CS698X: Topics in Prob. Modeling and Inference *Your Name & Roll no:*

Mid-sem Exam (2017-2018, Sem-II)

Instructions:

- This question paper is worth a total of 100 marks.
- Total duration is 2.5 hours.
- Please write your name and roll number at the top of this sheet as well as on the provided answer copy.
- The **true-false** questions have to be answered on this question sheet itself, in the space provided. Please also submit the question sheet.
- To answer the other parts of the question paper, please use the provided answer copy.
- For rough work, please use pages at the back of answer copy. Extra sheets can be provided if needed.

Some distributions and their properties:

- 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.: $\mathbb{E}[x] = \frac{a}{a+b}$.
- For $x \in \{0,1,2,\ldots\}$ (non-negative integers), $\text{Poisson}(x|\lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}$ where λ is the rate parameter.
- For $x \in \mathbb{R}_+$, $\text{Gamma}(x|a,b) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx)$ (shape and rate parameterization)
- 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: $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}$.
  Trace-based representation: $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\Big\{-\frac{1}{2}\text{trace}\left[\boldsymbol{\Sigma}^{-1}\mathbf{S}\right]\Big\}$, $\mathbf{S} = (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top$.
- For $\boldsymbol{\pi} = [\pi_1,\ldots,\pi_K]$, s.t. $\sum_{k=1}^K \pi_k = 1$, $\text{Dirichlet}(\boldsymbol{\pi}|\alpha_1,\ldots,\alpha_K) = \frac{1}{B(\alpha_1,\ldots,\alpha_K)} \prod_{k=1}^K \pi_k^{\alpha_k-1}$ where $B(\alpha_1,\ldots,\alpha_K) = \frac{\prod_{k=1}^K \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^K \alpha_k)}$, and $\mathbb{E}[\pi_k] = \frac{\alpha_k}{\sum_{k=1}^K \alpha_k}$
- 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 $N = 1$.

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 $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^\top\mathbf{L}\mathbf{A})^{-1}$.
- Marginal and conditional distributions for Gaussians: $p(\boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_a,\boldsymbol{\Sigma}_{aa})$, $p(\boldsymbol{x}_a|\boldsymbol{x}_b) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_{a|b},\boldsymbol{\Sigma}_{a|b})$ where $\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Lambda}_{aa}^{-1} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}$, $\boldsymbol{\mu}_{a|b} = \boldsymbol{\Sigma}_{a|b}\left\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b)\right\} = \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\boldsymbol{x}_b - \boldsymbol{\mu}_b)$, where symbols have their usual meaning. :)
- $\frac{\partial}{\partial\boldsymbol{\mu}}[\boldsymbol{\mu}^\top\mathbf{A}\boldsymbol{\mu}] = [\mathbf{A}+\mathbf{A}^\top]\boldsymbol{\mu}$, $\frac{\partial}{\partial\mathbf{A}}\log|\mathbf{A}| = \mathbf{A}^{-\top}$, $\frac{\partial}{\partial\mathbf{A}}\text{trace}[\mathbf{A}\mathbf{B}] = \mathbf{B}^\top$
- For a random variable vector $\boldsymbol{x}$, $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\top + \text{cov}[\boldsymbol{x}]$

Questions begin from the next page

1 True-False Problems (10 questions, 10 marks)

1. [**T**] A likelihood distribution $p(x|\theta)$ does not necessarily sum/integrate to 1 over all possible values of the parameter θ.
2. [**F**] Assuming continuous-valued model parameters, the posterior predictive distribution can be seen as a uniformly weighted average of infinite many point predictive distributions.
3. [**T**] The log-likelihood $\log p(\mathbf{X}|\Theta)$ can never change in the E step of the EM algorithm.
4. [**T**] Marginal posterior and conditional posterior are defined in a lower dimensional parameter space as compared to a joint posterior.
5. [**F**] The computational cost of fully Bayesian inference is independent of the amount of data and only depends on the number of parameters and latent variables.
6. [**F**] When using the Bayes rule to find the expression for the posterior, the denominator in the Bayes rule expression (i.e., the marginal likelihood) can be ignored.
7. [**T**] In GP regression/classification, the GP prior does not depend on the data being modeled.
8. [**F**] Generative classification in general requires fewer parameters to estimate as compared to discriminative classification.
9. [**F**] Choosing the best model m based on the comparing the posterior probabilities of different models is equivalent to choosing the model with the largest marginal likelihood $p(\mathbf{X}|m)$.
10. [**T**] For probabilistic PCA with $p(\boldsymbol{x}_n|\boldsymbol{z}_n) = \mathcal{N}(\mathbf{W}\boldsymbol{z}_n, \sigma^2\mathbf{I}_D)$ with $p(\boldsymbol{z}_n) = \mathcal{N}(0, \mathbf{I}_K)$, the "spread" of latent variable's posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ does not depend on the input $\boldsymbol{x}_n$.

2 Short Answer Problems (10 questions, 30 marks)

1. Rank the three parameter estimation approaches, namely maximum-likelihood estimation (MLE), maximum-a-posteriori (MAP) estimation, and fully Bayesian inference, based on their robustness against overfitting, and give a brief justification for your ranking.

   Answer: The order will be (maximum prone to least prone)- MLE > MAP > Fully Bayesian inference. MLE: point estimate, no prior/regularization. MAP: Point estimate but incorporates prior/regularizer. Fully Bayesian inference: Full posterior distribution and posterior averaging at prediction time.

2. Suppose a probability distribution $p(\boldsymbol{x}|\theta)$ has a form proportional to $\exp(-\boldsymbol{x}^\top\mathbf{A}\boldsymbol{x} - \boldsymbol{b}^\top\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^D$. Identify this distribution (i.e., its name) and write down the expressions for its parameters θ.

   Answer: One quadratic and one linear term in x, so it must be a Gaussian $\mathcal{N}(\mu, \Sigma)$. Comparing with forms, we get $\Sigma = (2\mathbf{A})^{-1}$ and $\mu = -(2\mathbf{A})^{-1}\boldsymbol{b}$

3. Intuitively, which of the two distributions - plug-in predictive distribution and posterior predictive distribution - is expected to have a larger variance at a test point? Briefly justify your answer.

   Answer: Posterior predictive is expected to have a larger variance since it uses the entire posterior of the parameters to make predictive (thus taking into account their uncertainty while making prediction), rather than relying on a single value of the parameter to make prediction (as done by plug-in predictive)

4. Consider data with N training examples $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$, and $D \gg N$. Suppose we have learned two models - GP regression and Bayesian linear regression. For calculating the mean of the posterior predictive, which of the two models will be faster, in terms of big-O notation, and why?

   Answer: GP takes $O(N)$ time to compute the mean of the posterior predictive (predictive mean of the form $\sum_{n=1}^N \alpha_n k(\boldsymbol{x}_n, \boldsymbol{x}_*)$), whereas Bayesian linear regression takes $O(D)$ time (predictive mean of the form $\mu_N^\top\boldsymbol{x}_*$ where $\mu_N \in \mathbb{R}^D$ is the mean of the posterior). If $D \gg N$, GP prediction will be faster.

5. Give two examples where Laplace approximation may not give a good appropriation for a probability distribution. You may draw figures to illustrate.

   Answer: Multimodal, asymmetic shaped distributions, distributions with only positive support, etc.

6. Consider two Gaussians $\mathcal{N}(x|\mu_1, \Sigma_1)$ and $\mathcal{N}(x|\mu_2, \Sigma_2)$. Suppose we use the following rule to assign a binary label y to a point x: if $\mathcal{N}(x|\mu_1, \Sigma_1) \geq \mathcal{N}(x|\mu_2, \Sigma_2)$ then $y = +1$, otherwise $y = -1$.

   Under what condition we can express the above rule as $y = \text{sign}[f(x)]$ where $f(x)$ is a *linear* function of x (i.e., of the form $w^\top x$), and what will be the expression for $f(x)$ in such a case?

   Answer: Given the condition, we must have $\log \mathcal{N}(x|\mu_1, \Sigma_1) \geq \log \mathcal{N}(x|\mu_2, \Sigma_2)$. Expanding, we get $-0.5 \log |\Sigma_1^{-1}| - (x-\mu_1)^\top \Sigma_1^{-1}(x-\mu_1) > -0.5 \log |\Sigma_2^{-1}| - (x-\mu_2)^\top \Sigma_2^{-1}(x-\mu_2)$. The quadratic terms in x will vanish if $\Sigma_1 = \Sigma_2 = \Sigma$ (say), and we will be left with $w^\top x + b \geq 0$, where $w = \Sigma^{-1}(\mu_1 - \mu_2)$ and $b = 0.5(\mu_2^\top \Sigma^{-1} \mu_2 - \mu_1^\top \Sigma^{-1} \mu_1)$. Thus $f(x) = w^\top x + b$.

7. Briefly explain why marginal likelihood can be seen as a special case of a posterior predictive.

   Answer: Posterior predictive integrates out the likelihood over the posterior ($p(y_*|\boldsymbol{y}) = \int p(y_*|\theta) p(\theta|\boldsymbol{y}) d\theta$) and marginal likelihood does it over the prior ($p(y_*) = \int p(y_*|\theta) p(\theta) d\theta$). Therefore, marginal likelihood can be thought of as the posterior predictive for y_* with $N = 0$, i.e., no "training" data $\boldsymbol{y}$

8. When using EM for probabilistic PCA, can the dimensionality of the latent space be more than the data dimensionality? If yes, why? If no, why not?

   Answer: Yes. There is nothing in the PPCA model definition or its EM algorithm, that requires $K \leq D$ (unlike standard PCA which uses eigendecomposition and can therefore can't have $K > D$.

9. Briefly explain why, for EM, the M step of finding the optimal Θ is a convex problem, if the joint distribution of data and latent variables given parameters, i.e., $p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$ is in exponential family.

   Answer: Note that the joint distribution will be of the form $p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) = h(\boldsymbol{x}_n, \boldsymbol{z}_n) \exp(\Theta^\top \phi(\boldsymbol{x}_n, \boldsymbol{z}_n) - A(\Theta)$. The M step maximizes the expectation of the log of this quantity (summed over all observations). Thus (ignoring the expectation) the objective will be of the form $\Theta^\top \sum_{n=1}^N \phi(\boldsymbol{x}_n, \boldsymbol{z}_n) - NA(\Theta)$. Since $A(\Theta)$ is convex (and its negative concave) and the first term is linear, the overall objective is a concave function of Θ (and if we *minimize* it then the objective will be its negative, which is convex).

10. Consider a latent variable model where the prior over the latent variables is of the form $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})$, i.e., the distribution of $\boldsymbol{z}_n$ depends on the *value* of the latent variable of the previous observation. Assuming $\boldsymbol{z}_n$ to be discrete with K possibilities, which distribution would you use to model $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})$ and how many parameters would be needed to model this $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})$?

    Answer: Note that, since $\boldsymbol{z}_n$ is discrete with K possibilities, for each value (say ℓ) of $\boldsymbol{z}_{n-1}$) we can model $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1} = \ell)$ using a multinoulli with K dimensional probability vector $\boldsymbol{\pi}_\ell$. We will need K such probability vectors (one for each value of $\boldsymbol{z}_{n-1}$) and therefore will need total K^2 parameters.

3 Medium-Length Answer Problems (5 questions, 40 marks)

1. The KL divergence between two discrete distributions $p(x)$ and $q(x)$ with their support being the set of the non-negative integers is defined as $KL(p||q) = \sum_{x=0}^\infty p(x) \log \frac{p(x)}{q(x)}$. If $q(x)$ is the Poisson distribution, i.e., $q(x) = \frac{\lambda^x \exp(-\lambda)}{x!}$ then what value of λ will minimize the KL divergence?

   Answer: Minimizing the KL w.r.t. $q(x)$, which in turn depends on λ will give

   $$\hat{\lambda} = \arg\min_\lambda - \sum_{x=0}^\infty p(x) \log q(x) = \arg\min_\lambda - \sum_{x=0}^\infty p(x)[x \log \lambda - \lambda]$$

   Differentiating w.r.t. λ and setting it to zero gives $\hat{\lambda} = \sum_{x=0}^\infty x p(x) = \mathbb{E}[x]$. (note: you can check that the second derivative will be positive, so it must be the minimizer)

2. Express the Bernoulli $p(x|\theta) = \theta^x(1-\theta)^{1-x}$ in the exponential family form. Write down the expressions for its natural parameters, the sufficient statistics, and the log-partition function. Also show that the first derivative of its log-partition function is equal to the expected value of its sufficient statistics.

   Answer: Rewriting $p(x|\theta)$ as $\exp(\log p(x|\theta))$ and comparing with the form of an exponential family distribution $p(x|\eta) = h(x)\exp(\eta^\top \phi(x) - A(\eta)$, we get $\eta = \log \frac{\theta}{1-\theta}$ as the natural parameters, $\phi(x) = x$ as the sufficient statistics, and $A(\eta) = -\log(1-\theta) = \log(1+exp(\eta))$ as the log-partition function. The first derivative of $A(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)} = \theta = \mathbb{E}[x]$

3. Consider a Gaussian Mixture Model (GMM) with K components where the posterior probabilities of cluster memberships for a data point $\boldsymbol{x}_n$ are defined as

$$p(\boldsymbol{z}_n = k|\boldsymbol{x}_n) = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\mu_k, \sigma^2\mathbf{I})}{\sum_{\ell=1}^K \pi_\ell \mathcal{N}(\boldsymbol{x}|\mu_\ell, \sigma^2\mathbf{I})}$$

   where π_k is the mixing proportion for k^{th} Gaussian, μ_k is its mean, and $\sigma^2\mathbf{I}$ is its covariance matrix (assumed the same for all the Gaussians). Show that as σ^2 tends to zero, the EM algorithm for this GMM reduces to a K-means style algorithm. Also briefly summarize the steps of this resulting EM algorithm, e.g., what would each step compute, writing down the necessary expressions.

   Answer: The probability of $\boldsymbol{x}_n$ belonging to cluster k computed in E step is

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\mu_k, \sigma^2\mathbf{I})}{\sum_{\ell=1}^K \pi_\ell \mathcal{N}(\boldsymbol{x}|\mu_\ell, \sigma^2\mathbf{I})} = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{||\boldsymbol{x}_n - \mu_k||^2}{2\sigma^2})}{\sum_{\ell=1}^K \pi_\ell \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{||\boldsymbol{x}_n - \mu_\ell||^2}{2\sigma^2})}$$

   Now, as σ^2 tends to zero, out the K Gaussians in the denominator, the sum will be dominated by the Gaussian for which $||\boldsymbol{x}_n - \mu_\ell||^2$ is the smallest. In that case, the cluster membership probability γ_{nk} will to close to 1 for that Gaussian and close to 0 for all other Gaussians. Thus the vector $\gamma = [\gamma_{n1}, \ldots, \gamma_{nK}]$ will be like a one-hot vector and GMM will reduce to a hard clustering algorithm. The EM algorithm would alternate between the following steps:

   - E step: For each $n = 1, \ldots, N$, suppose $k_* = \arg\min_\ell ||\boldsymbol{x}_n - \mu_\ell||^2$. Set $\boldsymbol{z}_n = k_*$
   - M step: For each $k = 1, \ldots, K$, set μ_k to be the mean of all points assigned to cluster k

   Although you can also estimate π_k as $\pi_k = \frac{\sum_{n:\boldsymbol{z}_n = k} \boldsymbol{x}_n}{N}$, the EM algorithm doesn't require π_k.

   Note: This technique where we take a probabilistic model and let the variance in the likelihod term go to zero is known as "small-variance asymptotics" (SVA), and is a popular way to convert a probabilistic model into a similar non-probabilistic model that has a faster inference. It is not limited to Gaussian likelihoods where variance has a natural interpretation but works for any exponential family distribution. It is also used as a fast point estimation algo. for nonparametric Bayesian models.

4. Consider a mixture model $p(\boldsymbol{x}) = \sum_{k=1}^K \pi_k p(\boldsymbol{x}|k)$ where $p(\boldsymbol{x}|k)$ denotes the k^{th} component distribution of this mixture model and π_k denotes its mixing proportion. Assume each observation $\boldsymbol{x}$ to consist of two parts, i.e., $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$. Show that the conditional distribution $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$ can also be written in form of a mixture distribution and derive the expressions for the mixing proportions and the component distributions for this mixture distribution. (Hint: Note that you can write $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$ as the ratio of a joint distribution and a marginal distribution)

   Answer: Note that $p(\boldsymbol{x}) = p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{k=1}^K \pi_k p(\boldsymbol{x}_1, \boldsymbol{x}_2|k)$. Therefore

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2)}{p(\boldsymbol{x}_2)} = \frac{\sum_{k=1}^K \pi_k p(\boldsymbol{x}_1, \boldsymbol{x}_2|k)}{\int \sum_{\ell=1}^K \pi_\ell p(\boldsymbol{x}_1, \boldsymbol{x}_2|\ell) d\boldsymbol{x}_1} = \frac{\sum_{k=1}^K \pi_k p(\boldsymbol{x}_1, \boldsymbol{x}_2|k)}{\sum_{\ell=1}^K \pi_\ell p(\boldsymbol{x}_2|\ell)}$$

   Finally, we can always write $p(\boldsymbol{x}_1, \boldsymbol{x}_2|k)$ as $p(\boldsymbol{x}_1, \boldsymbol{x}_2|k) = p(\boldsymbol{x}_1|\boldsymbol{x}_2, k)p(\boldsymbol{x}_2|k)$, we have

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \frac{\sum_{k=1}^K \pi_k p(\boldsymbol{x}_1|\boldsymbol{x}_2, k) p(\boldsymbol{x}_2|k)}{\sum_{\ell=1}^K \pi_\ell p(\boldsymbol{x}_2|\ell)} = \sum_{k=1}^K \eta_k(\boldsymbol{x}_2) p(\boldsymbol{x}_1|\boldsymbol{x}_2, k)$$

   where $\eta_k(\boldsymbol{x}_2) = \frac{\pi_k p(\boldsymbol{x}_2|k)}{\sum_{\ell=1}^K \pi_\ell p(\boldsymbol{x}_2|\ell)}$ denotes the k^{th} mixing proportion and $p(\boldsymbol{x}_1|\boldsymbol{x}_2, k)$ the k^{th} component distribution. Note that the mixing proportion as well as the component distribution depend on $\boldsymbol{x}_2$.

5. Consider a Gaussian Process (GP) prior $p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathbf{K})$ where $\boldsymbol{f} = [f_1, \ldots, f_N]$ is the vector of scores of a function f on each of the N inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and $\mathbf{K}$ is the $N \times N$ kernel/covariance matrix with $K_{nm} = \kappa(\boldsymbol{x}_n, \boldsymbol{x}_m)$ being the pairwise similarity between inputs $\boldsymbol{x}_n$ and $\boldsymbol{x}_m$.

   Assume the following likelihood model for each response $p(y_n|f_n) = \mathcal{N}(y_n|f_n, \beta^{-1})$. Denote $\boldsymbol{y} = [y_1, \ldots, y_N]$ as the vector of all the responses and derive the expression for the posterior distribution $p(\boldsymbol{f}|\boldsymbol{y})$. Feel free to use properties of Gaussian distributions (don't derive from scratch).

   Now consider another case when each response $y_n \in \{0, 1\}$. What would be the an appropriate likelihood model in this case? Suggest a method for derive the posterior $p(\boldsymbol{f}|\boldsymbol{y})$ in this case. You don't need to give the entire derivation – a brief outline of how you would do it in this case is fine.

   Answer: The prior $p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathbf{K})$ and the likelihood $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma^2\mathbf{I})$. Since both are Gaussian, the posterior $p(\boldsymbol{f}|\boldsymbol{y})$ must also be Gaussian and is given by (using reverse conditioning)

$$p(\boldsymbol{f}|\boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathbf{K})\mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma^2\mathbf{I}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

   where $\boldsymbol{\Sigma} = (\mathbf{K}^{-1} + \frac{1}{\sigma^2}\mathbf{I})^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}[\frac{1}{\sigma^2}\boldsymbol{y}] = (\sigma^2\mathbf{K}^{-1} + \mathbf{I}_N)^{-1}\boldsymbol{y}$.

   For the case when y_n is binary, closed form posterior is not available. Since the problem is similar to logistic regression (though with GP prior), one way to approximate the posterior would be Laplace approximation. To do so, we can first find the MAP estimate of $\boldsymbol{f}$ by maximizing the log-posterior $p(\boldsymbol{f}|\boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathbf{K}) \prod_{n=1}^{N} \text{Bernoulli}(y_n|\sigma(f_n))$. We can also take compute the Hessian of the log-posterior. The MAP and Hessian can be used as the mean and covariance matrix of the Gausssian used in the Laplace approximation of the posterior $p(\boldsymbol{f}|\boldsymbol{y})$.

4 Long Answer Problem (1 question, 20 marks)

Consider a regression problem setting where we don't just want to learn a single regression model on a single dataset but instead wish to learn T regression models on T datasets $\{\mathcal{D}_t\}_{t=1}^{T}$. Let's call each regression problem a "task", so there are a total of T tasks to be learned. Assume that each dataset $\mathcal{D}_t = (\mathbf{X}^{(t)}, \boldsymbol{y}^{(t)})$ consists of N_t training examples $(\boldsymbol{x}_n^{(t)}, y_n^{(t)})_{n=1}^{N_t}$, where $\boldsymbol{x}_n^{(t)} \in \mathbb{R}^D$ and $y_n^{(t)} \in \mathbb{R}$.

Assume each dataset/task $\mathcal{D}_t$ to be modeled by a linear regression model with weight vector $\boldsymbol{w}_t \in \mathbb{R}^D$. For simplicity, assume the Gaussian noise precision to be known and the same across all tasks. Our goal will be to learn the T regression weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T$. However, instead of learning each weight vector $\boldsymbol{w}_t$ *independently* only using $\mathcal{D}_t$, we wish to learn all the weight vectors in a "collaborative" fashion, so that these tasks can "share information" with each other (which may be especially helpful if the amount of training data for each task is very small).

To do this, let us assume that each $\boldsymbol{w}_t$ can be written as a linear combination of K vectors $\mathbf{B} = [\boldsymbol{b}_1, \ldots \boldsymbol{b}_K]$, i.e., $\boldsymbol{w}_t = \mathbf{B}\boldsymbol{z}_t = \sum_{k=1}^{K} z_{tk}\boldsymbol{b}_k$ where each $\boldsymbol{b}_k \in \mathbb{R}^D$, $k = 1, \ldots, K$, and $\boldsymbol{z}_t \in \mathbb{R}^K$ are the weights of this linear combination for task t. Note that the $D \times K$ matrix $\mathbf{B} = [\boldsymbol{b}_1, \ldots \boldsymbol{b}_K]$ is shared by all the tasks (which helps sharing information across all the tasks) and $\boldsymbol{z}_t \in \mathbb{R}^K$ is specific to task t. As you can see, since $\boldsymbol{w}_t = \mathbf{B}\boldsymbol{z}_t$, learning $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T$ now reduces to learning $\mathbf{B}$ and $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T]$.

For simplicity, assume zero mean, identity covariance Gaussian priors on each $\boldsymbol{b}_k$, $k = 1, \ldots, K$ and each $\boldsymbol{z}_t$, $t = 1, \ldots, T$, so $p(\boldsymbol{b}_k) = \mathcal{N}(\boldsymbol{b}_k|0, \mathbf{I}_D)$ and $p(\boldsymbol{z}_t = \mathcal{N}(\boldsymbol{z}_t|0, \mathbf{I}_K)$.

You can think of this model as a multiparameter model with unknowns $\mathbf{B}$ and $\mathbf{Z}$ (treat all other hyperparameters as known), and can break the inference problem into inferring the conditional posteriors of each of the unknowns, i.e., $p(\boldsymbol{b}_k|\mathbf{B}_{-k}, \mathbf{Z}, \{\mathcal{D}_t\}_{t=1}^{T})$, $k = 1, \ldots, K$, and $p(\boldsymbol{z}_t|\mathbf{B}, \mathbf{Z}_{-t}, \{\mathcal{D}_t\}_{t=1}^{T})$, $t = 1, \ldots, T$.Here $\mathbf{B}_{-k}$ denotes $\mathbf{B}$ with $\boldsymbol{b}_k$ excluded. Likewise, $\mathbf{Z}_{-t}$ denotes $\mathbf{Z}$ with $\boldsymbol{z}_t$ excluded.

- Give the expressions for the conditional posteriors $p(\boldsymbol{b}_k|\mathbf{B}_{-k}, \mathbf{Z}, \{\mathcal{D}_t\}_{t=1}^{T})$ and $p(\boldsymbol{z}_t|\mathbf{B}, \mathbf{Z}_{-t}, \{\mathcal{D}_t\}_{t=1}^{T})$.
- Also, using the obtained expression/form of $p(\boldsymbol{z}_t|\mathbf{B}, \mathbf{Z}_{-t}, \{\mathcal{D}_t\}_{t=1}^{T})$, explain why inferring the conditional posterior for each $\boldsymbol{z}_t$ is equivalent to inferring the posterior of a Bayesian linear regression model using only $\mathcal{D}_t$, but with the inputs $\boldsymbol{x}_n^{(t)}$ projected to a K dimensional space via a projection matrix.

Hint: With the parametrization $\boldsymbol{w}_t = \mathbf{B}\boldsymbol{z}_t$ assumed in this problem, given $\mathbf{B}_{-k}$ and $\mathbf{Z}$, one can think of $\boldsymbol{b}_k$ as the weight vector of an appropriately defined Bayesian linear regression model. Likewise, given $\mathbf{B}$ and $\mathbf{Z}_{-t}$, one can think of $\boldsymbol{z}_t$ as the weight vector of an appropriately defined Bayesian linear regression problem.

Answer: We will be using an alternating inference scheme in which we will infer one unknown at a time, keeping all other fixed, and cycling through until we converge.

- Let's first consider inferring each $\boldsymbol{b}_k$. The likelihood model for task t can be written as $y_n^{(t)} = \boldsymbol{w}_t^\top \boldsymbol{x}_n^{(t)} + \epsilon_n^{(t)}$. Since $\boldsymbol{w}_t = \sum_{k=1}^K z_{tk}\boldsymbol{b}_k$, the likelihood can be written as

$$y_n^{(t)} = \boldsymbol{b}_k^\top (z_{tk}\boldsymbol{x}_n^{(t)}) + \sum_{\ell \neq k} \boldsymbol{b}_\ell^\top (z_{t\ell}\boldsymbol{x}_n) + \epsilon_n^{(t)}$$

  Given $\mathbf{Z}$ and $\mathbf{B}_{-k}$ the summation above is a constant we can rewrite the above as

$$y_n^{(t)} - \sum_{\ell \neq k} \boldsymbol{b}_\ell^\top (z_{t\ell}\boldsymbol{x}_n) = \boldsymbol{b}_k^\top (z_{tk}\boldsymbol{x}_n^{(t)}) + \epsilon_n^{(t)}$$

  Note that the above is a standard probabilistic regression model with a "modified" input $\hat{\boldsymbol{x}}_n^{(t)} = z_{tk}\boldsymbol{x}_n^{(t)}$ and a "modified" response $\hat{y}_n^{(t)} = y_n^{(t)} - \sum_{\ell \neq k} \boldsymbol{b}_\ell^\top (z_{t\ell}\boldsymbol{x}_n)$. We can therefore take the "modified" data $(\hat{\boldsymbol{x}}_n^{(t)}, \hat{y}_n^{(t)})$ from all the tasks $t = 1, \ldots, T$, and learn a Bayesian linear regression model. The conditional posterior $p(\boldsymbol{b}_k | \mathbf{B}_{-k}, \mathbf{Z}, \{\mathcal{D}_t\}_{t=1}^T)$ will have a closed form since the prior on $\boldsymbol{b}_k$ is Gaussian and the conditional likelihood for the modified version of the data is still Gaussian (note the equation above).

  Also note that $\boldsymbol{b}_k$ is shared by all the tasks, which is why the conditional posterior for $\boldsymbol{b}_k$ will indeed depend on all the data from all the tasks (as can be seen from the likelihood model above).

- Now, let's consider inferring $\boldsymbol{z}_t$. We can again rewrite the likelihood model as

$$y_n^{(t)} = (\mathbf{B}\boldsymbol{z}_t)^\top \boldsymbol{x}_n^{(t)} + \epsilon_n^{(t)} = \boldsymbol{z}_t^\top (\mathbf{B}^\top \boldsymbol{x}_n^{(t)}) + \epsilon_n^{(t)}$$

  Given $\mathbf{Z}_{-t}$ and $\mathbf{B}$, the above is again a probabilistic linear regression model with weight vector $\boldsymbol{z}_t$ and inputs $\mathbf{B}^\top \boldsymbol{x}_n^{(t)}$. Note that the above inputs are a K dimensional projection of the original inputs $\boldsymbol{x}_n^{(t)}$ by a projection matrix defined by the $D \times K$ matrix $\mathbf{B}$. The conditional posterior $p(\boldsymbol{z}_t | \mathbf{B}, \mathbf{Z}_{-t}, \{\mathcal{D}_t\}_{t=1}^T)$ will have a closed form since the prior on $\boldsymbol{z}_t$ is Gaussian and the conditional likelihood for the modified version of the data is still Gaussian (note the equation above).

  Note that since $\boldsymbol{z}_t$ is specific to task t only, the conditional posterior for $\boldsymbol{z}_t$ actually only depends on the data from task t , so the correct notation will be $p(\boldsymbol{z}_t | \mathbf{B}, \mathcal{D}_t)$; in fact, it also doesn't depend on $\mathbf{Z}_{-t}$ (as can be seen from the likelihood model above).