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

QUESTION 1

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: April 20, 2021

Batch Bayesian Active Learning

Overview

Introduced the new Bayesian batch active learning approach that overcomes standard greedy procedures problems. In this algorithm produces diverse batches that enable efficient active learning at scale. We derive interpretable closed-form solutions akin to existing active learning procedures for linear models, and generalize to arbitrary models using random projections.

The key idea

Select the next **batch of data that best approximates the Expected complete data log posterior**.
Expected complete data log posterior is defined as follows:

$$\mathop{\mathbb{E}}_{\mathcal{Y}_p}[\log p(\boldsymbol{\theta}|\mathcal{D}_0 \cup (\mathcal{X}_p, \mathcal{Y}_p))] = \log p(\boldsymbol{\theta}|\mathcal{D}_0) + \sum_{m=1}^{M} \underbrace{\left(\mathop{\mathbb{E}}_{\mathbf{y}_p}[\log p(\mathbf{y}_p|\mathbf{x}_p, \boldsymbol{\theta})] + \mathbb{H}(\mathbf{y}_p|\mathbf{x}_p, \mathcal{D}_0) \right)}_{\mathcal{L}_m(\boldsymbol{\theta})} \tag{1}$$

This means that expectation of the parameters given all the data. Now the idea is to find the batch of data $\mathcal{D}'$ in a way that updated log posterior $P(\boldsymbol{\theta}|\mathcal{D}_0 \cup \mathcal{D}')$ best approximates the complete data log posterior $P(\boldsymbol{\theta}|\mathcal{D}_0 \cup \mathcal{D}_p)$.
As the first term depends only on $\mathcal{D}_0$.We can choose the batch that best approximates $\mathcal{L} = \sum_m \mathcal{L}_m$. where $\mathcal{L}_m : \boldsymbol{\Theta} \to \mathcal{R}$ considered as vector in function space. We can achieve this above idea by the following sparse objective.

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \|\mathcal{L} - \mathcal{L}((\boldsymbol{\omega})\|^2 \text{ such that } \omega_m \in \{0, 1\} \quad \forall m, \sum_m \omega_m \leq b \tag{2}$$

Relaxed objective

As solving Eq. (2) is generally intractable, To approximately solve the optimization problem Eq.(2) by relaxing the **binary weight constraint to be non-negative** and replace the **cardinality constraint with a polytope** constraint in Hilbert space.
Let $\sigma_m = \|\mathcal{L}_m\|$, $\sigma = \sum_m \omega_m \sigma_m$ and $\mathbf{K} \in \mathbb{R}^{M \times M}$ be a kernel matrix with $K_{mn} = \langle \mathcal{L}_m, \mathcal{L}_n \rangle$
Then the relaxed objective can be written as follows:

$$\arg\min_{\boldsymbol{\omega}} \ (1 - \boldsymbol{\omega})^T \mathbf{K} (1 - \boldsymbol{\omega}) \text{ subject to } \omega_m \geq 0 \quad \forall m, \sum_m \omega_m \sigma_m = \sigma \tag{3}$$

where we used $\|\mathcal{L} - \mathcal{L}((\boldsymbol{\omega})\|^2 = (1-\boldsymbol{\omega})^T \mathbf{K} (1-\boldsymbol{\omega})$.The polytope has vertices $\{\sigma/\sigma_m \mathbf{1}_m\}_{m=1}^M$ and contains the point $\boldsymbol{\omega} = [1, 1, ..., 1]^T$ Now we can solve equation3 using Frank-Wolfe algorithm which intern solves our objective Eq2.
The optimum of Eq(3) is attained at vertices of polytope. Frank-Wolfe algorithm runs for b

number of iterations to get $\boldsymbol{\omega}^*$, the weights are iteratively updated along the f-th vertex of polytope. Frank-Wolfe algorithm requires the choice of inner product1 $\langle \mathcal{L}_m \mathcal{L}_n \rangle$ and the norm $\sigma_n = \|\mathcal{L}_n\|$
choice1: Fisher inner product

$$\langle \mathcal{L}_m \mathcal{L}_n \rangle_{\hat{\pi},\mathcal{F}} = \mathop{\mathbb{E}}_{\hat{\pi}}[\nabla_{\boldsymbol{\theta}} \mathcal{L}_n(\boldsymbol{\theta})^T \nabla_{\boldsymbol{\theta}} \mathcal{L}_m(\boldsymbol{\theta})] \tag{4}$$

Advantage of this choice of inner product is that for specific models this leads to **simple, interpretable** expressions.
Choice2: d Euclidean inner product

$$\langle \mathcal{L}_m \mathcal{L}_n \rangle_{\hat{\pi},2} = \mathop{\mathbb{E}}_{\hat{\pi}}[\mathcal{L}_n(\boldsymbol{\theta}) \mathcal{L}_m(\boldsymbol{\theta})] \tag{5}$$

Advantage of this choice of inner product is that only **requires tractable likelihood** computations. For this we can use black-box method
Once we get the continuous weight from the Frank-Wolfe algorithm we project them we project them back to the feasible space using

$$\text{if } \omega_m^* > 0 : \tag{6}$$
$$\hat{\omega}_m^* = 1 \tag{7}$$
$$\text{else} \tag{8}$$
$$\hat{\omega}_m^* = 0 \tag{9}$$

Models having closed form expression:
By using the weighted Fisher inner product from we get the closed form expression for the two of the following models:

1. Bayesian linear regression
2. Probit regression

For the models for which we can't compute the gradient we used **random feature projection** which approximates the key quantities required. This method which resolves gradient issue and also linearly complex with the pool set $|\mathcal{P}|$. For this approach we use projections for the weighted Euclidean inner product.
The appropriate projection which can be calculated for any model with tractable likelihod

$$\hat{\mathcal{L}_n} = \frac{1}{\sqrt{J}}[\mathcal{L}_n(\boldsymbol{\theta}_1, ..., \mathcal{L}_n(\boldsymbol{\theta}_J]^T, \qquad \theta_j \sim \hat{\pi} \tag{10}$$

$\hat{\mathcal{L}}_n$ represents the J-dimensional projection of $\mathcal{L}_n$ in Euclidean space. Using above approximate we can approximate inner products as dot products between vectors

$$\langle \mathcal{L}_m \mathcal{L}_n \rangle_{\hat{\pi},2} = \hat{\mathcal{L}_n}^T \hat{\mathcal{L}_m} \tag{11}$$

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

QUESTION 2

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: April 20, 2021

Local Conjugacy and Gibbs Sampling
Given

$$x_1, x_2 ... x_N \sim \mathcal{N}(x|\mu, \beta^{-1}) \tag{12}$$
$$\mu \sim \mathcal{N}(\mu|\mu_0, s_0) \tag{13}$$
$$\beta \sim Gamma(\beta|a, b) \tag{14}$$

Now finding the conditional posterior of μ i.e $P(\mu|\mathbf{X}, \beta, \mu_0, s_0)$

$$P(\mu|\mathbf{X}, \beta, \mu_0, s_0) \propto p(\mathbf{X}|\mu, \beta)p(\mu|\mu_0, s_0) \tag{15}$$
$$\propto \mathcal{N}(\mathbf{X}|\mu, \beta^{-1})\mathcal{N}(\mu|\mu_0, s_0) \tag{16}$$
$$\text{by conjugacy} \tag{17}$$
$$P(\mu|\mathbf{X}, \beta, \mu_0, s_0) = \mathcal{N}(\mu_N, s_N^{-1}) \tag{18}$$
$$\text{where} \tag{19}$$
$$\mu_N = \frac{\mu_0 + s_0\beta\sum_{n=1}^{N} x_n}{\beta N s_0 + 1} \tag{20}$$
$$s_N^{-1} = \frac{1}{s_0} + N\beta \tag{21}$$

Conditional posterior of β i.e $P(\beta|\mathbf{X}, \mu, a, b)$

$$P(\beta|\mathbf{X}, \mu, a, b) \propto p(\mathbf{X}|\mu, \beta)P(\beta|a, b) \tag{22}$$
$$\propto \mathcal{N}(\mathbf{X}|\mu, \beta^{-1})Gamma(\beta|a, b) \tag{23}$$
$$\text{by conjugacy} \tag{24}$$
$$P(\beta|x, \mu, a, b) = Gamma(a + \frac{N}{2}, b + \frac{\sum_{n=1}^{N}(x_n - \mu)^2)}{2}) \tag{25}$$

Now using Gibbs Sampling algorithm for finding the combined posterior
Steps

Result: Set of samples $(\mu^{(s)}, \beta^{(s)})_{s=1}^{S}$
Intialize an estimate of μ^0, β^0 ;
for *s=1 to T* **do**
  Draw a random sample $\beta^{(s)} \sim Gamma(a + \frac{N}{2}, b + \frac{\sum_{n=1}^{N}(x_n - \mu^{(s-1)})^2)}{2})$
  Draw a random sample $\mu^{(s)} \sim \mathcal{N}(\mu_N(s), s_N^{-1}(s))$
end

Algorithm 1: Gibbs Sampling

These S random samples $(\mu^{(s)}, \beta^{(s)})_{s=1}^{S}$ represent joint posterior $P(\mu, \beta|X)$

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

QUESTION 3

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: April 20, 2021

EM for Sparse Modeling Given:
A linear regression model $\mathbf{y} = \mathbf{X}\boldsymbol{\omega} + \boldsymbol{\epsilon}$

$$\omega_d \sim \mathcal{N}(0, \sigma^2 k_{\gamma_d}) \quad \text{where } k_{\gamma_d} = \gamma_d v_1 + (1-\gamma_d)v_0 \text{ and } v_1 > v_0 > 0 \tag{26}$$
$$\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \sigma^{\mathbf{2}}\mathbf{K}) \quad \text{where } K = \text{diag}(k_{\gamma_1}, k_{\gamma_2}...k_{\gamma_d}) \tag{27}$$
$$\theta \sim Beta(a_0, b_0) \tag{28}$$
$$\gamma_d \sim Bernoulli(\boldsymbol{\theta}) \tag{29}$$
$$\sigma^2 \sim IG(v/2, v\lambda/2) \tag{30}$$
$$\epsilon_n \sim \mathcal{N}(0, \sigma^2) \tag{31}$$
$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I_N}) \tag{32}$$

Part 1
The effect of assuming the above prior on ω induces sparcity on the weight vector $\boldsymbol{\omega}$ although the sparsity is present for only some parameters.
We have chosen two types of variances in the prior of the weights. which mean that we are taking different amount of precision on different weight parameters. This type of prior first classifies the weight paramerts into 2 categories based on their importance.
Part 2
EM Algorithm:

step 1:
Initialize the following parameters randomly and set $t = 1$

$$\gamma^{(0)}, \quad \sigma^{2(0)}, \quad \theta^{(0)} \tag{33}$$

step2:
computing the posterior over ω

$$p(\boldsymbol{\omega}^{(t)}|\mathbf{y}, \mathbf{X}, \gamma^{(t-1)}, \sigma^{2(t-1)}, \theta^{(t-1)}) \propto p(\mathbf{y}|\boldsymbol{\omega}, \mathbf{X}, \sigma^{2(t-1)}) \quad p(\boldsymbol{\omega}|\sigma^{2(t-1)}, \gamma^{(t-1)}) \tag{34}$$
$$\propto \mathcal{N}(\mathbf{y}|\mathbf{X}\boldsymbol{\omega}, \sigma^{2(t-1)}\mathbf{I_N})\mathcal{N}(\boldsymbol{\omega}|\mathbf{0}, \sigma^{\mathbf{2(t-1)}}\mathbf{K^{(t-1)}}) \tag{35}$$
$$\text{using the gaussian properties} \tag{36}$$
$$= \mathcal{N}(\boldsymbol{\omega}^{(t)}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{37}$$
$$\boldsymbol{\mu}^{(t)} = (\mathbf{K^{(t-1)^{-1}}} + \mathbf{X^T X})^{-1}\mathbf{X^T y} \tag{38}$$
$$\boldsymbol{\Sigma}^{(t)} = \sigma^{2(t)}(\mathbf{K^{(t-1)^{-1}}} + \mathbf{X^T X})^{-1} \tag{39}$$

step3:
Maximize the expected complete log likelihood

$\mathbb{E}[\mathbf{CLL}]$

$$\mathbb{E}[\log p(\boldsymbol{\omega}^{(t)}, \mathbf{y}|\gamma, \theta, \sigma^2)] = \mathbb{E}[\log p(\mathbf{y}|\boldsymbol{\omega}^{(t)}, \gamma, \sigma^2) + \log p(\boldsymbol{\omega}^{(t)}|\gamma, \sigma^2)] \tag{40}$$

$$= \mathbb{E}[\log \mathcal{N}(\mathbf{y}|\mathbf{X}\boldsymbol{\omega}^{(t)}, \sigma^2\mathbf{I_N}) + \log \mathcal{N}(\boldsymbol{\omega}^{(t)}|\mathbf{0}, \sigma^2\mathbf{K})] \tag{41}$$

$$\begin{aligned} = \mathbb{E}[&-\frac{N+D}{2}\log(2\pi\sigma^2) - \frac{\mathbf{y^T y - 2y^T X\boldsymbol{\omega}^{(t)} + \boldsymbol{\omega}^{T(t)} X^T X\boldsymbol{\omega}^{(t)}}}{2\sigma^2} \\ &- \frac{1}{2}\sum_d \log k_{\gamma_d} - \frac{\boldsymbol{\omega}^{T(t)}\mathbf{K}^{-1}\boldsymbol{\omega}^{(t)}}{2\sigma^2}] \end{aligned} \tag{42}$$

$$\begin{aligned} = &-\frac{N+D}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_d \log k_{\gamma_d} \\ &- \frac{\mathbf{y^T y - 2y^T X}\mathbb{E}[\boldsymbol{\omega}^{(t)}] + \mathbb{E}[\boldsymbol{\omega}^{(t)^T}(\mathbf{X^T X} + \mathbf{K}^{-1})\boldsymbol{\omega}^{(t)}]}{2\sigma^2} \end{aligned} \tag{43}$$

where

$$\mathbb{E}[\boldsymbol{\omega}^{(t)}] = \boldsymbol{\mu}^{(t)}$$

$$\mathbb{E}[\boldsymbol{\omega}^{(t)^T}(\mathbf{X^T X} + \mathbf{K}^{-1})\boldsymbol{\omega}^{(t)}] = Tr((\mathbf{X^T X} + \mathbf{K}^{-1})\boldsymbol{\Sigma}^{(t)}) + \boldsymbol{\mu}^{(t)^T}(\mathbf{X^T X} + \mathbf{K}^{-1})\boldsymbol{\mu}^{(t)}$$

Ref: Matrixcookbook Eq(328)

Step 4:
maximization for MAP

$$\gamma^{(t)}, \sigma^{2(t)}, \theta^{(t)} = \arg\max_{\gamma,\sigma^2,\theta} \mathbb{E}[\text{CLL}] + \log p(\gamma, \sigma^2, \theta) \tag{44}$$

$$= \arg\max_{\gamma,\sigma^2,\theta} \mathbb{E}[\text{CLL}] + \log p(\gamma|\theta) + \log p(\sigma^2) + \log p(\theta) \tag{45}$$

$$= \arg\max_{\gamma,\sigma^2,\theta} \mathbb{E}[\text{CLL}] + \sum_d \log p(\gamma_d|\theta) + \log p(\sigma^2) + \log p(\theta) \tag{46}$$

where (47)

$$\log p(\gamma_d|\theta) = \gamma_d \log\theta + (1-\gamma_d)\log(1-\theta) \tag{48}$$

$$\log p(\sigma^2) = v/2\log(v\lambda/2) - \log(\Gamma(v/2)) - (v/2+1)\log\sigma^2 - \frac{v\lambda}{2\sigma^2} \tag{49}$$

$$\log p(\theta) = (a_0-1)\log\theta + (b_0-1)\log(1-\theta) + \log(\frac{\Gamma(a_0)\Gamma(b_0)}{\Gamma(a_0+b_0)}) \tag{50}$$

Derivating wrt σ^2 and equating to 0

$$0 = -\frac{N+D}{2\sigma^2} + \frac{\mathbf{y^T y - 2y^T X}\mathbb{E}[\boldsymbol{\omega}^{(t)}] + \mathbb{E}[\boldsymbol{\omega}^{(t)^T}(\mathbf{X^T X} + \mathbf{K}^{-1})\boldsymbol{\omega}^{(t)}]}{2\sigma^4} - \frac{v+2}{2\sigma^2} + \frac{v\lambda}{2\sigma^4} \tag{51}$$

$$\sigma^{2(t)} = \frac{\mathbf{y^T y - 2y^T X}\mathbb{E}[\boldsymbol{\omega}^{(t)}] + \mathbb{E}[\boldsymbol{\omega}^{(t)^T}(\mathbf{X^T X} + \mathbf{K}^{-1})\boldsymbol{\omega}^{(t)}] + v\lambda}{N+D+v+2} \tag{52}$$

As the $\gamma_d \in 0, 1$ we can directly evaluate it. $\arg\max_{\gamma_d}$ can be written (having the terms only related to γ_d) as follows:

$$\gamma_d^{(t)} = \arg \max_{\gamma_d \in \{0,1\}} \left\{ -\frac{1}{2} \log k_{\gamma_d} - \frac{\mathbb{E}[\boldsymbol{\omega}^{(t)^T} (\mathbf{X^T X} + \mathbf{K^{-1}}) \boldsymbol{\omega}^{(t)}]}{2\sigma^{2(t)}} \right. \tag{53}$$

$$\left. + \gamma_d \log \theta^{(t-1)} + (1 - \gamma_d) \log(1 - \theta^{(t-1)}) \right\} \tag{54}$$

We can directly subtitute the values of $\gamma_d = 0, 1$ and then find out the value for which is has the above equation has max value
Derivating wrt θ and equating to 0

$$0 = \frac{\sum_d \gamma_d^{(t)}}{\theta} - \frac{D - \sum_d \gamma_d^{(t)}}{1 - \theta} + \frac{a_0 - 1}{\theta} - \frac{b_0 - 1}{1 - \theta} \tag{55}$$

$$\theta^{(t)} = \frac{\sum_d \gamma_d^{(t)} + a_0 - 1}{D + a_0 + b_0 - 2} \tag{56}$$

Step5 :
if not converged set $t = t + 1$ and go to step2
else return $\gamma^{(t)}, \sigma^{2(t)}, \theta^{(t)}$ and $p(\boldsymbol{\omega}^{(t)} | \mathbf{y}, \mathbf{X}, \gamma^{(t-1)}, \sigma^{2(t-1)}, \theta^{(t-1)})$

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

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: April 20, 2021

QUESTION

4

Part 1: GP Posterior

Given

$$\text{likelihood}: \quad p(\mathbf{y}|\mathbf{f}) \quad : \quad \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I_N) \tag{57}$$

$$\text{prior}: \quad p(\mathbf{f}|\mathbf{X}) \quad : \quad \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \tag{58}$$

$$\text{where} \quad \mathbf{K}_{ij} = k(x_i, x_j) \tag{59}$$

By using the gaussian properties from the lecture 4.1
The posterior

$$p(\mathbf{f}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X}) \tag{60}$$

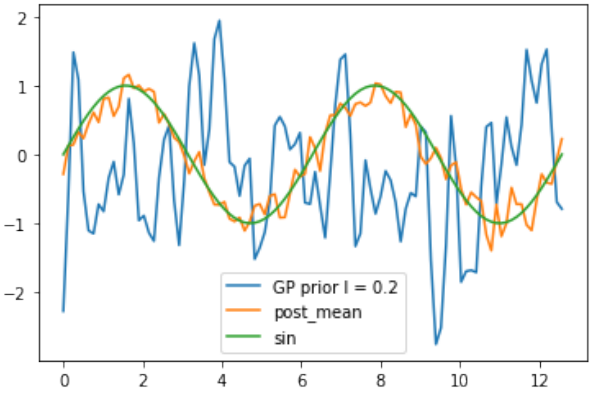
$$= \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N) \tag{61}$$

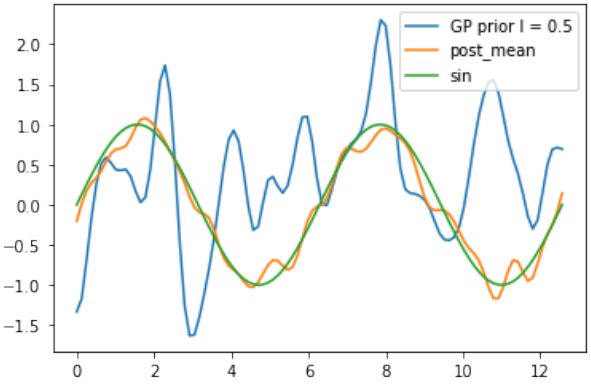
$$\text{where} \tag{62}$$

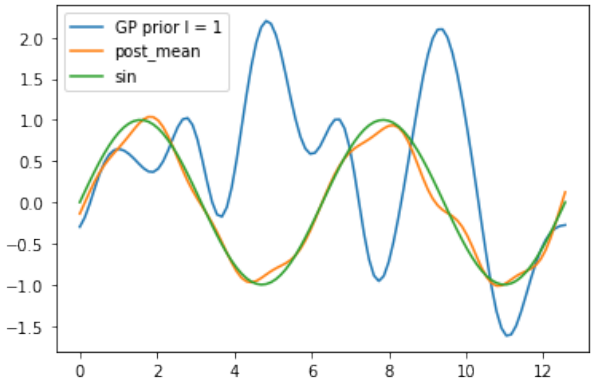
$$\boldsymbol{\mu}_N = (\sigma^2 \mathbf{K^{-1}} + \mathbf{I_N})^{-1}\mathbf{y} \tag{63}$$

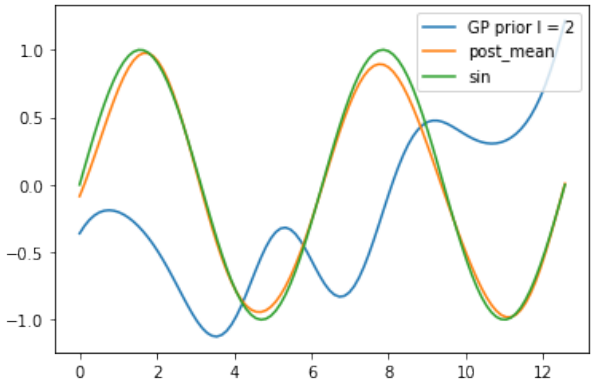
$$\boldsymbol{\Sigma}_N = (\mathbf{K^{-1}} + \sigma^{2^{-1}}\mathbf{I}_N)^{-1} \tag{64}$$

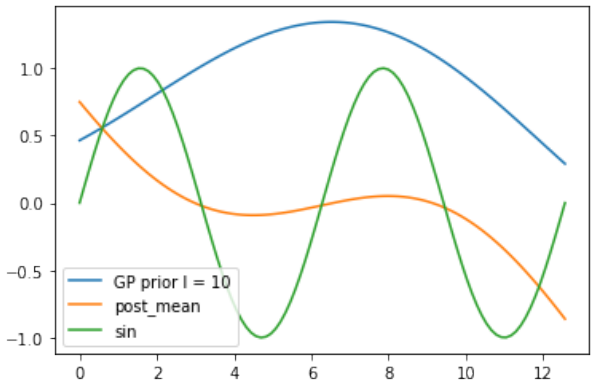
Part 2: Visualizing GP Priors and Posteriors for Regression











l signifies how close x and $x\prime$ should be to influence each other significantly. Lower l values create more wiggles in function with wide uncertainty between training data points.
As the l value increases we can see that prior function of the GP becomes smoother. Where as the posterior mean tries to overfit the data for small values of l but later it gets closer to the original sin function but after larger values of of l it also becomes oversmooth and underfits the sin curve

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

QUESTION 5

Student Name: Musale Krushna Pavan
Roll Number: 20111268
Date: April 20, 2021

Speeding up Gaussian Processes
Part 1 Given:

$$\text{training data:} \tag{65}$$
$$(\mathbf{X}, \mathbf{f}) = \{x_n, f_n\}_{n=1}^{N} \tag{66}$$
$$y_n = f(\mathbf{x}_n) = f_n \tag{67}$$
$$\mathbf{f} \sim \mathcal{GP}(0, k) \tag{68}$$
$$\text{posterior predictive distribution} \tag{69}$$
$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{f}) = \mathcal{N}(f_*|\mathbf{k}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}\mathbf{f}, k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}\mathbf{k}_*) \tag{70}$$

Let's reduce this cost to make GPs more scalable. To do this, suppose there are

$$\text{pseudo training inputs} \tag{71}$$
$$\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_M\} \qquad M << N \tag{72}$$
$$\text{pseudo training outputs} \tag{73}$$
$$\mathbf{t} = \{t_1, ..., t_M\} \tag{74}$$
$$\text{modeled by} \tag{75}$$
$$t_m = f(\mathbf{z}_m) \tag{76}$$

Note: that $(\mathbf{Z}, \mathbf{t})$ are NOT known
Let likelihood for each training output f_n with PPD as follows

$$p(f_*|\mathbf{x}_*, \mathbf{Z}, \mathbf{t}) = \mathcal{N}(f_*|\tilde{\mathbf{k}}_*^{\mathbf{T}}\tilde{\mathbf{K}}^{-\mathbf{1}}\mathbf{t}, k(\mathbf{x}_*, \mathbf{x}_*) - \tilde{\mathbf{k}}_*^{\mathbf{T}}\tilde{\mathbf{K}}^{-\mathbf{1}}\tilde{\mathbf{k}}_*) \tag{77}$$

The likelihood can be shown as

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Z}, \mathbf{t}) = \prod_{n=1}^{N} p(f_n|\mathbf{x}_n, \mathbf{Z}, \mathbf{f}) \tag{78}$$
$$= N(\mathbf{f}|\mathbf{K}_{NM}\tilde{\mathbf{K}}^{-\mathbf{1}}\mathbf{t}, \Lambda) \tag{79}$$
$$\text{where } \Lambda \text{ is a diagonal matrix with} \tag{80}$$
$$\Lambda_{*,*} = k(\mathbf{x}_*, \mathbf{x}_*) - \tilde{\mathbf{k}}_*^{\mathbf{T}}\tilde{\mathbf{K}}^{-\mathbf{1}}\tilde{\mathbf{k}}_* \text{ and } [\mathbf{K}_{NM}]_{nm} = k(x_n, x_m) \tag{81}$$

finding the prior over pseudo data

$$p(\mathbf{t}|\mathbf{Z}) = \mathcal{N}(\mathbf{t}|\mathbf{0}, \tilde{\mathbf{K}}) \tag{82}$$

Here, we assume the pseudo-inputs Z are known. .finding the prior over pseudo data

$$p(\mathbf{t}|\mathbf{Z},\mathbf{X},\mathbf{f}) \propto p(\mathbf{f}|\mathbf{X},\mathbf{Z},\mathbf{t})p(\mathbf{t}|\mathbf{Z}) \tag{83}$$

$$\text{using gaussian properties} \tag{84}$$

$$= \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{85}$$

$$\text{where} \tag{86}$$

$$\boldsymbol{\Sigma}_t = (\tilde{\mathbf{K}}^{-\mathbf{1}} + (\mathbf{K}_{NM}\tilde{\mathbf{K}}^{-\mathbf{1}})^T\Lambda^{-1}\mathbf{K}_{NM}\tilde{\mathbf{K}}^{-\mathbf{1}})^{-1} \tag{87}$$

$$\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t\tilde{\mathbf{K}}^{-\mathbf{1}}\mathbf{K}_{NM}^T\Lambda^{-1}\mathbf{f} \tag{88}$$

expression of the posterior predictive distribution for the output y_* of a new input x_*

$$p(y_*|\mathbf{x}_*,\mathbf{X},\mathbf{f},\mathbf{Z}) = \int p(y_*|\mathbf{x}_*,\mathbf{t},\mathbf{Z})p(\mathbf{t}|\mathbf{Z},\mathbf{X},\mathbf{f})d\mathbf{t} \tag{89}$$

$$= \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \tag{90}$$

$$\text{where} \tag{91}$$

$$\boldsymbol{\mu}_* = \tilde{\mathbf{k}}_*^{\mathbf{T}}\tilde{\mathbf{K}}^{-\mathbf{1}}\boldsymbol{\mu}_t \tag{92}$$

$$\boldsymbol{\Sigma}_* = \tilde{\mathbf{k}}_*^{\mathbf{T}}\tilde{\mathbf{K}}^{-\mathbf{1}}\boldsymbol{\Sigma}_t\tilde{\mathbf{K}}^{-\mathbf{1}}\tilde{\mathbf{k}}_* + k(\mathbf{x}_*,\mathbf{x}_*) - \tilde{\mathbf{k}}_*^{\mathbf{T}}\tilde{\mathbf{K}}^{-\mathbf{1}}\tilde{\mathbf{k}}_* \tag{93}$$

Time Complexity:
We can see from the above equation that calculation $\boldsymbol{\mu}_*$ takes the most time
in which $\mathbf{K_{NM}}^T\Lambda^{-1}K_{NM}$ takes $\mathcal{O}(M^2N)$. Through the $\tilde{\mathbf{K}}_{-1}$ and some other calculations takes $\mathcal{O}(M^3)$ its less than $\mathcal{O}(M^2N)$ as $M << N$.
So the time complexity using this pseudo training data is $\mathcal{O}(M^2N)$.

Part 2
Marginal Likelihood

$$p(\mathbf{f}|\mathbf{X},\mathbf{Z}) = \int p(\mathbf{f}|\mathbf{X},\mathbf{Z},\mathbf{t})p(\mathbf{t}|\mathbf{Z})d\mathbf{t} \tag{94}$$

$$= \mathcal{N}(\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f) \tag{95}$$

$$\text{where} \tag{96}$$

$$\boldsymbol{\mu}_f = 0 \tag{97}$$

$$\boldsymbol{\Sigma}_f = \mathbf{K}_{NM}\tilde{\mathbf{K}}^{-\mathbf{1}}\mathbf{K}_{NM}^T + \boldsymbol{\Lambda} \tag{98}$$

Now we can use MLE-II for getting the values $\mathbf{Z}$

$$\tilde{\mathbf{Z}} = \arg\max_{\mathbf{Z}} \log p(\mathbf{f}|\mathbf{X},\mathbf{Z}) \tag{99}$$

$$= \arg\min_{\mathbf{Z}} (\log|\boldsymbol{\Sigma}_f| + \mathbf{f}^T\boldsymbol{\Sigma}_f^{-1}\mathbf{f}) \tag{100}$$

The above objective function can be solved using gradient ascent.