Probabilistic Machine Learning
(CS772A, Fall 2022)
Homework 3
Due Date: November 15, 2022 (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/3ytv9jet`). 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: Monte-Carlo Approximations (15 marks)

Consider approximating an expectation $\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$ using S samples $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(L)}$ drawn i.i.d. from $p(\boldsymbol{z})$. Denote the approximated expectation as $\hat{f} = \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{z}^{(\ell)})$. Show that this approximation is unbiased, i.e., $\mathbb{E}[\hat{f}] = \mathbb{E}[f]$. Also show that the variance of this approximation is given by var$[\hat{f}] = \frac{1}{S}\mathbb{E}[(f - \mathbb{E}[f])^2]$, i.e., the well-known result that the Monte-Carlo estimate's variance goes down as S increases.

Problem 2: Mean-Field VI for Sparse Bayesian Linear Regression (30 marks)

Assume N observations $\{\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ generated from a regression model $y_n \sim \mathcal{N}(y_n|\boldsymbol{w}^\top \boldsymbol{x}_n, \beta^{-1})$. Further assume a Gaussian prior on $\boldsymbol{w}$ with different component-wise precisions, i.e., $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \text{diag}(\alpha_1^{-1}, \ldots, \alpha_D^{-1}))$. Also assume gamma priors on the noise precision β and prior's precisions $\{\alpha_d\}_{d=1}^{D}$, i.e., $\beta \sim$ Gamma$(\beta|a_0, b_0)$ and $\alpha_d \sim$ Gamma$(\alpha_d|e_0, f_0), \forall d$. We will use the following parametrization of the gamma: Gamma$(\eta|\tau_1, \tau_2) = \frac{\tau_2^{\tau_1}}{\Gamma(\tau_1)}\eta^{\tau_1 - 1}\exp(-\tau_2\eta)$.

Derive the mean-field variational inference algorithm for approximating the posterior distribution

$$q(\boldsymbol{w}, \beta, \alpha_1, \ldots, \alpha_D) = q(\boldsymbol{w})q(\beta)q(\alpha_1)\ldots q(\alpha_D) \approx p(\boldsymbol{w}, \beta, \alpha_1, \ldots, \alpha_D|\boldsymbol{y}, \mathbf{X})$$

You may use the "recognition" method for inferring each q distribution, or explicitly write down the ELBO for this model and take derivatives w.r.t. the variational parameters of each q distribution to estimate these.

Problem 3: Gibbs Sampling (20 marks)

Suppose we are given a bunch of count-valued observations $x_1, \ldots, x_N$, assumed generated from the following hierarchical model: $p(x_n|\lambda_n) =$ Poisson$(x_n|\lambda_n)$, $p(\lambda_n|\alpha, \beta) =$ Gamma$(\lambda_n|\alpha, \beta)$, $n = 1, \ldots, N$, $p(\alpha|a, b) =$ Gamma$(\alpha|a, b)$, and $p(\beta|c, d) =$ Gamma$(\beta|c, d)$. Assume a, b, c, d to be fixed.

We would like to do Gibbs sampling for this model. To do so, derive the conditional posterior (CP) of each variable $\lambda_1, \ldots, \lambda_N$, α, and β, given all the other variables. Are all CPs available in closed form?

Problem 4: Using Samples for Prediction (15 marks)

Consider a matrix factorization model for a partially observed $N \times M$ matrix $\mathbf{R}$, where $p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j) = \mathcal{N}(r_{ij}|\boldsymbol{u}_i^\top \boldsymbol{v}_j, \beta^{-1})$, and $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ denote the latent factors of i-th row and j-th column or $\mathbf{R}$, respectively. The posterior predictive distribution of each r_{ij} is defined as $p(r_{ij}|\mathbf{R}) = \int p(r_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j)p(\boldsymbol{u}_i, \boldsymbol{v}_j|\mathbf{R})d\boldsymbol{u}_i d\boldsymbol{v}_j$, which is in general intractable. Suppose we are given a set of S samples $\{\mathbf{U}^{(s)}, \mathbf{V}^{(s)}\}_{s=1}^{S}$ generated by a Gibbs sampler for this matrix factorization model, where $\mathbf{U}^{(s)} = \{\boldsymbol{u}_i^{(s)}\}_{i=1}^{N}$ and $\mathbf{V}^{(s)} = \{\boldsymbol{v}_j^{(s)}\}_{j=1}^{M}$.

Given these samples, derive the expressions for the sample based approximation of the *mean (expectation)* as well as the *variance* of any entry r_{ij} of the matrix $\mathbf{R}$.

Hint: Note that we can write each r_{ij} as $\boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$ where $\epsilon_{ij} \sim \mathcal{N}(\epsilon_{ij}|0, \beta^{-1})$.

Problem 5: Implementing Samplers (20+20 = 40 marks)

(Part 1: Implementing A Rejection Sampler) Consider a distribution $p(x) = \frac{\tilde{p}(x)}{Z}$ where $\tilde{p}(x) = \mathcal{N}(x|20, 10^2) + \mathcal{N}(x|50, 5^2) + \mathcal{N}(x|80, 20^2)$. Implement a rejection sampler that generates 10,000 samples from $p(x)$ using a Gaussian proposal $q(x) = \mathcal{N}(x|50, 30^2)$. To do rejection sampling, you also need to find an M s.t. $Mq(x) \geq \tilde{p}(x), \forall x$. Note that a choice of $M = \max_x \frac{\tilde{p}(x)}{q(x)}$ will guarantee this (note that a larger M can also be used but will lead to larger rejection rate). You can find M by computing this maxima over a sufficiently large number of x values from a pre-specified range (e.g., [-50,100]). What value of M do you get?

Given this M, run your rejection sampler, and on the same figure, show (in different colors), the original unnormalized distribution $\tilde{p}(x)$, your proposal $q(x)$, and a histogram plot of the accepted samples. What's the acceptance rate? Does it make sense based on the value of M you found? Submit your code as well as the figure.

(Part 2: Implementing MH Sampling for 2-D Gaussian) In this problem, your task is to implement MH sampling to generate random samples from a 2-D Gaussian $p(\boldsymbol{z}) = \mathcal{N}\left(\begin{bmatrix}4\\4\end{bmatrix}, \begin{bmatrix}1 & 0.8\\0.8 & 1\end{bmatrix}\right)$.

To sample from $p(\boldsymbol{z})$, you will use a proposal distribution $q(\boldsymbol{z}^{(t)}|\boldsymbol{z}^{(t-1)}) = \mathcal{N}\left(\boldsymbol{z}^{(t-1)}, \begin{bmatrix}\sigma^2 & 0\\0 & \sigma^2\end{bmatrix}\right)$ and play with different values of the proposal distribution's variance σ^2. For generating the candidate sample from the proposal distribution, you can use existing functions from Python libraries, but you must not use the actual distribution $p(\boldsymbol{z})$ to generate the samples.

You will experiment with the following values of $\sigma^2 : 0.01, 1, 100$. For each of these cases, run the MH sampler long enough to collect 10,000 samples and show the plots of the generated samples on a 2-D plane for 100 samples, 1000 samples, and 10,000 samples (similar to the plots of slide 4, lecture-17).

Looking at the plots, which of the 3 proposals ($q(\boldsymbol{z}^{(t)}|\boldsymbol{z}^{(t-1)})$ with $\sigma^2 : 0.01, 1, 100$) seems the best choice to you? What is the rejection rate in each of these cases (rejection rate is the ratio of number of samples rejected and the total number of candidate samples generated)? Submit the code as well as the plots.