Probabilistic Machine Learning
(CS772A, Spring 2023)
Homework 1
Due Date: February 7, 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/cs772sp23hw1`). 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 `hw1sol.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 (10 marks)

(Prior Hierarchy) Consider a model m with parameters θ and hyperparameters λ. Assume priors $p(\theta|\lambda, m)$, $p(\lambda|m)$ and $p(m)$. Assume we have observed some data $\mathbf{X}$ and the likelihood is of the form $p(\mathbf{X}|\theta, \lambda, m)$. For this problem setup, write down the expressions for computing: (1) $p(\theta|\mathbf{X}, \lambda, m)$, (2) $p(\lambda|\mathbf{X}, m)$, and (3) $p(m|\mathbf{X})$. Also rank these three quantities in terms of the difficulty of computing them (easiest to hardest) and briefly justify your ranking. Note: Your answers should not assume any conjugacy. Also, all quantities in each of these 3 expression should clearly and explicitly show everything you need to condition on.

Problem 2 (10 marks)

(It Gets Better..) Recall that, for a Bayesian linear regression model with likelihood $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}, \beta^{-1})$ and prior $p(\boldsymbol{w}) = \mathcal{N}or(0, \lambda^{-1}\mathbf{I})$, the *predictive posterior* is $p(y_*|\boldsymbol{x}_*) = \mathcal{N}(\mu_N^\top \boldsymbol{x}_*, \beta^{-1} + \boldsymbol{x}_*^\top \Sigma_N \boldsymbol{x}_*) = \mathcal{N}(\mu_N^\top \boldsymbol{x}_*, \sigma_N^2(\boldsymbol{x}_*))$, where we have defined $\sigma_N^2(\boldsymbol{x}_*) = \beta^{-1} + \boldsymbol{x}_*^\top \Sigma_N \boldsymbol{x}_*$ and μ_N and Σ_N are the mean and covariance matrix of the Gaussian posterior on $\boldsymbol{w}$, s.t., $\mu_N = \Sigma(\beta \sum_{n=1}^N y_n \boldsymbol{x}_n)$ and $\Sigma_N = (\beta \sum_{n=1}^N \boldsymbol{x}_n \boldsymbol{x}_n^\top + \lambda \mathbf{I})^{-1}$. Here, we have used the subscript N to denote that the model is learned using N training examples. As the training set size N increases, what happens to the variance of the predictive posterior? Does it increase or decrease or remain the same? You must also prove your answer formally. You might find the following identity useful: You may make use the following matrix identity:

$$(\mathbf{M} + \boldsymbol{v}\boldsymbol{v}^\top)^{-1} = \mathbf{M}^{-1} - \frac{(\mathbf{M}^{-1}\boldsymbol{v})(\boldsymbol{v}^\top \mathbf{M}^{-1})}{1 + \boldsymbol{v}^\top \mathbf{M}^{-1} \boldsymbol{v}}$$

Where $\mathbf{M}$ denotes a square matrix and $\boldsymbol{v}$ denotes a column vector.

Problem 3 (20 marks)

(When You Integrate Out..) Suppose x is a scalar random variable drawn from a univariate Gaussian $p(x|\eta) = \mathcal{N}(x|0, \eta)$. The variance η itself is drawn from an exponential distribution: $p(\eta|\gamma) = \text{Exp}(\eta|\gamma^2/2)$, where $\gamma > 0$. Note that the exponential distribution is defined as $\text{Exp}(x|\lambda) = \lambda \exp(-\lambda x)$. Derive the expression of the marginal distribution of x, i.e., $p(x|\gamma) = \int p(x|\eta)p(\eta|\gamma)d\eta$ after integrating out η. What does the marginal distribution $p(x|\gamma)$ mean?

Plot both $p(x|\eta)$ and $p(x|\gamma)$ and include in the writeup PDF itself. What difference do you see between the shapes of these two distributions? **Note:** You don't need to submit the code used to generate the plots. Just the plots (appropriately labeled) are fine.

Hint: You will notice that $\int p(x|\eta)p(\eta|\gamma)d\eta$ is a hard to compute integral. However, the solution does have a closed form expression. One way to get the result is to compute the **moment generating function (MGF)**[1] of $\int p(x|\eta)p(\eta|\gamma)d\eta$ (note that this is a p.d.f.) and compare the obtained MGF expression with the MGFs of various p.d.f.s given in the table on the following Wikipedia page: `https://en.wikipedia.org/wiki/Moment-generating_function`, and identify which p.d.f.'s MGF it matches with. That will give you the form of distribution $p(x|\gamma)$. Specifically, name this distribution and identify its parameters.

Problem 4 (20 marks)

(Hierarchical Modeling) Suppose we have student data from M schools where N_m denotes the number of students in school m. The data for each school $m = 1, \ldots, M$ is in the following form: For student n in school m, there is a response variable (e.g., score in some exam) $y_n^{(m)} \in \mathbb{R}$ and a feature vector $\boldsymbol{x}_n^{(m)} \in \mathbb{R}^D$.

Assume a linear regression model for these scores, i.e., $p(y_n^{(m)}|\boldsymbol{x}_n^{(m)}, \boldsymbol{w}_m) = \mathcal{N}(y_n^{(m)}|\boldsymbol{w}_m^\top \boldsymbol{x}_n^{(m)}, \beta^{-1})$, where $\boldsymbol{w}_m \in \mathbb{R}^D$ denotes the regression weight vector for school m, and β is known. Note that this can also be denoted as $p(\boldsymbol{y}^{(m)}|\mathbf{X}^{(m)}, \boldsymbol{w}_m) = \mathcal{N}(\boldsymbol{y}^{(m)}|\mathbf{X}^{(m)}\boldsymbol{w}_m, \beta^{-1}\mathbf{I}_{N_m})$, where $\boldsymbol{y}^{(m)}$ is $N_m \times 1$ and $\mathbf{X}^{(m)}$ is $N_m \times D$. Assume a prior $p(\boldsymbol{w}_m) = \mathcal{N}(\boldsymbol{w}_m|\boldsymbol{w}_0, \lambda^{-1}\mathbf{I}_D)$, λ to be known and $\boldsymbol{w}_0$ to be unknown.

[1] MGF of a p.d.f. $p(x)$ is defined as $M_X(t) = \int_{-\infty}^{\infty} e^{tx} p(x) dx$

Derive the expression for the **log** of the MLE-II objective for estimating $\boldsymbol{w}_0$. **You do not need to optimize this objective w.r.t.** $\boldsymbol{w}_0$; just writing down the final expression of objective function is fine. Also state what is the benefit of this approach as opposed to fixing $\boldsymbol{w}_0$ to some value, if our goal is to learn the school-specific weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$? (Feel free to make direct use of properties of Gaussian distributions; you may refer to the provided results in the prob-stats refresher slides, or in books).

Problem 5 (10 marks)

(Peeking into the neighborhood) Consider a regression model where the joint distribution of any input $\boldsymbol{x} \in \mathbb{R}^D$ and its output $y \in \mathbb{R}$ is $p(\boldsymbol{x}, y) = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x} - \boldsymbol{x}_n, y - y_n)$ where $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ denotes the training examples. Further assume $f(\boldsymbol{x} - \boldsymbol{x}_n, y - y_n) = \mathcal{N}([\boldsymbol{x} - \boldsymbol{x}_n, y - y_n]^\top | \boldsymbol{0}, \sigma^2 \mathbf{I}_{D+1})$. For this model, derive the conditional distribution of the output y given the input, i.e., $p(y|\boldsymbol{x})$, as well as the expectation $\mathbb{E}[y|\boldsymbol{x}]$. Also give a brief justification as to why the expressions $p(y|\boldsymbol{x})$ and $\mathbb{E}[y|\boldsymbol{x}]$ make intuitive sense.

Problem 6 (30 marks): Programming Assignment

(Bayesian Linear Regression) Consider a toy data set consisting of 10 training examples $\{x_n, y_n\}_{n=1}^{10}$ with each input x_n as well as the output y_n being scalars. The data is given below.

$$\begin{aligned} \boldsymbol{x} &= [-2.23, -1.30, -0.42, 0.30, 0.33, 0.52, 0.87, 1.80, 2.74, 3.62]; \\ \boldsymbol{y} &= [1.01, 0.69, -0.66, -1.34, -1.75, -0.98, 0.25, 1.57, 1.65, 1.51] \end{aligned}$$

We would like to learn a Bayesian linear regression model using this data, assuming a Gaussian likelihood model for the outputs with fixed noise precision $\beta = 4$. However, instead of working with the original scalar-valued inputs, we will map each input x using a degree-k polynomial as $\phi_k(x) = [1, x, x^2, \ldots, x^k]^\top$. Note that, when using the mapping ϕ_k, each original input becomes $k+1$ dimensional. Denote the entire set of mapped inputs as $\phi_k(\boldsymbol{x})$, a $10 \times (k+1)$ matrix. Consider $k = 1, 2, 3$ and 4, and learn a Bayesian linear regression model for each case. Assume the following prior on the regression weights: $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \mathbf{I})$ with $\boldsymbol{w} \in \mathbb{R}^{k+1}$.

1. For each k, compute the posterior of $\boldsymbol{w}$ and show a plot with 10 random functions drawn from the inferred posterior (show the functions for the input range $x \in [-4, 4]$). Also show the original training examples on the same plot to illustrate how well the functions fit the training data.

2. For each k, compute and plot the **mean** of the posterior predictive $p(y_*|\phi_k(x_*), \phi_k(\boldsymbol{x}), \boldsymbol{y}, \beta)$ on the interval $x_* \in [-4, 4]$. On the same plot, also show the predictive posterior mean plus-and-minus two times the predictive posterior standard deviation.

3. Compute the log marginal likelihood $\log p(\boldsymbol{y} \mid \phi_k(\boldsymbol{x}), \beta)$ of the training data for each of the 4 mappings $k = 1, 2, 3, 4$. Which of these 4 "models" seems to explain the data the best?

4. Using the MAP estimate $\boldsymbol{w}_{MAP}$, Compute the log likelihood $\log p(\boldsymbol{y}|\boldsymbol{w}_{MAP}, \phi_k(\boldsymbol{x}), \beta)$ for each k. Which of these 4 models seems to have the highest log likelihood? Is your answer the same as that based on the log marginal likelihood (part 3)? Which of these two criteria (highest log likelihood or highest log marginal likelihood) do you think is more reasonable to select the best model and why?

5. For your best model, suppose you could include an additional training input x' (along with its output y') to "improve" your learned model using this additional example. Where in the region $x \in [-4, 4]$ would you like the chosen x' to be? Explain your answer briefly,

Your implementation should be in Python notebook (and should not use an existing implementation of Bayesian linear regression from any library).

Submit the plots as well as the code in a single zip file (named `yourrollnumber.zip`).