Topics in Probabilistic Modeling & Inference (CS698X), Spring 2021
Indian Institute of Technology Kanpur
Homework Assignment Number 2

QUESTION 1

Student Name: Ajita Shree
Roll Number: 20111262
Date: April 23, 2021

Solution: Batch Bayesian Active Learning

- **a) Equation 4: Expected Complete data log posterior What does this quantity mean and how is it used to select a batch D of inputs whose labels should be queried from the oracle. How the sparse approximation based objective function will accomplish this along with a brief intuitive meaning of this objective function (100 words)?**?

  $E_{y_p}[logp(\theta/D_0 \cup (X_p, Y_p))] = E_{y_p} log(p(\theta/D_0) + \sum_{m \in M} E_{y_m}[log(y_m/x_m, \theta)] + H[y_m/x_m, D_0])$

  Expected Complete data log-posterior can be denoted by the above equation. The posterior is calculated based on initial data, D_0 and new set $\{X_p, Y_p\}$. The new batch is chosen such that updated log posterior $log(\theta/D_0 U D')$ best approximates the complete data log posterior $log(\theta/D_0 U D_p)$. This batch approach is better as it is infeasible to re-train the model after every acquired data point and also addition of a single point to training data does not have significant effect on the posterior. Only 2nd term is important as it is dependent on new data X_p, Y_p
  The batch construction is posed as sparse subset approximation in the following ways.

  - Goal is to choose the batch that will best approximates $\sum_m L_m(\theta) (i.e. E_{y_m}[log((y_m/_m, \theta)) + H(y_m/x_m, D_0)])$.
  - Suppose $w \in 0, 1^M$ indicate which points to include in the AL batch, then $L(w) = \sum_m w_m L_m$.
  - Then, Sparse subset approximation will be as follows:
  - $w^* = minimize_w ||L - L(w)||^2$ subject to $w_m \in 0, 1$ and $\sum_m 1_m <= b$
  - if we pick the right set, the true posterior will be approximately equal to the expected posterior, but solving this NP-hard problem and hence, intractable.So, other approaches needs to be followed.
  -

- **b) Sparse approximation based objective is difficult to optimize, what relaxed objective is minimized instead in paper (200 words)?**

  As objective is intractable, following approaches are used to get the approximate solutions.

  Inner Products : Batch construction to be done in a Hilbert space induced by an inner product $< L_n, L_m >$ between function vectors, with associated norm. The choice adds the notion of directionality, thus accounting for similarity between selected points. There are two types of inner products that can be explored.

- **Weighted Inner Fisher Product** $< L_n, L_m >_\pi = E_\pi[\nabla_\theta L_n(\theta)^T \nabla_\theta L_m(\theta)]$ - it requires taking gradient of expected log-likelihood terms.
- **Weighted Euclidean inner product**

$$< L_n, L_m >_\pi = E_\pi[L_n(\theta) L_m(\theta)]$$

  - it requires marginal likelihood of data points. The advantage is that it only requires tractable likelihood computations.

Frank Wolfe Optimization

- Another approach is to relax the binary weight constraint to be non-negative and replace the cardinality constraint with a polytope constraint
- Assuming $\sigma_m = ||L_m||, \sigma = \sum_m \sigma_m$ and $K_m n =< L_m, L_n >$
- The objective will be as follows.
- $minimize_w (1-w)^T K (1-w)$ s.t. $w_m >= 0 \; \forall m$, $\sum_m w_m \sigma_m = \sigma$
- Original objective $||L - L(w)||^2$ got converted to $(1-w)^T K (1-w)$
- The new objective can be easily solved by Frank-Wolfe algorithm, yielding optimal weights after b iterations.
- Greedily select L_f point (most aligned with the error $L - L(w)$)based on
- $argmax_n < L - L(w), L(n)/\sigma_n >= 1/\sigma_n sum_{m \in N}(1 - w_m) < L_m, L_n >$
- Weights are updated based on line search along the fth vertex of the polytope
- The algorithm runs for b iterations, in each iterations maximum one point can be selected and algo can reselect indices from previous iterations. Hencem resulting weight vector has <= b non-zero entries.
- Final step is to project weights back to the feasible space.

- **c) For what kind of supervised learning models, the acquisition function proposed in the paper has a closed form expression? For other type of models, where the acquisition function won't be available, what scheme does the paper propose to get closed form expressions?**

  Above approach can be used to derive closed form expression for following models.

  - **1) Bayesian Linear Regression** Fisher Inner product for linear regression setting (has a closed form)can be written as $< L_n, L_m >_{\pi,F} = (x_n^T x_m / \sigma_0^4) x_n^T \Sigma_\theta x_m$
  - In above equation, π is chosen to be posterior $p(\theta / D_0, \sigma_0^2)$
  - **2) Probit Regression** Here, Fisher inner product, also has a closed form, can be written as $< L_n, L_m >_{\pi,F} = x_n^T x_m (bivariate{-}normalCDF(\zeta_n, \zeta_m, \rho_{n,m}) - \phi(\zeta_n)\phi(\zeta_m))$
  - In above equation, ζ and ρ has a closed form.
  - **3) Alternate scheme for closed form solutions**
  - Compute random feature projections for the weighted fisher inner product
  - The projection will have the form $L_n^{'} = 1/\sqrt{J}[L_n(\theta_1).....L_n(\theta_J)]^T, \theta_j \in \pi$
  - L_n' denotes the J-dimensional projection of L_n in Euclidean space. Given this projection, inner product can be approximated as $< L_n, L_m >= L_n^{T\prime} L_m'$, where $L_n^{T\prime} L_m'$ using **J monte-carlo samples** from the posterior π

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2021
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Ajita Shree
Roll Number: 20111262
Date: April 23, 2021

QUESTION 2

Solution

Goal Derive conditional posterior of μ, β and describe how it can be used to compute their joint posterior.

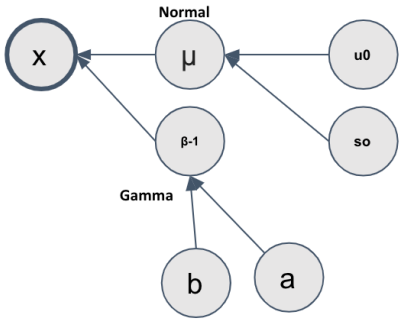


Figure 1: Graphical model diagram

- The joint probability of all the variables in the given model can be written as
- $p(X, \mu, \beta^{-1}/\mu_o, s_o, b, a) = \prod_{n \in N} N(x_n/\mu, \beta^{-1}) * N(\mu/mu_o, s_o) * Gamma(\beta/a, b)$
- The conditional probability of parameters (μ, β) can be written as follows
- CP of $\mu, p(\mu_t/X, \mu_o, s_o, a, b, \beta_{t-1}) = \prod_{n \in N} N(x_n/\mu, \beta^{-1}) * N(\mu/\mu_o, s_o)$
  - Using Gaussian property, CP of $\mu = N(\mu/\mu_N, \Sigma_N)$
  - where variance, $\Sigma_N = s_o^{-1} + N\beta$ and $\mu_N = \Sigma_N(s_o^{-1} * \mu_o + \beta \sum_{n \in N} x_n)$
- CP of $\beta, p(\beta_t/X, \mu_o, s_o, a, b, \mu_t) = \prod_{n \in N} N(x_n/\mu, \beta^{-1}) * Gamma(\beta/a, b)$
  - Using Gaussian property, CP of $\beta = Gamma(\beta/a + 0.5 * N, b + 0.5 * \sum_{n \in N}(x_n - \mu)^2$

Gibbs Sampler will work as follows:

- Step 1: Initialize μ_0, β_0
- Step 2: Loop t = 1, 2, 3....T
  - Draw samples for μ from $p(\mu_t/X, \mu_o, s_o, a, b, \beta_{t-1})$
  - Draw samples for β from $p(\beta_t/X, \mu_o, s_o, a, b, \mu_t)$
- Step 3: T samples represent the joint posterior $p(\mu, \beta/X)$

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2021
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Ajita Shree
Roll Number: 20111262
Date: April 23, 2021

QUESTION 3

EM for Sparse Modeling

Consider a linear regression model y = Xw + ϵ with y = $[y_1, ..., y_N]^T$ is the N × 1 response vector, X is the N×D feature-matrix,and $\epsilon = [\epsilon_1, ..., \epsilon_N]^T$ is the N×1 vector of i.i.d.Gaussian noise N(0,σ^2).The priors p(γ_d) = Bernoulli(),p(θ) = Beta(a0, b0), and p(σ^2) = IG(v/2, $v\lambda/2$), prior on w$_d$p(w$_d|\sigma, \gamma_d$) = $N(0, \sigma^2 * \kappa_{\gamma_d})$ with $\kappa_{\gamma_d} = \gamma_d * v1 + (1 - \gamma_d)v0$ where $v1 > v0 > 0$.

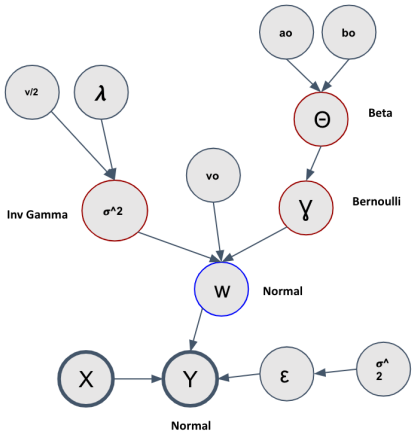


Figure 2: Graphical model diagram

a) Effect of w prior:The effect of assuming the above prior in w is that it will be able to model sparsity really well. Depending on the random variable, the distribution will have more variance or less variance (v1 > v2). When ever distribution has low variance, it will be high peaked Gaussian and most of the samples would be near zero ensuring sparsity.

b) Design EM Algorithm

- Joint probability distribution, $P(Y, w, \gamma, \sigma^2, \theta/X)$
- Let K_D be the matrix = $\sigma^2 * [k_{\gamma 1}, k_{\gamma 2}...k_{\gamma d}]$
- $N(Y/XW, \sigma^2 I_N) * N(w/0, K_D) * IG(\sigma^2/v/2, v\lambda/2) * Bernoulli(\gamma_d/\theta) * Beta(\theta/a_0, b_0)$
- **E step: Computation of CP of w**
- $p(w/Y, \gamma, \sigma^2, \theta, X) = N(w/0, \sigma^2 K_{\gamma_d}) * N(Y/Xw, \sigma^2 I_N)$
- Using Gaussian property, the CP of w will be Gaussian $N(w/\mu_N, \Sigma_N)$ where
- $\Sigma_N = (\sigma^2)^{-1}(K_D^{-1} + X^T X)^{-1}$

- $\mu_N = (K_D^{-1} + X^T X)^{-1} X^T Y$
- **M step: Computation of γ, σ^2, θ by maximising the complete data log likelihood**
- $argmax_{\gamma,\sigma^2,\theta}\ E_{p(w/y,\Theta,X)} logp(y, w/\Theta)$ where $\Theta = \gamma, \sigma^2, \theta$
- $E_{p(w/\Theta)} log(N(Y/Xw, \sigma^{2-1} * I_N) + log(N(w/0, K_D)$
- Expanding the Gaussian terms, we will get
- $\propto E_{p(w/\Theta)}\ log(exp((-1/2)(y - Xw)^T(\sigma^2 I_n)^{-1}(y - Xw))) + log(exp((-1/2)w^T K_D^{-1} w)) + log(|\sigma^2 I_n|^{-1/2}) + log(|K_D|^{-1/2})$
- $\propto (-1/2) * E_{p(w/\Theta)}[(y - Xw)^T(y - Xw)] E_{p(w/\Theta)}(w^T K_D^{-1} w) + log(|\sigma^2 I_n|^{-1/2}) + log(|K_D|^{-1/2})$
- Using the identity, $E[(Ax + a)^T(Bx + b)] = Tr(AMB^T) + (Am + a)^T(Bm + b)$ where m and M are mean and covariance of rv x. and identity $E[x^T Ax] = Tr(AM) + m^T Am$, the expection of CLL can be simplified as follows
- **maximizing exp-CLL** $= argmax_{\gamma,\sigma^2,\theta}\ -1/2\epsilon * Tr(X\Sigma_N X^T) + (y - X\mu_N)^T(y - X_m u_N) + Tr(K_D^{-1}\Sigma_N) + \mu_N^T K_D^{-1}\mu$
- **Maximization of the MAP**
- $\gamma_d^{'} \propto argmax_\gamma$ exp-CLL + log $(\theta^{\gamma_d}(1 - \theta)^{1-\gamma_d})$: As γ_d can take value either 0 or 1, it will be easy to evaluate without taking gradient.
- $\sigma^{2\prime} \propto argmax_\sigma^2$ exp-CLL + log $(\sigma^{2(-v/2-1)} exp(-v/2\sigma^2))$: Taking derivative w.r.t. σ^2 and equate it to 0 will give σ^2 value.
- $\theta' \propto \arg\max_\theta\ exp - CLL + log(\theta^{a_0-1}(1 - \theta)^{b_0-1})$ Take derivative, we will have $\theta = (a_0 - 1/a_0 - b_0)$
- differentiating the respective equations and equating it to 0, we will get the following values for the parameters.

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2021
Indian Institute of Technology Kanpur
Homework Assignment Number 2

Student Name: Ajita Shree
Roll Number: 20111262
Date: April 23, 2021

QUESTION

4

Part a: GP Posterior

Assume a zero mean GP prior p(f) = GP(0, κ) which, from the GP definition, is equivalent to p(f) = N (0, K) where $f = [f(x_1,), .., f(x_N)]^T$ is an N × 1 vector and K is the N × N kernel matrix with $K_{nm} = \kappa(x_n, x_m)$. Assume a likelihood model $p(y_n|x_n, f) = N(y_n|f(x_n), \sigma^2)$, where $f = GP(0, \kappa)$. To Derive the expression for the GP posterior, i.e., p(f/y).

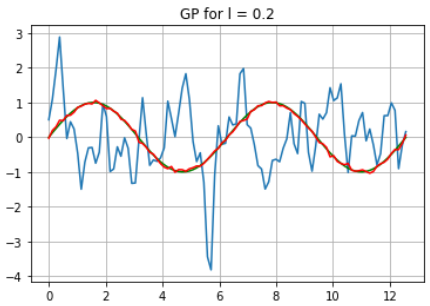
Solution

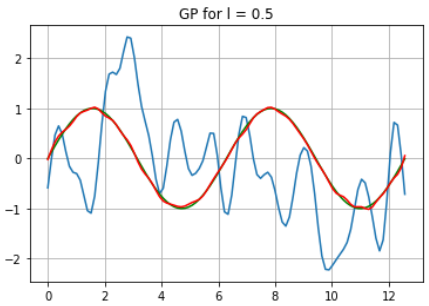
- Likelihood, $p(y_n/x_n, f) = N(y_n/f(x_n), \sigma^2)$, where f is $GP(0, \kappa)$
- Prior, $p(f) = N(f/0, K)$
- Posterior, $p(f/y) = p(f)p(y/f)$
  1. $= \prod_{n \in N} N(y_n/f_n, \sigma^2) * N(f/0, K)$
  2. Using Gaussian property, posterior can be written as $N(f/\mu_N, \Sigma_N)$
  3. where $\Sigma_N = (K^{-1} + (\sigma^2)^{-1} * I_N)^{-1}$
  4. and $\mu_N = (\sigma^2 K^{-1} + I_N)^{-1} Y$

Part b: Visualizing GP Priors and Posteriors for Regression

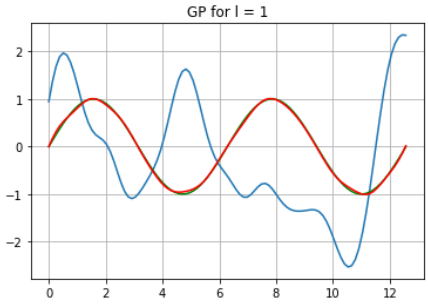
Assume a GP prior GP(0, κ) where $\kappa(x, x') = \rho * exp(-1 * (x - x')^2/l^2)$. scalar inputs using the model $y = sin(x) + \epsilon_n$ with $\epsilon N(0, \sigma^2)$ where $\sigma^2 = 0.05$. For each of the following 5 values of l from [0.2, 0.5, 1, 2, 10], plot shows following.

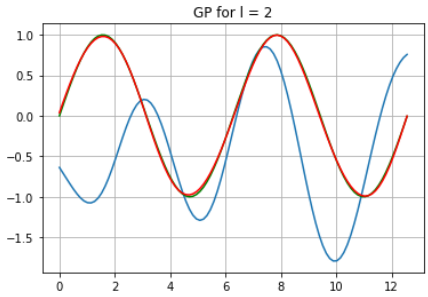
- Random sample from the GP prior p(f) = N (0, K) − > Blue curve
- Plot of the mean of the GP posterior − > Red curve
- True function, sin(x) − > Green curve



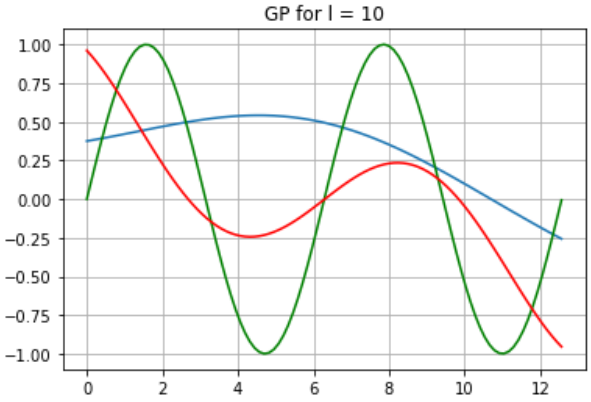


l = 0.5





l = 2



Conclusion

It can concluded from the above plots that when l value is small i.e. 0.2 or 0.5, there is a over-fitting in data-set. For l = 1, 2, the curve seems to fit the true function value. As l increases further more, it is under-fitting the data. The same logic goes for posterior mean, it is completely coinciding with the data in the plots where l is very small. As l increases, it is not fitting that well with the true function sin(x).

Topics in Probabilistic Modeling & Inference (CS698X), Spring 2021
Indian Institute of Technology Kanpur
Homework Assignment Number 2

QUESTION 5

Student Name: Ajita Shree
Roll Number: 20111262
Date: April 23, 2021

Speeding up Gaussian Processes

Consider Gaussian Process (GP) regression where $y_n = f(x_n) + \epsilon_n$ with f modeled by $GP(0, \kappa)$ where GP mean function is 0 and kernel/covariance function is κ, and noise $\epsilon_n \in$ N (0, σ^2). For simplicity, we will assume the noiseless setting, so $y_n = f(x_n) = f_n$. Given N training inputs $(X, f) = \{x_n, f_n\}_{n \in N}1$, The posterior predictive distribution for a new input x_* is

$$p(f_*/x_*, X, f) = N(f_*/k_*^T K^{-1} f, \kappa(x_*, x_*) - k_*^T K^{-1} k_*)$$

In the above, K is the N $\times$ N kernel matrix of training inputs and k_* is N $\times$ 1 vector of kernel based similarities of x_* with each of the training inputs. As we know, the above has $O(N^3)$ cost due to N $\times$ N matrix inversion. To reduce the cost, suppose a set of pseudo training inputs Z = z1,...,zM with M N, with their noiseless pseudo outputs t = t1,...,tM modeled by the same GP, i.e., $t_m = f(z_m)$. The likelihood for fn to be modeled by a posterior predictive having the same form as the GP regression's posterior predictive but with (Z, t).

$p(f_n/x_n, Z, t) = N(f_n/k_*'^T K_M^{-1} t, \kappa(x_n, x_n) - k_n'^T K_M^{-1} k_n')$ Here, K_M is M*M kernel matrix of pseudo inputs Z and k_n' is kernel based similarity of x_n with each of z1, z2, ...zm.

Solution: Part a

- **Derive** $p(y_*/x_*, X, f, Z)$
- As given in ques, likelihood, $p(f_n/x_n, Z, t) = N(k_n'^T K_M^{-1} t, \kappa(x_n, x_n) - k_n'^T K_M^{-1} k_n')$
- $p(f/X, Z, t) = \prod_{n \in N} p(f_n/x_n, Z, t) = N(f/K_Z, K_M^{-1} t, \Sigma_Z)$
- where $K_Z = Z * M$ kappa matrix $\kappa(x_n, z_m)$ and $K_M = M * M$ kappa matrix $\kappa(z_n, z_m)$
- and Σ_Z = Covariance matrix where nth, mth entry is $\kappa(x_n, x_m) - k_n^T K_m^{-1} k_n$
- **Eqn 1: PPD** $p(y_*/x_*, X, f, Z) = \int p(y_*/x_*, X, f, Z, t) p(t/X, f, Z) dt$
- $p(t/X, f, Z) = p(f/X, Z, t) * p(t/Z) = N(K_Z K_M^{-1} t, \Sigma_Z) * N(t/0, K_M)$
- Using Gaussian Property, above posterior can be written as
- N(t/μ_T, Σ_T) where
- $\mu_T = \Sigma_T K_M^{-1} K_Z^T \Sigma_Z^{-1} f$
- $\Sigma_T = (K_M^{-1} + K_M^{-1} K_Z^T \Sigma_Z^{-1} K_Z K_M^{-1})^{-1}$
- We know $p(f_*/f) = N(k_*^T K_M^{-1} t, \kappa(x_*, x_*) - k_*^T K_M^{-1} k_*)$
- Now, plugging $p(f_*/f)$ and $N(t/\mu_T, \Sigma_T)$ values in equation 1, we will have

- $PPD = \int p(f_*/f, X, Z, t) * N(t/\mu_T, \Sigma_T)$
- $PPD = \int N(k_*^T K_M^{-1} t, \kappa(x_*, x_*) - k_*^T K_M^{-1} k_*) * N(t/\mu_T, \Sigma_T)$
- PPD = $N(f_*/\mu_*, \Sigma_*)$, where
- $\mu_* = k_*^T K_m^{-1} \mu_T$
- $\Sigma_* = \kappa(x_*, x_*) - k_*^T K_m^{-1} k_* + k_*^T K_m^{-1} \Sigma_T K_m^{-1} k_*$

Solution: Part b

- $p(f/X, Z) = \int p(f/X, Z, t) p(t/Z) dt$
- $\int N(f/K_Z * K_M^{-1} t, \Sigma_Z) N(t/0, K_M) dt$
- Using Gaussian integration property, we will have
- $= N(f/\mu_F, \Sigma_F)$, where
- $\mu_F = K_Z K_M^{-1} * 0 = 0$ (mean value of t = 0)
- $\Sigma_F = \Sigma_Z + K_Z * K_M^{-1} * K_M * K_M^{-1} * K_Z^T = \Sigma_Z + K_Z * K_M^{-1} * K_Z^T$
- **For solving Z, MLE-2, we have following**
- $Z' = argmax_Z p(f/X, Z)$
- $argmax_Z \ log(det(\Sigma_F)^{(-1/2)} exp(-1/2(f^T \Sigma_F^{-1} f)))$
- $argmax_Z - 1/2 log(|\Sigma_F|) - 1/2(f^T \Sigma_F^T f)$
- The above objective function can be solved using gradient methods