Name:

Roll No.: **Dept.:**

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: April 28, 2024

Instructions: *Total:* **100 marks**

1. Total duration: **3 hours**. Please write your name, roll number, department on **all pages**.
2. This booklet has **8 pages**. Answer to each question must be written in space provided under that question. For rough work, additional blank sheets may be provided if needed.
3. Avoid showing very detailed derivations but please do show the key steps.
4. Some essential formulae, equations for probability distributions, etc., are provided at the end of the question paper.

Section 1 (True or False: 20 X 1 = 20 marks)**.** For each of the following simply write **T** or **F** in the box.

1. [T] When using conformal prediction for a classification problem, the prediction set computed by the algorithm is larger for more difficult inputs.
2. [F] The posterior predictive distribution (PPD) can be computed in closed form when using Laplace approximation of the posterior.
3. [T] Standard variational inference tends to underestimate the variance of the posterior distribution.
4. [T] Gaussian Process is usually slower at test time than a deep neural network.
5. [F] Bayesian Optimization is used as an efficient method to compute point estimates of the parametes of a probabilistic model.
6. [T] For latent Dirichlet allocation (LDA) topic model, given only word-to-topic assignments of all the words in the training corpus, we can compute the topic proportions for each document.
7. [F] The standard denoising diffusion model is also a dimensionality reduction method.
8. [F] Gibbs sampler can be used to compute the (conditional) posterior distribution for the weight vector of any generalized linear model (GLM). Note: CP of $\boldsymbol{w}$ isn't guranteed to have a closed form (e.g., in logistic regression). To derive a Gibbs sampler for $\boldsymbol{w}$ and other unknowns of the model, lack of closed form CP of $\boldsymbol{w}$ will be an issue, but we can sample from CP of $\boldsymbol{w}$ using an MH-within-Gibbs sampler. I actually only intended to ask if CP of $\boldsymbol{w}$ has a closed form but mention of Gibbs sampler possibly caused confusion. We will accept both T and F as correct.
9. [T] Black-box VI which uses score-function based gradient is applicable for a much broader class of probabilistic models than the reparametrization trick based VI.
10. [T] Amortized VI is useful/needed only in probabilistic models that contain local latent variables (e.g., one latent variable per data point).
11. [F] When computing the posterior predictive distribution (PPD) at test-time, using a VI based posterior is always faster than using a sampling (e.g., MCMC) based posterior.
12. [T] The prior distribution of the latent variables in a hidden Markov model has more parameters than the prior distribution of the latent variables in a Gaussian mixture model.
13. [F] Unlike ensemble methods, in frequentist statistics, we do not need to train the model multiple times to get model/predictive uncertainty.
14. [T] Unlike MCMC, when using rejection sampling to generate a number of samples from some distribution, we don't need to use any thinning to ensure that the samples are uncorrelated.
15. [F] When using a symmetric proposal in MCMC, the acceptance probability becomes equal to 1.
16. [F] The EM algorithm cannot be used to compute MAP estimates of the parameters.
17. [T] Unlike GAN, generative models like VAE and denoising diffusion models do not suffer so much from the mode collapse issue.
18. [F] The softmax probabilities from a linear classifier or a deep neural net classifier trained using NLL loss function are well-calibrated.
19. [T] Low entropies of the predictive distributions computed using a classification model imply that the trained model has poor calibration.
20. [F] A generative supervised learning model can only be used for classification problems and not for regression problems.

Name: □

Roll No.: □ Dept.: □

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: April 28, 2024

Section 2 (10 questions: 80 marks). .

1. **(10 marks)** Marginal likelihood is a measure of "goodness" of a model. Given training data $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, show that the log of the marginal likelihood, i.e., $\log p(\mathcal{D}|m)$, can be written as a summation consisting of N terms where each of the terms in the summation is akin to the (log of the) posterior predictive distribution (PPD) of one of the N training observations. Also clearly specify what will be the corresponding posterior distribution in each of these N PPDs?

   Now consider another method (let's call it "method 2") where the summation used in the above method (let's call it "method 1") does not go from $n = 1$ to N but from $n = k$ to N, where k is some number that is reasonably larger than 1. Would method 2 be better than method 1? Briefly justify your answer.

   Finally consider "method 3", which is known as leave-one-out cross-validation (LOO-CV) where we compute the quantity $\mathcal{L}_{LOO}(m) = \sum_{n=1}^{N} \log p(x_n|\hat{\theta}(\mathcal{D}_{-n}), m)$ where $\hat{\theta}(\mathcal{D}_{-n})$ denotes a point estimate of θ using the remaining $N-1$ observations. Briefly compare and contrast method 3 with method 1.

   For method 1, using the chain rule, we can write $p(\mathcal{D}|m)$ as $p(x_1, x_2, \ldots, x_N|m) = \prod_{n=1}^{N} p(x_n|x_{<n}, m)$ where $x_{<n}$ denotes the observations from x_1 to x_{n-1}.

   Therefore, for method 1, $\log p(\mathcal{D}|m) = \sum_{n=1}^{N} \log p(x_n|x_{<n}, m)$, which is a summation of N log-PPDs, since $p(x_n|x_{<n}, m)$ is equivalent to the PPD of x_n given $x_{<n}$ as training data and can be written as

   $$p(x_n|x_{<n}, m) = \int p(x_n|\theta, m) p(\theta|x_{<n}, m) d\theta$$

   where the posterior distribution is $p(\theta|x_{<n}, m)$, i.e., the posterior of θ given the observations $x_{<n}$.

   For method 2, the log marginal likelihood is defined as $\log p(\mathcal{D}|m) = \sum_{n=k}^{N} \log p(x_n|x_{<n}, m)$.

   Note that the PPDs in the initial few terms in the summation used in method 1 depend on very little amount of training data and thus those PPD estimates may not be very accurate (because the corresponding posterior approximation would also be poor). On the other hand, in method 2, the summation starts with $n = k$, so each PPD term in that summation uses k or more training observations, and thus we will get reasonable estimates of the PPD for each of the terms in the summation. Therefore, method 2 is expected to be better than method 1 (since basically we dropped the first $k-1$ "bad" terms).

   Method 3 (LOO-CV) is different from method 1 in two ways: (1) Each term in the summation is not a PPD but the predictive distribution of x_n based on the point estimate of θ, and (2) The predictive distribution of each x_n depends on all the other $N-1$ observations, not just the "previous" $n-1$ observations.

2. **(6 marks)** KL divergence between two distributions $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$ is defined as $\text{KL}[p_1(\boldsymbol{x})||p_2(\boldsymbol{x})] = \int p_1(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} d\boldsymbol{x}$. Assuming $p(\boldsymbol{x}|\theta_1)$ and $p(\boldsymbol{x}|\theta_2)$ to be two exponential family distributions from the same family, where θ_1 and θ_2 denote their respective natural parameters, derive the expression for $\text{KL}[p(\boldsymbol{x}|\theta_1)||p(\boldsymbol{x}|\theta_2)]$.

   Using the expression of exponential family distribution $p(\boldsymbol{x}|\theta) = h(\boldsymbol{x}) \exp[\theta^\top \phi(\boldsymbol{x}) - A(\theta)]$ where $\phi(\boldsymbol{x})$ denotes the sufficient statistics, the KL divergence between the given distributions can be written as $\text{KL}[p(\boldsymbol{x}|\theta_1)||p(\boldsymbol{x}|\theta_2)] = \int p(\boldsymbol{x}|\theta_1) \log \frac{p(\boldsymbol{x}|\theta_1)}{p(\boldsymbol{x}|\theta_2)} d\boldsymbol{x}$. Substituting the expressions of the distributions involved

   $$\begin{aligned} \text{KL}[p(\boldsymbol{x}|\theta_1)||p(\boldsymbol{x}|\theta_2)] = \int p(\boldsymbol{x}|\theta_1) \log \frac{p(\boldsymbol{x}|\theta_1)}{p(\boldsymbol{x}|\theta_2)} d\boldsymbol{x} &= \int p(\boldsymbol{x}|\theta_1)[(\theta_1 - \theta_2)^\top \phi(\boldsymbol{x}) - (A(\theta_1) - A(\theta_2))] d\boldsymbol{x} \\ &= (\theta_1 - \theta_2)^\top \mathbb{E}_{p(\boldsymbol{x}|\theta_1)}[\phi(\boldsymbol{x})] - (A(\theta_1) - A(\theta_2)) \end{aligned}$$

   Note that $\mathbb{E}_{p(\boldsymbol{x}|\theta_1)}[\phi(\boldsymbol{x})]$ above is the first derivative of the log-partition function $A(\theta_1)$.

Name:

Roll No.: Dept.:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: April 28, 2024

3. **(6 marks)** For a GMM $p(\boldsymbol{x}_n|\boldsymbol{z}_n,\mu,\Sigma) = \mathcal{N}(\boldsymbol{x}_n|\mu_{z_n},\Sigma_{z_n})$, can the latent variable $z_n \in \{1,2,\dots,K\}$ denoting the cluster assignment of data point $\boldsymbol{x}_n$, and having prior $p(z_n|\boldsymbol{\pi}) = \text{multinoulli}(z_n|\pi_1,\dots,\pi_K)$, be collapsed and posterior inference be performed only for the remaining unknowns of the model? Briefly justify your answer. Likewise, for a proabilistic PCA model $p(\boldsymbol{x}_n|\boldsymbol{z}_n,\mathbf{W},\sigma^2) = \mathcal{N}(\boldsymbol{x}_n|\mathbf{W}\boldsymbol{z}_n,\sigma^2\mathbf{I}_D)$, can the latent variables $\boldsymbol{z}_n \in \mathbb{R}^K$ having a Gaussian prior $\mathcal{N}(\boldsymbol{z}_n|0,\mathbf{I}_K)$ be collapsed and posterior inference be performed only for the remaining unknowns of the model? Briefly justify your answer.

In both cases, we can easily integrate out $\boldsymbol{z}_n$ from the model. For the GMM, we can integrate out $\boldsymbol{z}_n$ to obtain $p(\boldsymbol{x}_n|\mu,\Sigma) = \sum_{k=1}^K p(\boldsymbol{z}_n = k|\pi)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k,\mu,\Sigma) = \sum_{k=1}^K \pi_k\mathcal{N}(\boldsymbol{x}_n|\mu_k,\Sigma_k)$ and use it as the likelihood, along with priors on μ and Σ and we won't need to estimate $\boldsymbol{z}_n$.

For the PPCA model, we can likewise integrate out $\boldsymbol{z}_n$ easily to obtain the likelihood $p(\boldsymbol{x}_n|\mathbf{W},\sigma^2)$ which will still be a Gaussian (since PPCA is a linear Gaussian model).

4. **(6 marks)** The entropy of a distribution $p(x|\theta)$ is defined as $\mathbb{H}[p(x|\theta)] = -\int p(x|\theta)\log p(x|\theta)dx$. Derive the expression for the entropy of a univariate Gaussian distribution $p(x|\theta) = \mathcal{N}(x|\mu,\sigma^2)$ and briefly explain why the expression you obtain makes intuitive sense.

Plugging in $p(x|\theta) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$ into the expression of entropy, we get

$$\begin{aligned}\mathbb{H}[p(x|\theta)] &= -\int p(x|\theta)\left[\log\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(x-\mu)^2}{2\sigma^2}\right]dx \\ &= -\log\frac{1}{\sqrt{2\pi\sigma^2}} + \mathbb{E}_{p(x|\theta)}\left[\frac{(x-\mu)^2}{2\sigma^2}\right] \\ &= \log\sqrt{2\pi\sigma^2} + \frac{1}{2} \qquad \text{(using } \mathbb{E}[(x-\mu)^2] = \sigma^2\end{aligned}$$

As we know, entropy is a measure of randomness/uncertainty. As shown above, the entropy of a Gaussian distribution is proportional to its variance σ^2 (interestingly, it doesn't depend on the mean!). Larger variance of the distribution implies large uncertainty which in turn implies high entropy..

5. **(8 marks)** Consider Gaussian Process (GP) classification with likelihood $p(y_n|\boldsymbol{x}_n) = \text{Bernoulli}[y_n|\sigma(f_n)]$ where $f_n = f(\boldsymbol{x}_n)$ is the score of input $\boldsymbol{x}_n$, and $\sigma(.)$ denotes the sigmoid function. Note that a high positive (cf, high negative) score f_n means a high probability of $\boldsymbol{x}_n$ belonging to the positive (cf, negative) class. For GP, given a new test input $\boldsymbol{x}_*$, we know that $p(f_*|\boldsymbol{f})$, where $f_* = f(\boldsymbol{x}_*)$ and $\boldsymbol{f} = [f_1,\dots,f_N]$ are scores of N training inputs, has the form $\mathcal{N}(f_*|\mu_*,\sigma_*^2)$. For an input $\boldsymbol{x}_*$, suggest appropriate choices of active learning acquisition functions $A(\boldsymbol{x}_*)$ which depend on: (1) only μ_* (2) only σ_*^2 and (3) both μ_* and σ_*^2. Briefly justify why each of these 3 acquisition functions make sense for active learning.

(1) Note that a very small value (positive of negative) of μ_* implies $p(y_n = 1|\boldsymbol{x}_n) \approx 0.5$, i.e., the model is uncertain about the predicted label (both classes are roughly equally likely). Thus we can define $A(\boldsymbol{x}_*) = |\mu_*|$ and we pick the optimal input as the one with smallest $A(\boldsymbol{x}_*)$. This is equivalent to active learning preferring inputs that fall close to the decision boundary (and thus have small margin).

(2) A high value is σ_*^2 implies high variance (low confidence) in the predicted label y_*. Thus we can define $A(\boldsymbol{x}_*) = \sigma_*^2$ and we pick the optimal input as the one with largest $A(\boldsymbol{x}_*)$.

(3) As we have seen, when using μ_* and σ_*^2 in isolation, we prefer choosing inputs with small absolute value of μ_* and large value of σ_*^2. A reasonable acquisition function that takes into account both these quantities can be $A(\boldsymbol{x}_*) = \frac{|\mu_*|}{\sigma_*^2}$ and we pick the optimal input as the one with smallest $A(\boldsymbol{x}_*)$.

6. **(8 marks)** Briefly explain why the objective $\min_G \max_D \ \mathbb{E}_{p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}))]$ used for learning a generative adversarial network (GAN) makes sense where D is optimized keeping G fixed, and vice-versa. Here D and G denote the discriminator and generator models, respectively, $p_{data}(\boldsymbol{x})$ denotes the distribution of real training data, and $p_{\boldsymbol{z}}(\boldsymbol{z})$ denotes the noise distribution.

   When G is fixed at $\hat{G}$ (current optimal), we solve $\max_D \ \mathbb{E}_{p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(\hat{G}(\boldsymbol{z})))]$. Maximization would yield a D such that $D(x)$ (where $x \sim p_{data}(x)$) is close to 1 and $D(\hat{G}(\boldsymbol{z}))$ (where $\boldsymbol{z} \sim p_z(\boldsymbol{z})$) is close to 0, i.e., discriminator D correctly predicts x as real, and $\hat{G}(z)$ as fake.

   When D is fixed at $\hat{D}$ (current optimal), we solve $\min_G \ \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(\hat{G}(\boldsymbol{z})))]$. Minimization would yield a G such that $\hat{D}(G(\boldsymbol{z}))$ (where $\boldsymbol{z} \sim p_z(\boldsymbol{z})$) is close to 1, i.e., G generates $G(\boldsymbol{z})$ which the discriminator is fooled into believing to be real.

7. **(6 marks)** Consider a regression model of the form $p(y_n|\boldsymbol{x}_n, \boldsymbol{w}_1, \boldsymbol{w}_2) = \mathcal{N}(y_n|\boldsymbol{w}_1^\top \boldsymbol{x}_n, \exp(\boldsymbol{w}_2^\top \boldsymbol{x}_n))$. Suppose we have estimated the weights using MLE/MAP. Denoting these estimates as $\hat{\boldsymbol{w}}_1$ and $\hat{\boldsymbol{w}}_2$, we can compute the predictive distribution of y_* for a new input $\boldsymbol{x}_*$ as $p(y_*|\boldsymbol{x}_*, \hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2) = \mathcal{N}(y_*|\hat{\boldsymbol{w}}_1^\top \boldsymbol{x}_*, \exp(\hat{\boldsymbol{w}}_2^\top \boldsymbol{x}_*))$.

   While this approach provides an estimate of predictive uncertainty about y_*, in what sense it is different from the Bayesian approach which also provides an estimate of predictive uncertainty via the variance of the PPD of y_*? In particular, if the amount of training data is small, which of these two approaches will provide a better estimate of the predictive uncertainty and why?

   While the first approach provides an estimate of predictive uncertainty via the variance $\exp(\hat{\boldsymbol{w}}_2^\top \boldsymbol{x}_*)$, it is only accounting for aleatoric uncertainty in the predictions. Since it only uses a point estimate of the weights, the predictions do not account for the model (epistemic) uncertainty. In contrast, a Bayesian approach also estimates model uncertainty via the posterior and thus the prediction also takes into account the model uncertainty ("prediction is uncertain because the model itself is uncertain").

   When the amount of training data is small, we expect a fair bit of model uncertainty (which will be important to take into account) and the Bayesian approach would typically do a better job. In contrast, the other approach which uses point estimates of the weights, will have a risk of overfitting due to limited training data.

8. **(6 marks)** Expected calibration error of a model f is defined as $\text{ECE}(f) = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{N} |\text{acc}(\mathcal{B}_b) - \text{conf}(\mathcal{B}_b)|$, where $\mathcal{B}_b$ denotes the number of inputs for which f's confidence (probability) for the predicted class falls in the b^{th} bin, and N denotes the total number of inputs. From the perspective of the reliability diagram, the ECE metric basically computes the "average" miscalibration. Suggest a modification of ECE that computes the *maximal* miscalibration instead of average and write down the expression of this metric.

   Also note that ECE is computed only using the information about the predicted class (and f's confidence in it) of each input. Suggest a modification of ECE that, for each input, considers not just predicted class but all K classes (with f's confidence in each of them), and write down the expression of this metric.

   Standard ECE performs a frequency based average of the miscalibrations (difference between accuracy and confidence) across all the B bins in the reliability diagram. The maximal miscalibration approach would care only about the maximum difference across the bins: $\text{ECE}(f) = \max_{b\in\{1,2,\ldots,B\}} |\text{acc}(\mathcal{B}_b) - \text{conf}(\mathcal{B}_b)|$.

   To consider all the K classes, one way is to construct K reliability diagrams (each with B bins) and compute a sum of ECE across all the K classes. The ECE can be (re)defined as follows:

$$\text{ECE}(f) = \frac{1}{K} \sum_{k=1}^{K} \sum_{b=1}^{B} \frac{|\mathcal{B}_{b,k}|}{N} |\text{acc}(\mathcal{B}_{b,k}) - \text{conf}(\mathcal{B}_{b,k})|$$

   where $\text{acc}(\mathcal{B}_{b,k})$ and $\text{conf}(\mathcal{B}_{b,k})$ denote the accuracy and confidence, respectively, of bin b in the reliability diagram of class k.

9. **(8 marks)** Consider an infinite sequence $\pi_1, \pi_2, \pi_3, \ldots$ where the π_k's are constructed by first generating an infinite sequence of random variables $\beta_1, \beta_2, \beta_3, \ldots$, each drawn i.i.d. from a beta distribution $\text{Beta}(\alpha, 1)$ and then defining the π_k's such that $\pi_1 = \beta_1$ and $\pi_k = \beta_k \pi_{k-1}$ for $k > 1$. What property do the values in this infinite sequence $\pi_1, \pi_2, \pi_3, \ldots$ have? Also, is it guaranteed to sum to 1? Briefly justify your answer.

Since each β_k is drawn from a beta distribution, it must be less than 1. Therefore, defining $\pi_k = \beta_k \pi_{k-1}$ ensures that each value in the sequence $\pi_1, \pi_2, \pi_3, \ldots$ will be smaller than the previous value. Therefore, we will get a non-increasing sequence where each value is between 0 and 1. You may think of this approach as producing a sequence of non-increasing probability values with $\pi_k \to 0$ as $k \to \infty$.

This infinite series is not guraranteed to sum to 1. Clearly, all we are doing is producing a sequence of numbers that is decreasing, but that alone does not guarantee that it will sum to 1.

(Not needed for the answer) As a side note, although you can view this process also as a stick-breaking process (using an initial stick length of 1), unlike the other stick-breaking construction we saw in class, here we don't recurse on the remaining stick, but recurse on the part we keep in every round (which is another a proof of why the sequence doesn't sum to 1).

(Not needed for the answer) This stick-breaking construction has been used in stick-breaking VAE where the length K of the VAE latent code $\boldsymbol{z}$ can be learned using this construction. Basically, we assume the dimensionality of the latent code $\boldsymbol{z}$ to be very very large but the probability of using each dimension becomes smaller and smaller with increasing dimension index. If interested, you may see the paper "Stick-Breaking Variational Autoencoders" (Nalisnick and Smyth, ICLR 2017).

10. **(16 marks)** Suppose we are given N training examples $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$. Assume each output y_n is generated using a probabilistic linear regression model of the form $p(y_n|z_n, \boldsymbol{x}_n, \boldsymbol{w}, \beta) = \mathcal{N}(y_n|\boldsymbol{w}_{z_n}^\top \boldsymbol{x}_n, \beta_{z_n}^{-1})$, where $z_n \in \{0, 1\}$ is a latent variable with prior $p(z_n|\pi) = \text{Bernoulli}(z_n|\pi)$, indicating which of the two regression models - one with parameters $\{\boldsymbol{w}_0, \beta_0\}$, and the other with parameters $\{\boldsymbol{w}_1, \beta_1\}$ - is assumed to generate the response y_n. Also assume priors on the weight vectors as $p(\boldsymbol{w}_0|\lambda_0) = \mathcal{N}(\boldsymbol{w}_0|\mathbf{0}, \lambda_0^{-1}\mathbf{I}_D)$ and $p(\boldsymbol{w}_1|\lambda_1) = \mathcal{N}(\boldsymbol{w}_1|\mathbf{0}, \lambda_1^{-1}\mathbf{I}_D)$, a prior $p(\pi|a, b) = \text{Beta}(\pi|a, b)$, and assume the hyperparameters $\theta = (\beta_0, \beta_1, \lambda_0, \lambda_1, a, b)$ to be known. The unknowns in this model are $\mathbf{W} = \{\boldsymbol{w}_0, \boldsymbol{w}_1\}$, $\mathbf{Z} = \{z_n\}_{n=1}^N$, and π.

Derive a Gibbs sampler for approximating the posterior $p(\mathbf{W}, \mathbf{Z}, \pi|\mathcal{D}, \theta)$? You do not need to show very detailed steps for the derivation but please do show the key steps, the final equations for the conditional posteriors, and the overall sketch of the Gibbs sampling algorithm.

This model is a mixture of two linear regression models. The joint distribution of data and unknowns

$$p(\boldsymbol{y}, \mathbf{Z}, \mathbf{W}, \pi|\mathbf{X}, \theta) = \left[\prod_{n=1}^{N} p(y_n|z_n, \boldsymbol{x}_n, \boldsymbol{w}_0, \boldsymbol{w}_1, \beta_0, \beta_1) p(z_n|\pi)\right] p(\boldsymbol{w}_0|\lambda_0) p(\boldsymbol{w}_1|\lambda_1) p(\pi|a, b)$$

CP of π is easy and is proportional to $p(\pi|a, b) \times \prod_{n=1}^N p(z_n|\pi)$ which is simply a beta-Bernoulli model where the latent variables $z_1, z_2, \ldots, z_N$ are treated as N binary observations. Because of conjugacy, the CP will also be beta and given by $p(\pi|\mathbf{Z}) = \text{Beta}(\pi|a + \sum_{n=1}^N z_n, b + N - \sum_{n=1}^N z_n)$.

The CP of $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ will be identical to the CP of $\boldsymbol{w}$ in standard Bayesian linear regression except that CP of $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ will only depend on those training examples for which $z_n = 0$ and $z_n = 1$, respectively.

$$\begin{aligned} p(\boldsymbol{w}_0|\mathbf{Z}, \mathcal{D}, \theta) &= \mathcal{N}(\boldsymbol{w}_0|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \quad \text{where} \quad \boldsymbol{\Sigma}_0 = \left(\sum_{n:z_n=0} \beta_0 \boldsymbol{x}_n \boldsymbol{x}_n^\top + \lambda_0 \mathbf{I}\right)^{-1} \quad \text{and } \boldsymbol{\mu}_0 = \boldsymbol{\Sigma}_0 \sum_{n:z_n=0} \beta_0 y_n \boldsymbol{x}_n \\ p(\boldsymbol{w}_1|\mathbf{Z}, \mathcal{D}, \theta) &= \mathcal{N}(\boldsymbol{w}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \quad \text{where} \quad \boldsymbol{\Sigma}_1 = \left(\sum_{n:z_n=1} \beta_1 \boldsymbol{x}_n \boldsymbol{x}_n^\top + \lambda_1 \mathbf{I}\right)^{-1} \quad \text{and } \boldsymbol{\mu}_1 = \boldsymbol{\Sigma}_1 \sum_{n:z_n=1} \beta_1 y_n \boldsymbol{x}_n \end{aligned}$$

Name:

Roll No.:

Dept.:

Finally, the CP of z_n will be proportional to $p(z_n|\pi)p(y_n|z_n, \boldsymbol{x}_n, \boldsymbol{w}_0, \boldsymbol{w}_1, \beta_0, \beta_1)$. Since z_n is binary, let's calculate the posterior probability of $z_n = 1$, which is

$$p(z_n = 1|\mathcal{D}, \pi, \theta) \propto p(z_n = 1|\pi)p(y_n|z_n = 1, \boldsymbol{x}_n, \boldsymbol{w}_0, \boldsymbol{w}_1, \beta_0, \beta_1) = \pi \times \mathcal{N}(y_n|\boldsymbol{x}_n, \boldsymbol{w}_1, \beta_1)$$

Therefore $p(z_n = 1|\mathcal{D}, \pi, \theta) = \frac{\pi \times \mathcal{N}(y_n|\boldsymbol{x}_n, \boldsymbol{w}_1, \beta_1)}{\pi \times \mathcal{N}(y_n|\boldsymbol{x}_n, \boldsymbol{w}_1, \beta_1) + (1-\pi) \times \mathcal{N}(y_n|\boldsymbol{x}_n, \boldsymbol{w}_0, \beta_0)} = \hat{\mu}_n$.

This can also be succintly written as $p(z_n|\mathcal{D}, \pi, \theta) = \text{Bernoulli}(z_n|\hat{\mu}_n)$.

So we now have all three CPs required for our Gibbs sampler. The overall sketch of the Gibbs sampler be be written as

- Initialize $\mathbf{W}, \mathbf{Z}, \pi$ as $\mathbf{W}^{(0}, \mathbf{Z}^{(0)}, \pi^{(0)}$
- For Gibbs sampling iteration $t = 1, 2, \ldots, T$
  - Sample π from its CP (beta distribution) using the most recently sampled values of the other unknowns.
  - Sample $\boldsymbol{w}_0$ from its CP (Gaussian distribution) using the most recently sampled values of the other unknowns.
  - Sample $\boldsymbol{w}_1$ from its CP (Gaussian distribution) using the most recently sampled values of the other unknowns.
  - Sample the latent variables z_n, $n = 1, 2, \ldots, N$, from their respective CPs (Bernoulli distributions) using the most recently sampled values of the other unknowns.

Some distributions and their properties:

- For $x \in \mathbb{R}$, Univariate Gaussian: $\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$. For $x \in \mathbb{R}^D$, D-dimensional Gaussian: $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}$.

- For $x \in \{0, 1\}$, $\text{Bernoulli}(x|\pi) = \pi^x(1-\pi)^{1-x}$ where $\pi \in (0, 1)$. For $x_k \in \{0, N\}$ and $\sum_{k=1}^K x_k = N$, $\text{multinomial}(x_1, \ldots, x_K|N, \boldsymbol{\pi}) = \frac{N!}{\boldsymbol{x}_1!\ldots,x_K!}\pi_1^{x_1} \ldots \pi_K^{x_K}$ where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$, s.t. $\sum_{k=1}^K \pi_k = 1$. The multinoulli is the same as multinomial with $N = 1$.

- For $x \in (0, 1)$, $\text{Beta}(x|a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)}$, where $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ and Γ denotes the gamma function s.t. $\Gamma(x) = (x-1)!$ for a positive integer x. Expectation of a Beta r.v.: $\mathbb{E}[x] = \frac{a}{a+b}$.

Some other useful results:

- If $\boldsymbol{x} = \mathbf{A}\boldsymbol{z} + \boldsymbol{b} + \epsilon$, $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$, $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \mathbf{L}^{-1})$ then $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\mathbf{A}\boldsymbol{z} + \boldsymbol{b}, \mathbf{L}^{-1})$, $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mathbf{A}\boldsymbol{\mu} + \boldsymbol{b}, \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top + \mathbf{L}^{-1})$, and $p(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\Sigma}\left\{\mathbf{A}^\top\mathbf{L}(\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\right\}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^\top\mathbf{L}\mathbf{A})^{-1}$.