Probabilistic Machine Learning
(CS772A, Spring 2023)
Homework 2
Due Date: March 30, 2023 (11:59pm)

Instructions:

- Only electronic submissions will be accepted. Your main PDF writeup must be typeset in LaTeX (please also refer to the "Additional Instructions" below).
- Your submission will have two parts: The main PDF writeup (to be submitted via Gradescope `https://www.gradescope.com/`) and the code for the programming part (to be submitted via this Dropbox link: `https://tinyurl.com/y44mmfjn`). Both parts must be submitted by the deadline to receive full credit (**delay in submitting either part would incur late penalty for both parts**). We will be accepting late submissions upto 72 hours after the deadline (with every 24 hours delay incurring a 10% late penalty, applied on per-hour delay basis). We won't be able to accept submissions after that.
- We have created your Gradescope account (you should have received the notification). Please use your IITK CC ID (not any other email ID) to login. Use the "Forgot Password" option to set your password.

Additional Instructions

- We have provided a LaTeX template file `hw2sol.tex` to help typeset your PDF writeup. There is also a style file `pmi.sty` that contain shortcuts to many of the useful LaTeX commends for doing things such as boldfaced/calligraphic fonts for letters, various mathematical/greek symbols, etc., and others. Use of these shortcuts is recommended (but not necessary).
- Your answer to every question should begin on a new page. The provided template is designed to do this automatically. However, if it fails to do so, use the `\clearpage` option in LaTeX before starting the answer to a new question, to *enforce* this.
- While submitting your assignment on the Gradescope website, you will have to specify on which page(s) is question 1 answered, on which page(s) is question 2 answered etc. To do this properly, first ensure that the answer to each question starts on a different page.
- Be careful to flush all your floats (figures, tables) corresponding to question n before starting the answer to question $n+1$ otherwise, while grading, we might miss your important parts of your answers.
- Your solutions must appear in proper order in the PDF file i.e. solution to question n must be complete in the PDF file (including all plots, tables, proofs etc) before you present a solution to question $n+1$.
- For the programming part, all the code and README should be zipped together and submitted as a single file named `yourrollnumber.zip`. Please DO NOT submit the data provided.

Problem 1 (15 marks)

(Gaussian-ify the Gamma!) The gamma p.d.f. is defined as $\text{Gamma}(x|a,b) = \frac{b^a}{\Gamma(a)}x^{a-1}e^{-bx}$, where a and b are the shape and rate parameters, respectively, and $\Gamma(a)$ denotes the gamma function.

- Approximate this gamma distribution by a Gaussian using Laplace approximation. How does the Laplace approximation compare with approximating $\text{Gamma}(x|a,b)$ by a Gaussian whose mean and variance are equal to the mean and variance, respectively, of $\text{Gamma}(x|a,b)$. Under what condition would these two approximations be roughly the same?
- Using the Laplace approximation, approximate the gamma function $\Gamma(a)$ which appears in the normalization constant of $\text{Gamma}(x|a,b)$. Plot this approximation as well as the the approximation of $\Gamma(a)$ computed using some standard library function, as function of a, and comment on the accuracy of the Laplace approximation.

Problem 2 (15 marks)

(Local Conjugacy and Gibbs Sampling) Suppose we have N scalar observations $x_1, x_2, \ldots, x_N$ drawn i.i.d. from a Gaussian $\mathcal{N}(x|\mu, \beta^{-1})$. The mean μ has a Gaussian prior $\mathcal{N}(\mu|\mu_0, s_0)$ where μ_0 is the mean and s_0 is the variance of the Gaussian prior. The precision β has a gamma prior $\text{Gamma}(\beta|a,b)$ where a and b are the shape and rate parameters, respectively, of the gamma prior. Derive the conditional posteriors for μ and β using the idea of local conjugacy.

Also describe how you will use these conditional posteriors in a Gibbs sampling algorithm to approximate the joint posterior μ and β. Sketch all the steps of this Gibbs sampler.

Problem 3 (30 marks)

(EM for Hyperparameter Estimation) In the class, we looked at the MLE-II approach for point estimation of hyperparameter for the probabilistic linear regression model. The MLE-II approach to learning the hyperparameters λ, β for this model reduces to doing MLE on marginal likelihood $p(\boldsymbol{y}|\mathbf{X}, \lambda, \beta) = \int p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X}, \lambda, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$.

Another approach would be to use the Expectation Maximization (EM) algorithm where instead of doing MLE on (log)$p(\boldsymbol{y}|\mathbf{X}, \lambda, \beta)$, we estimate λ, β by doing MLE on the expected complete data log-likelihood defined as $\mathbb{E}[\log p(\boldsymbol{y}, \boldsymbol{w}|\mathbf{X}, \lambda, \beta)]$, where the expectation is w.r.t. $\boldsymbol{w}$ which is treated here as the latent variable.

Assuming $p(y_n|\boldsymbol{w}, \boldsymbol{x}_n) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1})$ and $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \lambda^{-1}\mathbf{I})$, sketch the EM algorithm, clearly specifying the E and M steps. In particular,

- What would be estimated in the E step (has to be a distribution)? Give the complete expression for that.
- Derive the M step updates for λ and β (note: some calculations may require you to make use of properties of matrix trace). Clearly specify the required expectations you will need to compute for these updates and give their expressions.

Problem 4 (30 marks)

(EM for a Binary Classification Model) Consider binary classification with training data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \{0, 1\}$. Assume each binary label to be generated as $y_n = \mathbb{I}[z_n > 0]$, where z_n is a Gaussian latent variable with $p(z_n|\boldsymbol{w}, \boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n, 1)$, $\boldsymbol{w} \in \mathbb{R}^D$, and $\mathbb{I}[.]$ returns 1 if the condition is true, and 0 otherwise. The model parameter here is $\boldsymbol{w}$ and your goal here to do point estimation for $\boldsymbol{w}$.

Develop an EM algorithm to do this. In particular, write down the expected CLL expression, and derive the expressions for all the relevant quantities that are estimated in the E step (i.e., posteriors for latent variables,

the expectations you would need in the M step, etc.) and the M step update equations (MLE expression for $\boldsymbol{w}$). Sketch the overall EM algorithm as well.

Note: In deriving the EM algorithm for this model, you would most likely need to compute the expectation of a truncated normal distribution. You may visit the Wikipedia entry to look up the desired expression.

Problem 5: Gaussian Processes (30 marks)

Part 1: GP Posterior (10 marks)

When discussing about GP regression, we saw that we can bypass the computation of GP posterior and can directly compute the posterior predictive $p(y_*|\boldsymbol{y})$ for a new input $\boldsymbol{x}_*$. Suppose we do care about the GP posterior and would like to derive its expression, given training data $(\mathbf{X}, \boldsymbol{y}) = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$.

Assume a zero mean GP prior $p(f) = \mathcal{GP}(0, \kappa)$ which, from the GP definition, is equivalent to $p(\boldsymbol{f}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$ where $\boldsymbol{f} = [f(\boldsymbol{x}_1,), \ldots, f(\boldsymbol{x}_N)]^\top$ is an $N \times 1$ vector and $\mathbf{K}$ is the $N \times N$ kernel matrix with $K_{nm} = \kappa(\boldsymbol{x}_n, \boldsymbol{x}_m)$. Assume a likelihood model $p(y_n|\boldsymbol{x}_n, f) = \mathcal{N}(y_n|f(\boldsymbol{x}_n), \sigma^2)$, where $f \sim \mathcal{GP}(0, \kappa)$. Derive the expression for the GP posterior, i.e., $p(\boldsymbol{f}|\boldsymbol{y})$. You are free to use standard results for Gaussians.

Part 2: Visualizing GP Priors and Posteriors for Regression (20 marks)

Assume a GP prior $\mathcal{GP}(0, \kappa)$ where κ is the squared exponential (SE) kernel $\kappa(x, x') = \rho^2 \exp\left(-\frac{(x-x')^2}{\ell^2}\right)$. Note that this is the scalar input version of the SE kernel, and it can be generalized to the vector input case by replacing $(x - x')^2$ by $(\boldsymbol{x} - \boldsymbol{x}')^\top(\boldsymbol{x} - \boldsymbol{x}')$. Assume $\rho^2 = 1$.

Our data will consist of scalar inputs and will be generated using the model $y = \sin(x) + \epsilon_n$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ where $\sigma^2 = 0.05$. We will generate $N = 100$ uniformly spaced inputs $x_1, \ldots, x_N$ in the interval $[0, 4\pi]$ and generate the corresponding outputs $y_1, \ldots, y_N$ from the above model.

For each of the following 5 values of ℓ from $[0.2, 0.5, 1, 2, 10]$, your task will be the following

- Draw a random sample from the GP prior $p(\boldsymbol{f}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$ and plot it.
- Plot the *mean* of the GP posterior (from Part 1), again on the same figure but with a different color.
- On the same plot, also show the true function ($\sin(x)$) evaluated at the generated inputs (use a different color for this too). Note that this curve will be the same in all the 5 cases.

Your code should be submitted as a Python notebook.

You may use any library function to draw from a multivariate normal distribution. Also note that, when computing the kernel matrix $\mathbf{K}$, you may need to add a small positive number to the diagonal entries to make it invertible.

What difference do you see between the plots generated using $\ell = [0.2, 0.5, 1, 2, 10]$, in particular w.r.t. shapes of prior/posterior of GP vs the true function?