_0^2 + 1}$$

Conditional Posterior for β (considering σ s_0, a, b as constant)

$$\begin{aligned} p(\beta|\mathbf{X}, \mu, s_0, a, b) &= \frac{p(\beta|\mathbf{X}, \mu)p(\beta)}{p(\mathbf{X})} \\ &\propto \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1})Gamma(\beta|a, b) \\ &\propto \prod_{i=1}^{N} \mathcal{N}(x_i|\mu, \beta^{-1})[(\beta)^{(a-1)}exp(-b\beta)] \\ &= Gamma\left(\beta|a + \frac{N}{2}, b + \frac{\sum_{n=1}^{N}(x_n - \mu)^2}{2}\right) \\ &= (\beta)^{(a+\frac{N}{2}-1)}exp\left(-\left(b + \frac{\sum_{n=1}^{N}(x_n - \mu)^2}{2}\right)\beta\right) \end{aligned} \tag{4}$$

The Joint Posterior of μ and β using Gibbs Sampling Algorithm can be summarized as below

1. Initialize $\beta^{(0)}$
2. For s = 1 to S
   a. Draw random sample of $\mu^{(s)}$ from $\mathcal{N}(\mu|\mu_N, \sigma_N^2)$
   where

$$\frac{1}{\sigma_N^2} = N\beta^{(s-1)} + \frac{1}{s_0^2}$$

$$\mu_N = \frac{1}{N\beta^{(s-1)}\sigma_0^2} + \frac{N\beta^{(s-1)}\sigma_0^2}{N\beta^{(s-1)}\sigma_0^2 + 1}$$

   b. Draw random sample of $\beta^{(s)}$ from

$$\left((\beta)^{(s-1)}\right)^{(a+\frac{N}{2}-1)} exp\left(-\left(b + \frac{\sum_{n=1}^{N}(x_n - \mu^{(s)})^2}{2}\right)\beta^{(s-1)}\right)$$

3. Repeat Step 2 until we got enough samples for μ and β

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 2

QUESTION 3

Student Name: Piyush Kumar Gaurav
Roll Number: 20104442
Date: April 18, 2021

My solution to problem 3
The linear regression model is defined as $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$ where $\mathbf{y}$ is an $NX1$ vector such that $\mathbf{y} = [y_1, y_2, ..., y_N]^T$, $\mathbf{X}$ is an NXD matrix such that $\mathbf{X} = [\mathbf{x}_1{}^T, \mathbf{x}_2{}^T, ..., \mathbf{x}_N{}^T]^T$ (each row of the matrix $\mathbf{X}$ is a $DX1$ vector), $\mathbf{w}$ is a $DX1$ vector $[w_1, w_2, ..., w_D]^T$ and ϵ is noise of $NX1$ vector. Following are the distributions over some of the paramaters:

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2)$$

$$w_d \sim \mathcal{N}(0, \sigma^2 \kappa_{\gamma_d})$$

where $\kappa_{\gamma_d} = \gamma_d v_1 + (1 - \gamma_d) v_0$ also $0 < v_0 < v_1$ and $d = 1, 2, ..., D$

$$\sigma^2 \sim IG(v/2, v\lambda/2)$$

$$\gamma_d \sim Bernoulli(\theta)$$

$$\theta \sim Beta(a_0, b_0)$$

The steps for EM algorithm is defined as follows

1. Initialize $\gamma^{(0)}, \sigma^{2^{(0)}}, \theta^{(0)}$. (Lets denote this set of parameters as $\Theta^{(0)}$)
   where $\gamma^{(0)} = [\gamma_1^{(0)}, \gamma_2^{(0)}, \gamma_3^{(0)}, ..., \gamma_D^{(0)}]$
2. For t = 1 to T (until convergence)
   a. **Conditional Posterior of Latent Variable:**For each data point, the Conditional Posterior of latent variable ($\mathbf{w}$) given the current parameters ($\gamma^{(t-1)}, \sigma^{2^{(t-1)}}, \theta^{(t-1)}$ or simply $\Theta^{(t)}$) is defined as :

$$p(\mathbf{w}_n^{(t)} | \mathbf{y}_n, \mathbf{x}_n, \gamma^{(t-1)}, \sigma^{2^{(t-1)}}, \theta^{(t-1)}) \propto \underbrace{\prod_{d=1}^{D} p(\mathbf{w}_{d_n}^{(t)} | \Theta^{(t-1)})}_{Prior(\mathbf{w})} \underbrace{p(\mathbf{y}_n | \mathbf{w}_n^{(t)}, \mathbf{x}_n, \Theta^{(t-1)})}_{Likelihood(\mathbf{w})}$$

   where $Prior(\mathbf{w}) \sim \mathcal{N}(\mathbf{0}, diag(\sigma^2(\kappa_{\gamma_d}))_{DXD})$ and $Likelihood(\mathbf{w}) = \mathcal{N}(\mathbf{x}_n^T \mathbf{w}, \sigma^2)$

   Hence for the complete data, the Conditional Posterior of latent variable ($\mathbf{w}$) given the current parameters

$$\begin{aligned} p(\mathbf{w}^{(t)} | \mathbf{y}, \mathbf{X}, \gamma^{(t-1)}, \sigma^{2^{(t-1)}}, \theta^{(t-1)}) &\propto \underbrace{\prod_{d=1}^{D} p(\mathbf{w}_{d_n}^{(t)} | \Theta^{(t-1)})}_{Prior(\mathbf{w})} \underbrace{\prod_{n=1}^{N} p(\mathbf{y}_n | \mathbf{w}_n^{(t)}, \mathbf{x}_n, \Theta^{(t-1)})}_{Likelihood(\mathbf{w})} \\ &\propto \mathcal{N}(\mathbf{w} | \mathbf{0}, \mathbf{\Sigma}_K)\, \mathcal{N}(\mathbf{y} | \mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I}_N) \end{aligned} \tag{5}$$

Hence Using expression for the Posterior Distribution (as discussed in Prob. Linear Regression Lecture) we get,

$$Posterior(\mathbf{w}^{(t)}) = p(\mathbf{w}^{(t)}|\mathbf{y}, \mathbf{X}, \gamma^{(t-1)}, \sigma^{2^{(t-1)}}, \theta^{(t-1)}) \sim \mathcal{N}(\mu_N, \mathbf{\Sigma}_N) \tag{6}$$

where

$$\mathbf{\Sigma}_N = (\sigma^{2^{(t-1)}}(\mathbf{X}^T\mathbf{X}) + diag(\sigma^{2^{(t-1)}}(\kappa_{\gamma_d}))_{DXD})^{-1} = (\sigma^{2^{(t-1)}}(\mathbf{X}^T\mathbf{X}) + \mathbf{\Sigma}_K)^{-1}$$

$$\mu_N = \mathbf{\Sigma}_N[\frac{1}{\sigma^{2^{(t-1)}}}(\mathbf{X}^T\mathbf{y})]$$

b. **Expectation of Complete Data Log likelihood:** $log(p(\mathbf{w}, \mathbf{y}|\mathbf{X}, \sigma^{2^{(t-1)}}.\gamma^{(t-1)}))$ can be written as
(Note: $\mathbf{w}$ represents $\mathbf{w}^{(t)}$)

$$\begin{aligned} log(p(\mathbf{w}, \mathbf{y}|\mathbf{X}, \sigma^{2^{(t-1)}}, \gamma^{(t-1)})) &= log((\mathbf{y}|\mathbf{w}, \mathbf{X}, \sigma^{2^{(t-1)}})) + log(\mathbf{w}|\sigma^{2^{(t-1)}}, \gamma^{(t-1)}) \\ &= -\frac{N}{2}log(2\pi\sigma^{2^{(t-1)}}) - \frac{1}{2\sigma^{2^{(t-1)}}}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &- \frac{D}{2}log(2\pi\sigma^{2^{(t-1)}}) - \frac{1}{2}\mathbf{w}^T\Sigma_K^{-1}\mathbf{w} - \sum_{d=1}^{D}\frac{1}{2}log(\kappa_{\gamma_d}^{(t-1)}) \end{aligned} \tag{7}$$

The Expectation of CLL w.r.t. the Posterior of $\mathbf{w}$ derived in previous step is

$$\begin{aligned} \mathbb{E}\left[log(p(\mathbf{w}, \mathbf{y}|\mathbf{X}, \sigma^{2^{(t-1)}}, \gamma^{(t-1)}))\right] &= -\frac{1}{2\sigma^{2^{(t-1)}}}(\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbb{E}\left[\mathbf{w}\right]) \\ &- \frac{1}{2}(Trace((\frac{\mathbf{X}^T\mathbf{X}}{\sigma^{2^{(t-1)}}} + \Sigma_K^{-1})\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right])) \\ &- \frac{N+D}{2}log(2\pi\sigma^{2^{(t-1)}}) - \frac{1}{2}\sum_{d=1}^{D}log(\kappa_{\gamma_d{}^{(t-1)}}) \end{aligned} \tag{8}$$

where

$$\mathbb{E}\left[\mathbf{w}\right] = (\sigma^{2^{(t-1)}}(\mathbf{X}^T\mathbf{X}) + \mathbf{\Sigma}_K)^{-1}\left[\frac{1}{\sigma^{2^{(t-1)}}}\left(\mathbf{X}^T\mathbf{y}\right)\right]$$

$$\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right] = (\sigma^{2^{(t-1)}}(\mathbf{X}^T\mathbf{X}) + \mathbf{\Sigma}_K)^{-1} + \mathbb{E}\left[\mathbf{w}\right]\mathbb{E}\left[\mathbf{w}\right]^T$$

(Note: $\mathbf{w}$ represents $\mathbf{w}^{(t)}$)

c. **Maximization Step:** Now we maximize the Expected (Expectation taken over $Posterior(\mathbf{w}_n^{(t)})$ distribution) MAP w.r.t. parameters $(\gamma^{(t-1)}, \sigma^{2^{(t-1)}}, \theta^{(t-1)})$

$$\gamma_d^{(t)} = \underset{\gamma_d}{argmax}\mathbb{E}\left[log(p(\mathbf{y}, \mathbf{w}^{(t)}|\mathbf{X}, \gamma_d, \sigma^2, \theta)) + log(p(\gamma_d|\theta))\right]$$

$$\gamma_d^{(t)} = \underset{\gamma_d}{argmax}\mathbb{E}\left[log(p(\mathbf{y}, \mathbf{w}^{(t)}|\mathbf{X}, \gamma_d, \sigma^2, \theta))\right] + log(p(\gamma_d|\theta)) \tag{9}$$

where

$$log(p(\gamma_d|\theta)) = \gamma_d log(\theta) + (1 - \gamma_d)log(1 - \theta)$$

Similarly for σ^2

$$\sigma^{2(t)} = \underset{\sigma^2}{argmax}\mathbb{E}\left[log(p(\mathbf{y}, \mathbf{w}^{(t)}|\mathbf{X}, \gamma_d, \sigma^2, \theta)) + log(p(\sigma^2|v, \lambda))\right]$$

$$\sigma^{2(t)} = \underset{\sigma^2}{argmax}\mathbb{E}\left[log(p(\mathbf{y}, \mathbf{w}^{(t)}|\mathbf{X}, \gamma_d, \sigma^2, \theta))\right] + log(p(\sigma^2|v, \lambda)) \tag{10}$$

where

$$log(p(\sigma^2|v, \lambda)) = -\left(\frac{v}{2} + 1\right) log(\sigma^2) - \frac{v\lambda}{2\sigma^2}$$

Similarly for θ

$$\theta^{(t)} = \underset{\theta}{argmax}\mathbb{E}\left[log(p(\mathbf{y}, \mathbf{w}^{(t)}|\mathbf{X}, \gamma_d, \sigma^2, \theta)) + log(p(\theta|a_0, b_0))\right]$$

$$\theta^{(t)} = \underset{\theta}{argmax}\mathbb{E}\left[log(p(\mathbf{y}, \mathbf{w}^{(t)}|\mathbf{X}, \gamma_d, \sigma^2, \theta))\right] + log(p(\theta|a_0, b_0)) \tag{11}$$

where

$$log(p(\theta|a_0, b_0)) = (a_0 - 1)log(\theta) + (b_0 - 1)log(1 - \theta)$$

From (8) substituting the value of $p(\mathbf{y}, \mathbf{w}^{(t)}|\mathbf{X}, \gamma_d, \sigma^2, \theta)$ in the equations (9), (10), (11) and subsequently differentiating them w.r.t. σ^2 and θ respectively and equating each one to 0 we get,
Update for σ^2:

$$\sigma^{2(t)} = \frac{\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbb{E}\left[\mathbf{w}\right] + Tr((\mathbf{X}^T\mathbf{X} + diag(\kappa_{\gamma_d})^{-1})\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right]) + v\lambda}{N + D + v + 2} \tag{12}$$

Update for θ

$$\theta^{(t)} = \frac{\sum_{d=1}^{D}\gamma_d + a_0 - 1}{D + a_0 + b_0 - 2} \tag{13}$$

Update for γ_d given θ. Since $\gamma_d \in \{0, 1\}$ We can simply write

$$\gamma_d^{(t)} = \underset{\gamma_d \in \{0,1\}}{argmax} - \frac{1}{2\sigma^2\kappa_{\gamma_d}}\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right]_{dd} - \frac{1}{2}log(\kappa_{\gamma_d}) + \gamma_d log(\theta) + (1 - \gamma_d)log(1 - \theta) \tag{14}$$

(Note: $\mathbf{w}$ represents $\mathbf{w}^{(t)}$)

The E.M. Algorithm

1. Initialize $\gamma^{(0)}, \sigma^{2(0)}, \theta^{(0)}$. (Lets denote this set of parameters as $\Theta^{(0)}$) where $\gamma^{(0)} = [\gamma_1^{(0)}, \gamma_2^{(0)}, \gamma_3^{(0)}, ..., \gamma_D^{(0)}]$

2. For t = 1 to T (until convergence)

   a. **Computation of Posterior of weights (LV):**

$$p(\mathbf{w}^{(t)}|\mathbf{y}, \mathbf{X}, \gamma^{(t-1)}, \sigma^{2(t-1)}, \theta^{(t-1)}) \sim \mathcal{N}(\mu_N, \mathbf{\Sigma}_N)$$

   where

$$\mathbf{\Sigma}_N^{(t)} = (\sigma^{2(t-1)}(\mathbf{X}^T\mathbf{X}) + \mathbf{\Sigma}_K)^{-1}$$

$$\mu_N^{(t)} = \mathbf{\Sigma}_N[\frac{1}{\sigma^{2(t-1)}}(\mathbf{X}^T\mathbf{y})]$$

$$\mathbf{\Sigma}_K) = diag(\sigma^{2(t-1)}(\kappa_{\gamma_d^{(t-1)}}))$$

   b. **Updating Expectation terms:**

$$\mathbb{E}\left[\mathbf{w}\right]^{(t)} = (\sigma^{2(t-1)}(\mathbf{X}^T\mathbf{X}) + \mathbf{\Sigma}_K)^{-1}\left[\frac{1}{\sigma^{2(t-1)}}\left(\mathbf{X}^T\mathbf{y}\right)\right]$$

$$\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right]^{(t)} = (\sigma^{2(t-1)}(\mathbf{X}^T\mathbf{X}) + \mathbf{\Sigma}_K)^{-1} + \mathbb{E}\left[\mathbf{w}\right]\mathbb{E}\left[\mathbf{w}\right]^T$$

   c. **Maximization Step:** Maximization (Point estimation) of hyper parameters

$$\sigma^{2(t)} = \frac{\mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbb{E}\left[\mathbf{w}\right]^{(t)} + Tr((\mathbf{X}^T\mathbf{X} + diag(\kappa_{\gamma_d}^{(t-1)})^{-1})\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right]^{(t)}) + v\lambda}{N + D + v + 2}$$

$$\theta^{(t)} = \frac{\sum_{d=1}^{D}\gamma_d^{(t-1)} + a_0 - 1}{D + a_0 + b_0 - 2}$$

$$\gamma_d^{(t)} = \underset{\gamma_d \in \{0,1\}}{argmax} - \frac{1}{2\sigma^{2(t)}\kappa_{\gamma_d^{(t-1)}}}\mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right]_{dd}^{(t)} - \frac{1}{2}log(\kappa_{\gamma_d}^{(t-1)}) + \gamma_d^{(t-1)}log(\theta)^{(t)} + (1 - \gamma_d^{(t-1)})log(1 - \theta^{(t)})$$

3. Check for Convergence of σ^2, θ and γ_d. If not converged Repeat step 2

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 2

QUESTION 4

Student Name: Piyush Kumar Gaurav
Roll Number: 20104442
Date: April 18, 2021

My solution to Problem 4 Part 1
Consider that the training data be $(\mathbf{X}, \mathbf{y}) \equiv \{\mathbf{x}_n, y_n\}_{n=1}^N$. The mapping between input and output is carried out using a non linear function "f" such that

$$\mathbf{f} = [\mathbf{f}_1\ \mathbf{f}_2\ ...\mathbf{f}_N]^T = [\mathbf{f}(x_1)\ \mathbf{f}(x_2)\ ...\mathbf{f}(x_N)]^T$$

Likelihood is given by

$$p(\mathbf{y}|\mathbf{X}, f) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{f}_n, \sigma^2) = \mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I}_N)$$

Prior is given by

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$$

where $\mathbf{0} = [0\ 0\ ...0]$ is a $NX1$ vector and $\mathbf{K}$ is a NXN Kernel matrix such that

$$\mathbf{K} = \begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \dots & \kappa(x_1, x_N) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \dots & \kappa(x_2, x_N) \\ \vdots & \vdots & \ddots & \\ \kappa(x_N, x_1) & \kappa(x_N, x_2) & \dots & \kappa(x_N, x_N) \end{bmatrix}$$

Posterior will thus be

$$\begin{aligned} p(\mathbf{f}|\mathbf{y}) &\propto \underbrace{\mathcal{N}(\mathbf{0}, \mathbf{K})}_{Prior}\underbrace{\mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I}_N)}_{Likelihood} \\ &\propto exp\left(-\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\right)\ exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{f})^T[\sigma^2\mathbf{I}_N]^{-1}(\mathbf{y}-\mathbf{f})\right) \\ &\propto exp\left(-\frac{1}{2}(\mathbf{f}^T[\mathbf{K}^{-1} + (\sigma^2\mathbf{I}_N)^{-1}]\mathbf{f} - 2\mathbf{f}^T(\sigma^2\mathbf{I}_N)^{-1}\mathbf{y} + constant\right) \end{aligned} \tag{15}$$

Comparing the above equation with standard Gaussian equation where the equation is of the form $exp(-\frac{1}{2}(\mu^T\Sigma^{-1}\mu - 2\mathbf{x}^T\Sigma^{-1}\mu + constant))$
Comparing the form (completing the squares) we get the Posterior as

$$Posterior = \mathcal{N}(\mu_N, \Sigma_N^{-1})$$

where

$$\Sigma_N = \left[\mathbf{K}^{-1} + \left(\frac{1}{\sigma^2}\mathbf{I}_N\right)\right]^{-1}$$

$$\mu_N = \left[\mathbf{K}^{-1} + \left(\frac{1}{\sigma^2}\mathbf{I}_N\right)\right]^{-1}\left(\frac{1}{\sigma^2}\mathbf{I}_N\right)\mathbf{y}$$

My solution to Problem 4 Part 2

As we it is evident through the following graphs, if we increase l the Kernel function becomes more and more non informative and exponential tends to go towards 0. We can also see in a way that with increase in l, the term l^2 dominates over similarity $(x - x^{'})^2$ and hence kernel function tends to make prior more and more uniform. The effect of prior is also seen on posterior (although slightly). When l is low (i.e. prior is more informative) posterior follows the actual function more precisely.

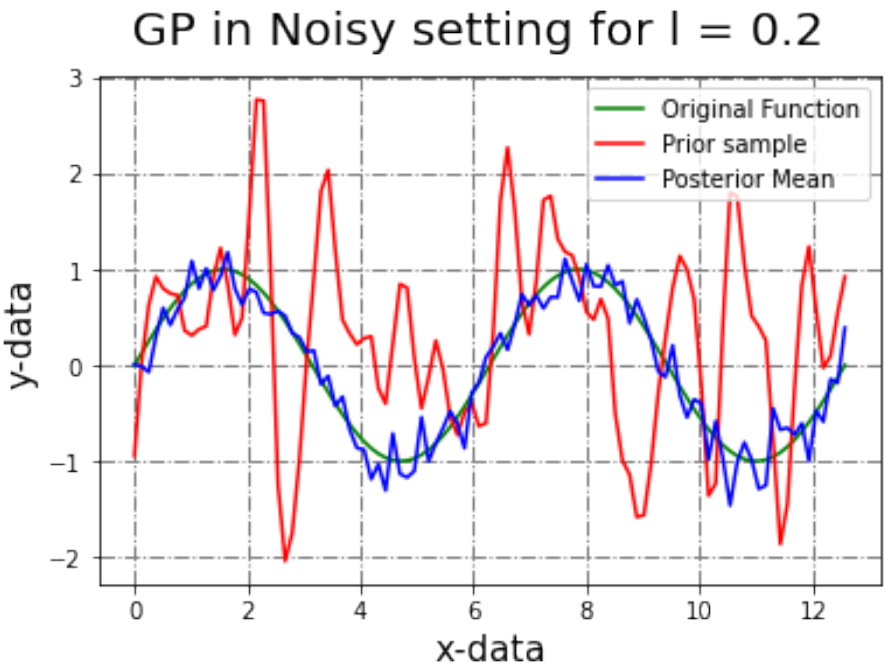

Figure 1: Plot showing original function , prior sample and Posterior Mean for GP in noisy setting for l=0.2

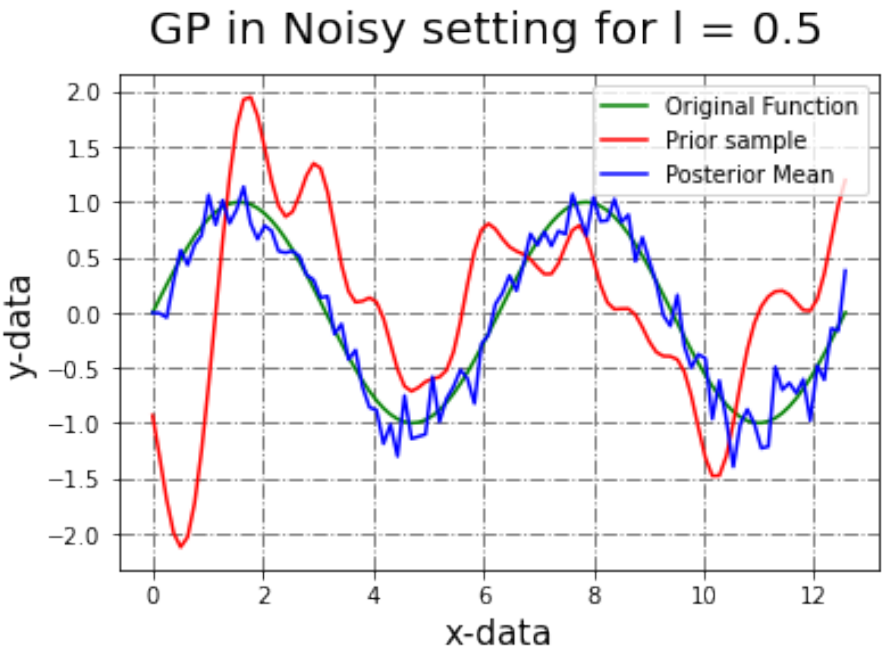

Figure 2: Plot showing original function , prior sample and Posterior Mean for GP in noisy setting for l=0.5

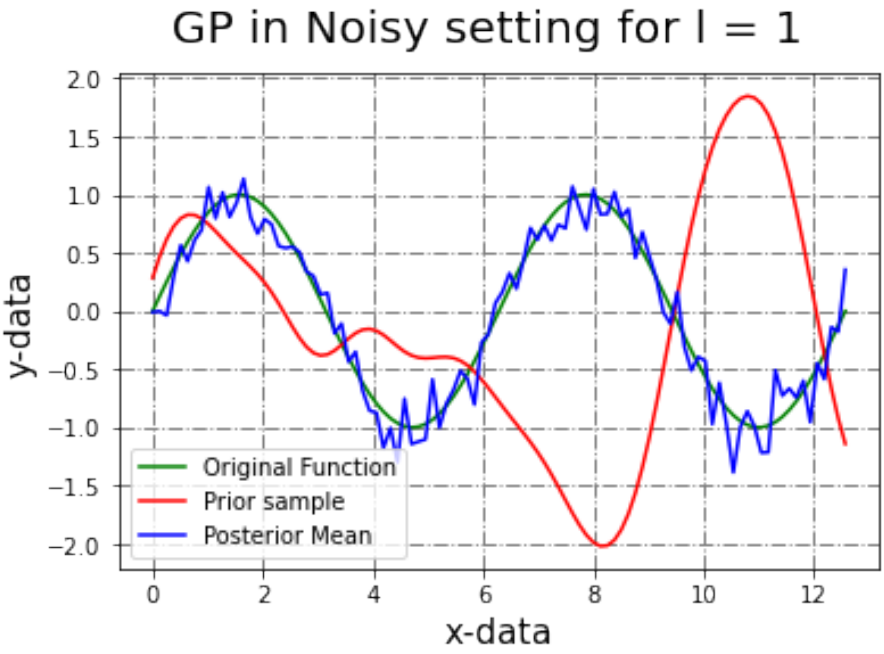

Figure 3: Plot showing original function , prior sample and Posterior Mean for GP in noisy setting for l=1

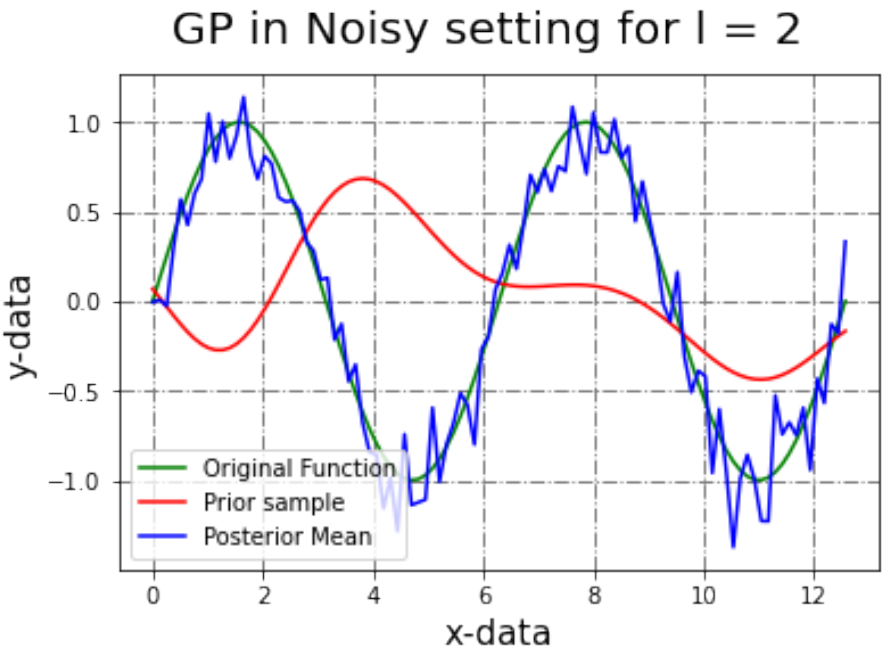

Figure 4: Plot showing original function , prior sample and Posterior Mean for GP in noisy setting for l=2

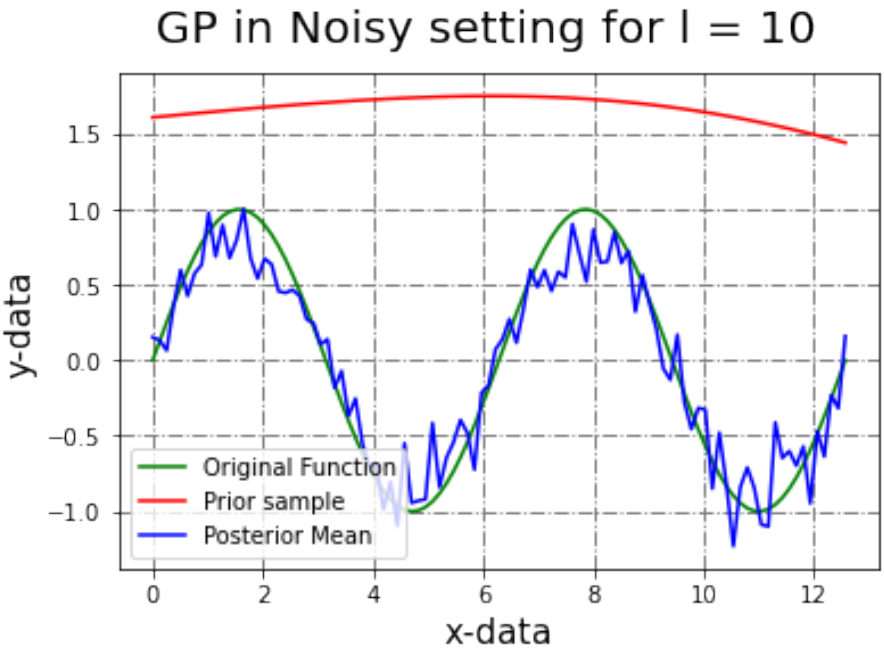

Figure 5: Plot showing original function , prior sample and Posterior Mean for GP in noisy setting for l=10

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2019
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Piyush Kumar Gaurav
Roll Number: 20104442
Date: April 18, 2021

QUESTION

5

My solution to problem 5
In a noisy setting, Gaussian Regression Model can be represented as

$$y_n = f(x_n) + \epsilon_n$$

where function f is modelled by $GP(0, \kappa)$ and $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$

Whereas in a noiseless setting, the Gaussian Process can be illustrated as

$$y_n = f(x_n) = f_n$$

Now, consider N training points as $(\mathbf{X}, \mathbf{f}) \equiv \{x_n, f_n\}_{n=1}^N$. The PPD for the new input will thus be

$$p(y_*|x_*, \mathbf{X}, \mathbf{f}) = \mathcal{N}(f_*|\mathbf{k}_*^T, \mathbf{K}^{-1}\mathbf{f}, \kappa(x_*, x_*) - \mathbf{k}_*^T, \mathbf{K}^{-1}\mathbf{k}_*^T) \tag{16}$$

where $\mathbf{K}$ is a NXN kernel matrix of all the training inputs, $\mathbf{k}_*$ is $NX1$ kernel vector where each element denotes similarity of x_* w.r.t. each of the N training inputs. The computation of PPD has a complexity of $O(N^3)$ and hence computation becomes very costly as data size increases.
In order to speed up the computation of PPD, we consider a smaller number (M) pseudo inputs. Consider a set of pseudo input - output pairs $\mathbf{Z}, \mathbf{t}$ modelled by the same Gaussian Process. Here, $\mathbf{Z} = \{z_1, z_2 \ldots, z_M\}$ and $\mathbf{t} = \{t_1, t_2, \ldots, t_M\}$ with $M << N$, Now considering the pseudo data we can predict the value of f_n (output of each training data) from PPD by making use of $\mathbf{x}_n, \mathbf{Z}, \mathbf{t}$. Hence,

$$p(f_n|x_n, \mathbf{Z}, \mathbf{t}) = \mathcal{N}(f_n|\tilde{\mathbf{k}}_n^T, \tilde{\mathbf{K}}^{-1}\mathbf{t}, \kappa(x_n, x_n) - \tilde{\mathbf{k}}_n^T, \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{k}}_n) \tag{17}$$

where $\tilde{\mathbf{K}}$ is a MXM kernel matrix of the pseudo inputs, $\tilde{\mathbf{k}}_n$ is $MX1$ kernel vector where each element denotes similarity of x_n w.r.t. each of the M pseudo inputs $(z_1, z_2, \ldots, z_M)$.

With above details in hand, we need to compute PPD in order to predict output y_* by making use of its input x_*, $\mathbf{X}, \mathbf{f}, \mathbf{Z}$ i.e. $p(y_*|x_*, \mathbf{X}, \mathbf{f}, \mathbf{Z})$. Here, $\mathbf{Z}$ is assumed to be known where as $\mathbf{t}$) needs to be integrated out (marginalized) as it is unknown. We can write the PPD as

$$p(y_*|x_*, \mathbf{X}, \mathbf{f}, \mathbf{Z}) = \int p(y_*|x_*, \mathbf{X}, \mathbf{f}, \mathbf{Z}, \mathbf{t})\, p(\mathbf{t}|\mathbf{X}, \mathbf{f}, \mathbf{Z}) d\mathbf{t} \tag{18}$$

As per Baye's rule we can write

$$p(\mathbf{t}|\mathbf{X}, \mathbf{f}, \mathbf{Z}) \propto p(\mathbf{f}|\mathbf{X}, \mathbf{Z}, \mathbf{t})\, p(\mathbf{t}|\mathbf{Z}) \tag{19}$$

Now, using (17) we can write, the first term in the RHS of (19) as

$$
\begin{aligned}
p(\mathbf{f}|\mathbf{X},\mathbf{Z},\mathbf{t}) &= \prod_{n=1}^{N} p(f_n|x_n,\mathbf{Z},\mathbf{t}) \\
&= \mathcal{N}(\mathbf{f}|\mathbf{A}\tilde{\mathbf{K}}^{-1}\mathbf{t},\mathbf{B})
\end{aligned}
\tag{20}
$$

where $\mathbf{B}$ is a NXN diagonal matrix such that the diagonal entries $\mathbf{B}_{nn} = \kappa(x_n,x_n) - \tilde{\mathbf{k}}_*^T,\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{k}}_*^T)$ and $\mathbf{A}$ is a NXM matrix such that each row is a kernel vector corresponding to each input data point

$$
\mathbf{A} = \begin{bmatrix}
\kappa(x_1,z_1) & \kappa(x_1,z_2) & \dots & \kappa(x_1,z_M) \\
\kappa(x_2,z_1) & \kappa(x_2,z_2) & \dots & \kappa(x_2,z_M) \\
\vdots & \vdots & \ddots & \\
\kappa(x_N,z_1) & \kappa(x_N,z_2) & \dots & \kappa(x_N,z_M)
\end{bmatrix}
$$

The second term in the RHS of (19) can be written as

$$
p(\mathbf{t}|\mathbf{Z}) = \mathcal{N}(\mathbf{0},\tilde{\mathbf{K}}) \tag{21}
$$

Using (20) and (21) in equation (19), We can write (by comparing with std. relations like Posterior equation from Linear Regression) ,

$$
p(\mathbf{t}|\mathbf{X},\mathbf{f},\mathbf{Z}) = \mathcal{N}(\mu,\Sigma) \tag{22}
$$

where

$$
\Sigma = \tilde{\mathbf{K}}^{-1}\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A}\tilde{\mathbf{K}}^{-1}
$$

$$
\mu = \Sigma\tilde{\mathbf{K}}^{-1}\mathbf{A}^T\mathbf{B}^{-1}\mathbf{f}
$$

Now in order to compute our primary goal i.e. (18) we utilise (17) and (22) (and by comparing to standard results like - PPD of Linear regression where marginalisation is done for two Gaussian) we can write,

$$
p(y_*|x_*,\mathbf{X},\mathbf{f},\mathbf{Z}) = \mathcal{N}(\mu_f,\Sigma_f) \tag{23}
$$

where

$$
\mu_f = \tilde{\mathbf{k}}_*^T,\tilde{\mathbf{K}}^{-1}\Sigma\tilde{\mathbf{K}}^{-1}\mathbf{A}^T\mathbf{B}^{-1}\mathbf{f}
$$

$$
\Sigma_f = \tilde{\mathbf{k}}_*^T,\tilde{\mathbf{K}}^{-1}\Sigma\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{k}}_* + \kappa(x_*,x_*) - \tilde{\mathbf{k}}_*^T,\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{k}}_*
$$

Regarding Complexity: The final expression ((18)) has a complexity $O(M^2N)$ as compared to the PPD using the training data ((16)) where complexity was $O(N^3)$. Since $M << N$ the complexity drastically reduces. In addition to that in the expression of PPD based on pseudo input B is just a diagonal matrix of size NXN so its inverse not computationally intensive

My Solution to Problem 5 Part 2
We can write

$$p(\mathbf{f}|\mathbf{X},\mathbf{Z}) = \int p(\mathbf{f}|\mathbf{X},\mathbf{Z},\mathbf{t})\, p(\mathbf{t}|\mathbf{Z}) d\mathbf{t}$$

Using (20) we can write

$$\mathbf{f} = \mathbf{A}\tilde{\mathbf{K}}^{-1}\mathbf{t} + \epsilon)$$

Hence ,

$$p(\mathbf{f}|\mathbf{X},\mathbf{Z}) = \mathcal{N}(\mathbf{0}, \mathbf{A}\tilde{\mathbf{K}}^{-1}\mathbf{A} + \mathbf{B})$$

Optimal $\mathbf{Z}$ will thus be

$$\hat{\mathbf{Z}} = argmax_{\mathbf{Z}}\, p(\mathbf{f}|\mathbf{X},\mathbf{Z})$$

$$\begin{aligned} \hat{\mathbf{Z}} &= argmax_{\mathbf{Z}}\, log(p(\mathbf{f}|\mathbf{X},\mathbf{Z})) \\ &= argmax_{\mathbf{Z}}(-log(\mathbf{A}\tilde{\mathbf{K}}^{-1}\mathbf{A} + \mathbf{B}) - log(\mathbf{f}^T(\mathbf{A}\tilde{\mathbf{K}}^{-1}\mathbf{A} + \mathbf{B})^{-1}\mathbf{f})) \end{aligned} \tag{24}$$