Your name: *Your Roll no.:*

Mid-Sem Exam (CS772A/CS698X: Probabilistic Machine Learning)

Duration: 3 hours, Total marks: 60

1 True/False Problems (10 marks, 1 mark each)

Please answer ALL of the following problems: Write T/F in the square bracket at the beginning of each problem. Please do not explain your answer as there are no partial credits for this part.

1. [**T**] A linear regression model with Gaussian prior $p(\boldsymbol{w}) = \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$ where the precision (inverse variance) λ is *set* to very large value would regularize $\boldsymbol{w}$ too much and may not be preferable.
2. [**F**] In binary classification using logistic regression, if the predicted label probability $p(y = 1|\boldsymbol{x}, \boldsymbol{w})$ is close to zero for some example $\boldsymbol{x}$, it implies that the prediction is ambiguous.
3. [**F**] Gaussian prior on the weight vector $\boldsymbol{w}$ allows us to get closed-form posterior in logistic regression.
4. [**F**] Any probability distribution can be represented as an exponential family distribution.
5. [**T**] In EM, the incomplete data likelihood $p(\mathbf{X}|\theta)$ can increase only in the M step.
6. [**T**] If a set of r.v.'s is jointly Gaussian then ***any*** finite subset of these r.v.'s is also jointly Gaussian.
7. [**F**] In a Gaussian Mixture Model, the *sum* of posterior probabilities γ_{nk} of all points $n = 1, \ldots, N$ belonging to a particular cluster k, is equal to one, i.e., $\sum_{n=1}^{N} \gamma_{nk} = 1$
8. [**F**] Unlike K-means, EM for Gaussian Mixture Model is not sensitive to initialization.
9. [**F**] Unlike PCA, which is a dimensionality *reduction* method, probabilistic PCA accomplishes the opposite goal, i.e., it *increases* the dimensionality of data.
10. [**T**] If we increase the amount of training data in a GP regression model, it will take longer to predict the response y_* for each test example $\boldsymbol{x}_*$.

2 Short Answer Problems (20 marks, 2 marks each)

Please answer ALL of the following problems (using 1-5 sentences; try to be short but precise):

1. Using a Gaussian prior $p(\boldsymbol{w}) = \mathcal{N}(0, \lambda^{-1}\mathbf{I}_D)$ on the weight vector in a linear/logistic regression model imposes ℓ_2 regularization (shrinkage) on $\boldsymbol{w} \in \mathbb{R}^D$. However, it assumes the degree of shrinkage to be the same for all components of $\boldsymbol{w}$. Suppose we want to impose component-specific shrinkage (to reflect that the D features are of unequal importances). Suggest a prior on $\boldsymbol{w}$ to accomplish this.

   Answer: $p(\boldsymbol{w}) = \mathcal{N}(0, \boldsymbol{\Psi})$ where $\boldsymbol{\Psi} = \text{diag}(\lambda_1^{-1}, \ldots, \lambda_D^{-1})$ is a diagonal matrix and the d-th diagonal element λ_d controls the extent of shrinkage on the d-th coefficient of $\boldsymbol{w}$. Another way to write the same would be $p(\boldsymbol{w}) = \prod_{d=1}^{D} \mathcal{N}(0, \lambda_d^{-1})$, i.e., a product of D univariate Gaussians with different precisions.

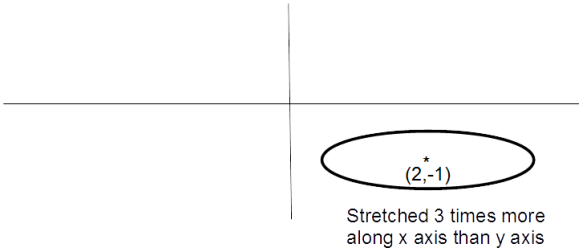
2. What's the benefit of MAP estimation over MLE? What's the benefit of fully Bayesian estimation over both MLE and MAP, in terms of predictive performance?

   Answer: MAP estimation allows us to imporporate a prior distribution on the parameters, which acts also as a regularizer. Consequently, the model is less prone to overfitting than MLE. In terms of predictive performance, the benefit of Bayesian estimation over MLE/MAP is its ability to make an "averaged" prediction by integrating the predictings over the posterior distribution of the parameters (i.e., using every parameter value to make predictions, appropriately weighting by that parameter's *posterior* probability). Therefore the predictions take into account the uncertainty in the estimated parameters, unlike MLE/MAP which only use a "point" (single best) estimate of the parameters.

3. Why (and in what situations) you would prefer soft-assignments of points to clusters when doing data clustering? Does GMM allow that? Does K-means?

   Answer: If we have overlapping clusters then soft-assignment would be especially useful because we don't have to assign a data point exclusively to a single cluster but can assign it "fractionally" to multiple clusters. GMM allows that whereas K-means doesn't.

4. Suppose we have a 2D Gaussian with mean $\boldsymbol{\mu} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$ and covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$. Sketch a contour (topo-map) of this distribution below. (I don't care about the scale of the covariance, so long as it is roughly properly shaped.)



5. Consider the following pdf for D dimensional data: $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}_D)$, where $\mathbf{W}$ is of size $D \times K$ and $K \ll D$. What benefit it offers, as opposed to modeling $p(\boldsymbol{x})$ using a general Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with no special structure on $\boldsymbol{\Sigma}$, and for what type of data it might be an ideal thing to do (assuming $K \ll D$)?

   Answer: Requires fewer parameters - $O(DK)$, as compared to the general Gaussian case when $\boldsymbol{\Sigma}$ requires $O(D^2)$ parameters. For data where each example is a high-dimensional vector with a lower intrinsic dimensionality, this may be very useful because we would require much fewer number of parameters to model such data, which can be especially desirable if we don't have enough examples.

6. Suppose $p(y|\boldsymbol{a}, \boldsymbol{x}, \beta) = \mathcal{N}(y|\boldsymbol{a}^\top\boldsymbol{x}, \beta^{-1})$ and $p(\boldsymbol{a}|\lambda) = \mathcal{N}(\boldsymbol{a}|\mathbf{0}, \lambda^{-1}\mathbf{I}_D)$, where $\boldsymbol{x}, \boldsymbol{a} \in \mathbb{R}^D$ and $y \in \mathbb{R}$. Write down the expression for the marginal distribution $p(y|\boldsymbol{x}, \beta, \lambda)$, i.e., the conditional distribution $p(y|\boldsymbol{a}, \boldsymbol{x}, \beta)$ averaged over $p(\boldsymbol{a}|\lambda)$.

   Answer: Marginal of a Gaussian w.r.t. a Gaussian is another Gaussian. Note that $p(y|\boldsymbol{a}, \boldsymbol{x}, \beta) = \mathcal{N}(y|\boldsymbol{a}^\top\boldsymbol{x}, \beta^{-1})$ is equivalent to $y = \boldsymbol{a}^\top\boldsymbol{x} + \epsilon = \boldsymbol{x}^\top\boldsymbol{a} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \beta^{-1})$. This is basically a transformation of a random variable $\boldsymbol{a}$ with mean 0 and covariance matrix $\lambda^{-1}\mathbf{I}$. From the transformation rule of random variables, $\mathbb{E}[y] = \mathbb{E}[\boldsymbol{x}^\top\boldsymbol{a} + \epsilon] = 0$ and $\text{var}(y) = \boldsymbol{x}^\top\text{cov}(\boldsymbol{a})\boldsymbol{x} + \beta^{-1} = \boldsymbol{x}^\top\lambda^{-1}\mathbf{I}\boldsymbol{x} + \beta^{-1} = \frac{1}{\lambda}\boldsymbol{x}^\top\boldsymbol{x} + \beta^{-1}$. Thus $p(y|\boldsymbol{x}, \beta) = \mathcal{N}(y|0, \lambda^{-1}\boldsymbol{x}^\top\boldsymbol{x} + \beta^{-1})$. Note: Many of you got it wrong by writing $\text{var}(y) = \lambda^{-1}\boldsymbol{a}^\top\boldsymbol{a} + \beta^{-1}$ which is perhaps the side-effect of referring to the slides too religiously during the exam and doing "pattern matching" (for $\mathbf{A}\boldsymbol{x} + \mathbf{b}$) without realizing that $p(y|\boldsymbol{x}, \beta)$ cannot depend on $\boldsymbol{a}$ because you are averaging over $p(\boldsymbol{a})$, and it's $\boldsymbol{x}$ that is fixed here. :)

7. Given an $N \times M$ partially observed ratings matrix $\mathbf{X}$ (N users, M movies), you apply a matrix factorization model $\mathbf{X} \approx \mathbf{U}\mathbf{V}^\top$ and learned latent factor matrices $\mathbf{U}$ ($N \times K$) and $\mathbf{V}$ ($M \times K$). Your task now is to recommend, for a given user (from the N users), the top-10 movies (from the M movies) he/she might be most interested in watching. How would you do that using $\mathbf{U}$ and $\mathbf{V}$?

   Answer: Note that $\mathbf{U}$ consists of user latent factors (each row is a user) and $\mathbf{V}$ consists of movie latent factors (each row is a movie). For the given user, let's say user n, we can compute the similarity of its latent factor $\boldsymbol{u}_n$ (row n in $\mathbf{U}$) with the latent factors of all the movies (latent factors given by the rows of $\mathbf{V}$ matrix) and find 10 most similar movies (e.g., based on the dot product $\boldsymbol{u}_n^\top\boldsymbol{v}_m$)

8. For logistic regression, which of the two methods - gradient descent and Newton's method - would converge faster in terms of the *number of iterations* taken to converge (and why)? Which of the two would be faster in terms of the *per-iteration run-time* (and why)?

   Answer: Newton's method would take fewer iterations (because it can utilize the curvature of the loss function to move to good directions) than gradient descent, but each Newton iteration is more expensive than gradient descent in terms of run-time because Newton's method require computing and inverting a Hessian matrix.

9. Exponential family distributions have sufficient statistics. Why is sufficient statistics particularly useful in parameter estimation for such distributions even when we have lots of data?

   Answer: Sufficient statistics are useful because they summarize everything that you need from the data in order to estimate the parameters of the distribution. Regardless of the amount of data, the size of the sufficient statistics will remain fixed, which is nice because it won't grow with more and more data, and you don't need to store the data but only the sufficient statistics.

10. In many probabilistic models, the unknowns in the model are of two types: local variables and global variables. How would you identify if a variable is local or global? Taking the example of Gaussian Mixture Model, what are the local variables and what are the global variables?

    Answer: Local variables are specific/tied to each data point, whereas global variables are shared by all data points (or at least a group of data points, in which they can be called semi-global). In a model, the number of local variable depends on the amount of data whereas the number of global variables is fixed. In a GMM, the cluster indicators $\boldsymbol{z}_n$'s are local variables, and all other variables ($\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$) are global variables.

3 Medium Answer Problems (20 marks, 5 marks each)

1. We have N observations $y_1, \ldots, y_N$ drawn i.i.d. from a univariate Gaussian, i.e., $y_n \sim \mathcal{N}(y_n|\mu, \beta^{-1})$, $\forall n = 1, \ldots, N$. where the scalar mean $\mu \in \mathbb{R}$ has a univariate Gaussian prior, i.e., $\mu \sim \mathcal{N}(\mu|\mu_0, \lambda^{-1})$. Assuming the hyperparameters β, μ_0, and λ to be known, compute the MAP estimate of μ and show that it is a convex combination of two terms: MLE estimate and prior's mean μ_0. For small enough N, what would happen to the MAP estimate when the prior's precision becomes very high ? Likewise, what would happen when the prior's precision becomes very small? **(3+1+1 marks)**.

   To do MAP estimation, we can write the log-joint probability of data and parameter which is nothing but the sum of the log-likelihood plus the log-prior. This can be written as

   $$\sum_{n=1}^{N} \log \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2}(y_n - \mu)^2\right) + \log \sqrt{\frac{\lambda}{2\pi}} \exp\left(-\frac{\lambda}{2}(\mu - \mu_0)^2\right)$$

   Taking derivative and setting it to zero, we get

   $$\mu_{MAP} = \frac{N\beta\bar{y} + \lambda\mu_0}{N\beta + \lambda} = \frac{N\beta}{N\beta + \lambda}\bar{y} + \frac{\lambda}{N\beta + \lambda}\mu_0$$

   where $\bar{y} = \sum_{n=1}^{N} y_n/N$ is simply the empirical mean of the data which, for the univariate Gaussian case, is the MLE of the mean of the Gaussian. The expression above is clearly a convex combination of empirical mean/MLE estimate $\bar{y}$ and the prior mean μ_0. When λ is very small, the second term will vanish and the MAP estimate will reduce to $\bar{y}$, i.e., the MLE estimate. When λ is very large, the second term will dominate and will simply be μ_0 (the prior's mean)

2. In this problem, we will consider another classic parameter estimation method called the **Method of Moments** (MoM) estimator. Basically, for a probability distribution with K parameters $\theta_1, \theta_2, \ldots, \theta_K$, given a set of N observations $X_1, X_2, \ldots, X_N$ from that distribution, the MoM matches the first

K moments of this distribution to the corresponding *empirical* moments (computed using data), to estimate the K parameters of the distribution.

For univariate Gaussian $\mathcal{N}(\mu, \sigma^2)$ which has two parameters μ and σ^2, we would have to match the first and the second moments ($\mathbb{E}[X]$ and $\mathbb{E}[X^2]$) to their corresponding empirical estimates. Suppose we are given N observations $X_1, X_2, \ldots, X_N$, each drawn i.i.d. from $\mathcal{N}(\mu, \sigma^2)$. Use the method of moments to estimate the parameters μ and σ^2. How does this compare to MLE estimates? **(4+1 marks)**.

For the univariate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, the first and the second moments are $\mathbb{E}[X] = \mu$ and $\mathbb{E}[X^2] = \mathbb{E}[X]^2 + \text{var}(X) = \mu^2 + \sigma^2$, respectively. Matching these moments with the corresponding empirical moments, we will have $\frac{\sum_{n=1}^N X_n}{N} = \mu$ and $\frac{\sum_{n=1}^N X_n^2}{N} = \mu^2 + \sigma^2$. The first relation immediately gives us μ which is the same as the MLE estimate of Gaussian's mean. Also using the second relation and further simplifying we get $\sigma^2 = \frac{\sum_{n=1}^N (X_n - \mu)^2}{N}$, which is again equal to the MLE estimate of the Gaussian's variance.

3. Recall that the solution to regularized linear regression can be written as $\boldsymbol{w} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_D)^{-1}\mathbf{X}^\top\boldsymbol{y}$, where $\mathbf{X}$ is the $N \times D$ feature matrix and $\boldsymbol{y}$ is the $N \times 1$ response vector and λ is the regularization coefficient. Show that the above solution can be re-expressed as $\boldsymbol{w} = \mathbf{X}^\top\boldsymbol{\alpha}$ where $\boldsymbol{\alpha} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}_N)^{-1}\boldsymbol{y}$. If needed, you may make use of the matrix inversion lemma: $(\mathbf{P}^{-1} + \mathbf{B}^\top\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{R}^{-1} = \mathbf{P}\mathbf{B}^\top(\mathbf{B}\mathbf{P}\mathbf{B}^\top + \mathbf{R})^{-1}$ **(3 marks)**

Comparing $(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_D)^{-1}\mathbf{X}^\top$ with $(\mathbf{P}^{-1}+\mathbf{B}^\top\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{R}^{-1}$, we immediately identify that $\mathbf{B} = \mathbf{X}$, $\mathbf{R} = \mathbf{I}_N$, and $\mathbf{P} = \lambda^{-1}\mathbf{I}_D$. Susbstituting these into $\mathbf{P}\mathbf{B}^\top(\mathbf{B}\mathbf{P}\mathbf{B}^\top + \mathbf{R})^{-1} = \lambda^{-1}\mathbf{X}^\top(\mathbf{X}^\top\lambda^{-1}\mathbf{I}_D\mathbf{X} + \mathbf{I}_N)^{-1}$, which further simplifies to $\mathbf{X}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_N)^{-1}$. Thus $\boldsymbol{w} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}_N)^{-1}\boldsymbol{y} = \mathbf{X}^\top\boldsymbol{\alpha}$

Note that we have now written $\boldsymbol{w}$ as $\boldsymbol{w} = \mathbf{X}^\top\boldsymbol{\alpha}$. Now, consider the expression $\boldsymbol{\alpha} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}_N)^{-1}\boldsymbol{y}$. Here, what does each entry of the $N \times N$ matrix $\mathbf{X}\mathbf{X}^\top$ represent? It turns out that a small change to this would turn this into a nonlinear regression model. What would be that change? **(2 marks)**

Each entry (i, j) of the $N \times N$ matrix $\mathbf{X}\mathbf{X}^\top$ represents the inner-product (*linear/Euclidean* similaity) between a pair of examples i and i, i.e., $\boldsymbol{x}_i^\top\boldsymbol{x}_j$. To make the model nonlinear, we can replace $\boldsymbol{x}_i^\top\boldsymbol{x}_j$ by a nonlinear similarity (kernel) function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (which would mean replacing $\mathbf{X}\mathbf{X}^\top$ by a kernel matrix $\mathbf{K}$ where $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which now looks like what we had in the GP regression case.

4. Prove the EM identity $\log p(\mathbf{X}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \text{KL}(q||p_z)$ **(3 marks)**

$$\begin{aligned}\mathcal{L}(q, \theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log\left\{\frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}\right\} \\ \text{KL}(q||p_z) &= -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log\left\{\frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})}\right\}\end{aligned}$$

We can use $p(\mathbf{X}, \mathbf{Z}|\theta) = p(\mathbf{Z}|\mathbf{X}, \theta) + p(\mathbf{X}|\theta)$. Plugging this into the expression of $\mathcal{L}(q, \theta)$, expanding, and using the fact that $\sum_{\mathbf{Z}} q(\mathbf{Z}) = 1$, we get the identity.

How $\mathcal{L}(q, \boldsymbol{\theta})$ behaves in E and M steps (w.r.t. the previous M and E steps, respectively). Does it increase/decrease/stay the same? Also, how $\text{KL}(q||p_z)$ behaves in each of these steps? **(2 marks)**

$\mathcal{L}(q, \boldsymbol{\theta})$ increases in both steps. $KL(q||p)$ decreases (to zero) in the E step and increases in the M step.

4 Long Answer Problems (10 marks)

Please answer ONE of the following problems. Be sure to show all the steps in your solution as individual steps will have partial credits (write your solution on the provided extra sheets):

1. Consider the standard probabilistic linear regression model $y_n = \boldsymbol{w}^\top\boldsymbol{x}_n + \epsilon_n$ with Gaussian noise $\epsilon_n \sim \mathcal{N}(0, \beta^{-1})$ where the noise is independent of $\boldsymbol{w}$. Assume the following Gaussian prior on the

weight vector: $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$. Denote the $N \times D$ feature matrix $\mathbf{X} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_N^\top]^\top$ and the $N \times 1$ response vector $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$. The responses $\boldsymbol{y}$ are assumed i.i.d. given $\boldsymbol{w}$ and features $\mathbf{X}$.

(a) Write down the joint conditional distribution of all the responses, i.e., $p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta)$. **(1 mark)**

$p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta) = \mathcal{N}(\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I})$

Some of you have written $p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta)$ as a product of N likelihoods (one per data point) which is also okay, although I was expecting an answer in the multivariate normal form (as above).

(b) Show that the marginal distribution $p(\boldsymbol{y}|\mathbf{X}, \beta)$ of the responses $\boldsymbol{y}$ will be a Gaussian, i.e., $p(\boldsymbol{y}|\mathbf{X}, \beta) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$, by averaging $p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \sigma^2)$ over $p(\boldsymbol{w})$. Write down the expressions for both $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$. **(2 marks)**

We know that the marginal will be a Gaussian. Note that $p(\boldsymbol{y}|\boldsymbol{w}, \mathbf{X}, \beta) = \mathcal{N}(\mathbf{X}\boldsymbol{w}, \beta^{-1}\mathbf{I})$ which is equivalent to writing $\boldsymbol{y} = \mathbf{X}\boldsymbol{w} + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \beta^{-1}\mathbf{I})$. Since $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$, using the transformation rule (here $\boldsymbol{w}$ transformed by $\mathbf{X}$), $p(\boldsymbol{y}|\mathbf{X}, \beta)$ will be a Gaussian with mean $\mu_y = \mathbb{E}[\mathbf{X}\boldsymbol{w} + \epsilon] = \mathbf{X}\mathbb{E}[\boldsymbol{w}] + 0 = \mathbf{X}\mu_w$ and covariance $\Sigma_y = \mathbf{X}\text{cov}(\boldsymbol{w})\mathbf{X}^\top + \beta^{-1}\mathbf{I} = \mathbf{X}\Sigma_w\mathbf{X}^\top + \beta^{-1}\mathbf{I}$

(c) Using $p(\boldsymbol{w})$ and $p(\boldsymbol{y}|\mathbf{X}, \beta)$, now write down the joint distribution of $\boldsymbol{w}$ and $\boldsymbol{y}$. Hint: From the property of Gaussians, it will again be another Gaussian of the form

$$p\left(\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_w \\ \boldsymbol{\mu}_{\boldsymbol{y}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_w & \boldsymbol{\Sigma}_{wy} \\ \boldsymbol{\Sigma}_{yw} & \boldsymbol{\Sigma}_y \end{bmatrix}\right)$$

where $\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w$ are known, $\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y$ will be given by the solution to (b). The only extra term you need is $\boldsymbol{\Sigma}_{wy} = \boldsymbol{\Sigma}_{yw}^\top = \text{cov}(\boldsymbol{w}, \boldsymbol{y})$, so you will need to compute this covariance. If it helps your calculations, recall that the vector of responses $\boldsymbol{y}$ can be written as $\boldsymbol{y} = \mathbf{X}\boldsymbol{w} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_N]^\top$ is an $N \times 1$ noise vector. **(4 marks)**

$\boldsymbol{\Sigma}_{wy} = \boldsymbol{\Sigma}_{yw}^\top = \text{cov}(\boldsymbol{w}, \boldsymbol{y}) = \text{cov}(\boldsymbol{w}, \mathbf{X}\boldsymbol{w} + \epsilon) = \text{cov}(\boldsymbol{w}, \mathbf{X}\boldsymbol{w}) = \text{cov}(\boldsymbol{w}, \boldsymbol{w})\mathbf{X}^\top = \Sigma_w\mathbf{X}^\top$. Note that we have used the fact that the noise vector ϵ is independent of $\boldsymbol{w}$. The joint distribution will be

$$p\left(\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{y} \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_w \\ \mathbf{X}\boldsymbol{\mu}_w \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_w & \Sigma_w\mathbf{X}^\top \\ \mathbf{X}\Sigma_w & \mathbf{X}\Sigma_w\mathbf{X}^\top + \beta^{-1}\mathbf{I} \end{bmatrix}\right)$$

(d) Using the joint distribution obtained from part (c) and the formula of Gaussian conditional/posterior distribution given the joint distribution, derive the posterior $p(\boldsymbol{w}|\boldsymbol{y})$. **(3 marks)**

The posterior will be a Gaussian $\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean

$$\boldsymbol{\mu} = \boldsymbol{\mu}_w + \Sigma_{wy}\Sigma_y^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y) = \boldsymbol{\mu}_w + \Sigma_w\mathbf{X}^\top(\mathbf{X}\Sigma_w\mathbf{X}^\top + \beta^{-1}\mathbf{I})^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y)$$

and covariance matrix

$$\Sigma = \Sigma_w - \Sigma_{wy}\Sigma_y^{-1}\Sigma_{yw} = \Sigma_w - \Sigma_w\mathbf{X}^\top(\mathbf{X}\Sigma_w\mathbf{X}^\top + \beta^{-1}\mathbf{I})^{-1}\mathbf{X}\Sigma_w$$

2. **Supervised Probabilistic PCA:** Consider a *supervised* version of the PPCA model where, in addition to the inputs $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ with each $\boldsymbol{x}_n \in \mathbb{R}^D$, you are also given the corresponding responses $\{y_1, \ldots, y_N\}$. Assume each response to be real-valued (as in linear regression), i.e., $y \in \mathbb{R}$. Recall that PPCA assumes $\boldsymbol{x} \in \mathbb{R}^D$ to be generared from a lower-dimensional $\boldsymbol{z} \in \mathbb{R}^K$: $\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n$

In the supervised variant of PPCA, we assume that $\boldsymbol{z}$ generates the input $\boldsymbol{x}$ as well as the response y:

$$\boldsymbol{x}_n = \mathbf{W}_x\boldsymbol{z}_n + \epsilon_n^{(x)} \quad \text{where } \epsilon_n^{(x)} \sim \mathcal{N}(0, \sigma_x^2\mathbf{I}_D) \tag{1}$$

$$y_n = \boldsymbol{w}_y^\top\boldsymbol{z}_n + \epsilon_n^{(y)} \quad \text{where } \epsilon_n^{(y)} \sim \mathcal{N}(0, \sigma_y^2) \tag{2}$$

where $\mathbf{W}_x$ is a $D \times K$ matrix and $\boldsymbol{w}_y$ is a $K \times 1$ vector. One way to think of this would be to treat it as a combination of two models: (1) $\boldsymbol{z}$ to $\boldsymbol{x}$ mapping, which is basically a PPCA model; and (2)

$\boldsymbol{z}$ to y mapping which is like a linear regression model. Also note that $\boldsymbol{x}_n$ and y_n are conditionallly independent, given $\boldsymbol{z}_n$.

Your goal is to estimate the posterior distribution over $\boldsymbol{z}_n$, i.e., $p(\boldsymbol{z}_n|\boldsymbol{x}_n, y_n)$. To (considerably :)) simplify things, let us assume that $\mathbf{W}_x$, $\boldsymbol{w}_y$, σ_x, and σ_y are all given to you (e.g., estimated from the previous M step of an EM algorithm for this model).

Hint: There are several ways to do this. One way (due to the conditional independence of $\boldsymbol{x}$ and $\boldsymbol{y}$) is to break the estimation into two steps: (a) First estimate the posterior over $\boldsymbol{z}_n$ assuming only the PPCA model (1). For this, you can just write down the expression of the posterior of $\boldsymbol{z}_n$ from the PPCA model (1) (no need to derive it); (b) Now use this posterior as a prior over $\boldsymbol{z}_n$ for the second regression-like model (2) with a *single* observation y_n, $\boldsymbol{w}_y$ known as well, and $\boldsymbol{z}_n$ being the unknown "weight vector" whose posterior you have to estimate.

There are, of course, other (perhaps even simpler :)) ways to solve this once you have made the above-mentioned assumption that $\mathbf{W}_x$, $\boldsymbol{w}_y$, σ_x, and σ_y are all given. You are free to use any.

(**Note:** There will be partial credits for this problem as well. It's just that the break-up will depend on the approach you take and on how close you get to the right solution).

Let's first write the posterior of $\boldsymbol{z}_n$ conditioned only on $\boldsymbol{x}_n$. This is given by the usual PPCA posterior for $\boldsymbol{z}_n$, i.e., $p(\boldsymbol{z}_n|\boldsymbol{x}_n, \mathbf{W}_x) = \mathcal{N}(\boldsymbol{z}_n|\mathbf{M}^{-1}\mathbf{W}_x^\top \boldsymbol{x}_n, \sigma_x^2\mathbf{M}^{-1})$ where $\mathbf{M} = \mathbf{W}_x^\top\mathbf{W}_x + \sigma_x^2\mathbf{I}$.

Now consider the second equation. Structurally, this is like a regression model:

$$y_n = \boldsymbol{w}_y^\top \boldsymbol{z}_n + \epsilon_n^{(y)} = \boldsymbol{z}_n^\top \boldsymbol{w}_y + \epsilon_n^{(y)}$$

where w_y acts an input, y_n as a response, and $\boldsymbol{z}_n$ as the regression weight vector with the new prior $p(\boldsymbol{z}_n|\boldsymbol{x}_n, \mathbf{W}_x) = \mathcal{N}(\boldsymbol{z}_n|\mathbf{M}^{-1}\mathbf{W}_x^\top \boldsymbol{x}_n, \sigma_x^2\mathbf{M}^{-1})$ and the likelihood model $p(y_n|\boldsymbol{w}_y, \boldsymbol{z}_n) = \mathcal{N}(y_n|\boldsymbol{z}_n^\top \boldsymbol{w}_y, \sigma_y^2)$. The posterior will be another Gaussian

$$p(\boldsymbol{z}_n|\boldsymbol{x}_n, y_n, \mathbf{W}_x, \boldsymbol{w}_y) = \mathcal{N}(\boldsymbol{z}_n|\mu_z, \Sigma_z)$$

where, using the Gaussian posterior update formula,

$$\begin{aligned}
\Sigma_z &= (\frac{1}{\sigma_x^2}\mathbf{M} + \frac{1}{\sigma_y^2}\boldsymbol{w}_y\boldsymbol{w}_y^\top)^{-1} = (\frac{1}{\sigma_x^2}\mathbf{W}_x^\top\mathbf{W}_x + \frac{1}{\sigma_y^2}\boldsymbol{w}_y\boldsymbol{w}_y^\top + +\mathbf{I}_K)^{-1} \\
\mu_z &= \Sigma_z(\frac{1}{\sigma_y^2}\boldsymbol{w}_y y_n + \frac{1}{\sigma_x^2}\mathbf{M}\mathbf{M}^{-1}\mathbf{W}_x^\top \boldsymbol{x}_n) = \Sigma_z(\frac{1}{\sigma_y^2}\boldsymbol{w}_y y_n + \frac{1}{\sigma_x^2}\mathbf{W}_x^\top \boldsymbol{x}_n)
\end{aligned}$$

Note: An alternative method would be to write the joint likelihood of $\boldsymbol{x}_n, y_n$ which will again be a Gaussian with mean $\begin{bmatrix} \mathbf{W}_x\boldsymbol{z}_n \\ w_y^\top \boldsymbol{z}_n \end{bmatrix}$ and a block diagonal covariance matrix due to conditional independence of $\boldsymbol{x}_n$ and y_n given by $\begin{bmatrix} \sigma_x^2\mathbf{I}_D & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$, and then because the prior on $\boldsymbol{z}_n$ is Gaussian as well, the posterior can be obtained easily using Gaussian posterior formula.