CS772A: Probabilistic Machine Learning *Your Name & Roll no:*

End-sem Exam (2017-2018, Sem-I)

Instructions:

- This question paper is worth a total of 100 marks.
- Total duration is 3.5 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 and **fill-in-the-blank** 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**] It is possible to perform fully Bayesian inference even if we are using a uniform prior distribution.
2. [**F**] Importance sampling enables us to generate samples from a difficult-to-sample distribution $p(z)$ by drawing samples from a simpler proposal distribution $q(z)$ and associating a weight to each sample.
(No. Importance sampling is a method for computing an intractable expectation.)
3. [**T**] Inference using Metropolis-Hastings sampling does not require the model to have conjugacy.
4. [**F**] GLMs, which use exponential family distributions to model the response's distribution, are appealing because posterior inference is easy in these models due to conjugacy.
(If the prior on the weight vector $\boldsymbol{w}$ is Gaussian then, apart from Gaussian GLM (real-valued response), other GLMs have likelihoods that are not conjugate to the Gaussian prior.)
5. [**F**] When using a Gaussian Process prior over the function f in supervised learning problems (e.g., regression/classification), we don't need to compute the posterior over f and can compute the posterior predictive $p(y_*|\boldsymbol{y})$ directly.
(The posterior computation can only be bypassed in the GP regression case. For the classification case, we need to compute the posterior (actually approximate it since the model is no longer conjugate, just like logistic regression case).)
6. [**T**] Kalman smoothing is computationally more expensive as compared to Kalman filtering.
7. [**F**] In a deep latent Gaussian model, all layers are modeled using Gaussian distributions.
(Only the hidden layers are modeled using Gaussians. The last (data) layer can be modeled using any distribution depending on the type of data.)
8. [**T**] Stochastic variational inference (SVI) only makes sense (i.e., is of practical use) for models that consist of both local and global variables.
9. [**F**] Active Learning is based on querying the labels of those unlabeled examples whose predicted label distribution has the smallest entropy.
(Not those with smallest but with largest entropy (i.e., on which the model is most uncertain about).
10. [**T**] An HMM cannot be used if the latent variables underlying the observed sequence are real-valued.

2 Fill-in-the-blanks (5 questions, 5 marks)

1. Assume you are modeling count-valued observations $y_1, \ldots, y_N$ using a Poisson with rate parameter λ. The name of distribution that can be used as a conjugate prior for λ is Gamma.
2. Given random variables X and Y drawn from Uniform$(-1, 1)$, the value of $\mathbb{E}\left[\mathbb{I}[|X| + |Y| \leq 1]\right]$, where $\mathbb{I}[.]$ denotes the indicator function, is equal to 2/4=0.5. (hint: Think geometrically)
3. Name of the quantity maximized in the M step of EM is expected complete data log-likelihood.
4. Name of the quantity that MLE-II maximizes w.r.t. hyperparameters marginal likelihood.
5. Total number of unique global parameters (each scalar-value counted as one parameter) in a K component, D dim. GMM, with each Gaussian having a full covariance = $K + KD + KD(D+1)/2$

3 Short Answer Problems (10 questions, 30 marks)

1. Consider a likelihood model $p(x|\theta) = \mathcal{N}(x|\theta, \sigma^2)$ with σ^2 known, and assume a prior distribution $p(\theta) = \exp(A\theta^2 + B\theta + C)$. Is $p(\theta)$ conjugate to the likelihood $p(x|\theta)$? If yes, why? If no, why not?
Ans: Note that the prior $p(\theta)$ is actually Gaussian, which is conjugate to the likelihood (also Gaussian).

2. Briefly describe a scheme to sample from a bivariate Gaussian, assuming that you only have access to a function that can generate samples from a univariate Gaussian.
Ans: Use Gibbs sampling to draw from $p(x_1|x_2)$ and $p(x_2|x_1)$, both of which are univariate Gaussians. These are simple Gaussian conditional distributions whose expressions are easy to obtain from the joint bivariate Gaussian $p(x_1, x_2)$. **Note:** Some students have drawn $[z_1, z_2]$ from two zero mean Gaussians and then used $\boldsymbol{x} = \mu + L[z_1, z_2]^\top$ where μ is the mean and L is a Cholesky decomposition of the covariance matrix Σ of the target Gaussian distribution. While this will also give samples from the desired two-dim Gaussian, it requires another function to first compute Cholesky decomposition!

3. Briefly describe why Expectation Maximization (EM) can be seen as a special case of variational Bayesian (VB) inference. In EM, what is the analogue of the VB evidence lower bound (ELBO)?
Ans: EM only estimates the posterior distribution of latent variables and finds point estimate of parameters. VB treats everything as latent variable and estimates the (approximate) posterior of all these latent variables. The ELBO $\mathcal{L}(q)$ in VB corresponds to the $\mathcal{L}(q, \Theta)$ function of EM which depends on latent variables as well as parameters, which is maximized in two steps - once w.r.t. q with Θ fixed and then vice-versa. In contrast, VB only involves a maximization of the ELBO $\mathcal{L}(q)$ w.r.t. q.

4. Briefly describe how the posterior inference for latent variables, i.e., finding $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$, in a variational autoencoder (VAE) model is different from the traditional approach of inferring $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ using MCMC or variational inference methods, and what are the advantages of the VAE approach?
Ans: Instead of iteratively inferring the posterior $p(\boldsymbol{z}_n|\boldsymbol{x}_n)$ for each data point as done by MCMC/VB, VAE learns a single feedforward network defined by global (not data-point specific) parameters ϕ to estimate the posterior $q_\phi(\boldsymbol{z}_n|\boldsymbol{x}_n)$. Once learned, inferring the posterior over the latent variable $\boldsymbol{z}_*$ for any data point $\boldsymbol{x}_*$ doesn't require an iterative procedure and can thus be done in a fast manner - the parameters of the posterior $q_\phi(\boldsymbol{z}_*|\boldsymbol{x}_*)$. can be quickly obtained by pushing $\boldsymbol{x}_*$ through the neural net.

5. For an $N \times K$ binary matrix $\mathbf{Z}$ with $z_{nk} \sim \text{Bernoulli}(\pi_k)$, $\pi_k \sim \text{Beta}(\alpha/K, 1)$, what is the expected number of nonzero entries in $\mathbf{Z}$ as $K \longrightarrow \infty$?
Ans: This will be NK times the expectation of each binary entry z_{nk}, i.e., $NK \times \mathbb{E}[z_{nk}]$. Note that $\mathbb{E}[z_{nk}]$ after integrating out π_k is simply $\frac{\alpha/K}{\alpha/K+1}$ (expectation of beta r.v. π_k). This the expected number of nonzero entries is $\frac{N\alpha}{\alpha/K+1}$, which as $K \to \infty$ is $N\alpha$.

6. Briefly describe what collapsing means in the context of Bayesian inference? Why is it useful?
Ans: Collapsing refers to integrating out some variables from the model. This results in fewer unknowns to be estimated and can often be desirable if we don't actually care about or are not interested in some "nuisance" variables in the model (which are primarily used to define the overall model, but may not be of individual interest per se). If desired, often it is possible to estimate these using the other variables that we have done inference on. Collapsing often leads to faster convergence of inference methods since the inference method has to "explore" less.

7. Suppose you are using Metropolis-Hastings (MH) sampling to draw samples from the posterior distribution of a logistic regression model. Does the computational cost of computing the MH acceptance probability depend on the size of the dataset? If yes, why? If no, why not?
Ans: MH does need to evaluate the (unnormalized) posterior during evaluation of acceptance probability. The unnormalized posterior clearly depends on the likelihood which in turn depends on all the data. Therefore computing the acceptance probability does depend on the size of the dataset.

8. Why do MCMC sampling algorithms typically keep a burn-in phase, and after burn-in why don't we keep every sample but discard a few samples between two collected samples?
Ans: We typically wait until the burn-in period to ensure that the Markov chain has reached near the

actual target distribution (so that the samples we are collecting are from that distribution). We discard samples between every two collected samples to ensure that the collected samples are uncorrelated (recall that consecutive MCMC samples are correlated).

9. Consider the information form for a D-dimensional Gaussian $p(\boldsymbol{x}|\boldsymbol{\xi}, \boldsymbol{\Lambda}) \propto \exp[\boldsymbol{\xi}^\top \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Lambda}\boldsymbol{x}]$, where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$. Show that we can express this Gaussian in form of a Markov Random Field (MRF) consisting of D nodes. Identify the clique potential functions of this MRF.
Ans: We can expand and write $p(\boldsymbol{x}|\boldsymbol{\xi}, \boldsymbol{\Lambda}) \propto \exp[\sum_{i=1}^{D} \xi_i x_i - \frac{1}{2}\sum_{i=1}^{D}\sum_{j=1}^{D} \Lambda_{ij} x_i x_j]$, which can be written as a product of per-node clique potentials $\psi(x_i) = \exp(\xi_i x_i)$ and pairwise clique potentials $\psi(x_i, x_j) = \exp(-\Lambda_{ij} x_i x_j)$. **Note:** Depending on how you define the cliques, there can be other ways of defining the clique potentials (e.g., if you consider the entire graph as a clique then the clique potential will just be $\psi(\boldsymbol{x}) = \exp[\boldsymbol{\xi}^\top \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Lambda}\boldsymbol{x}]$.

10. What are local and global variables in a probabilistic latent variable model? In the Latent Dirichlet Allocation (LDA) model, what are the local and what are the global variables?
Ans: Local variables are data point specific (one for each data points) and global variables are shared by all data points. In LDA, mixture proportion vector θ_d of each document and the topic assignments $z_{d,n}$ of each word are local variables; the topic vectors β_k are global variables.

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

1. Consider a K component Gaussian mixture model with parameters $\Theta = \{\pi_k, \boldsymbol{\mu}_k, \sigma_k^2\mathbf{I}\}_{k=1}^K$. Note that the covariance matrices of all the Gaussians are assumed to be spherical. Consider the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x}, \Theta)$, of the cluster assignment $\boldsymbol{z} \in \{1, \ldots, K\}$ for an observation $\boldsymbol{x} \in \mathbb{R}^D$.

   Write down the expression for $p(\boldsymbol{z}|\boldsymbol{x}, \Theta)$. What will be the entropy of this posterior distribution if the variances σ_k^2 of each Gaussian are set to a very small value (close to zero)? Justify your answer.
   Ans: The posterior will be $p(\boldsymbol{z} = k|\boldsymbol{x}, \Theta) \propto p(\boldsymbol{z} = k)p(\boldsymbol{x}|\boldsymbol{z} = k, \Theta)$. Therefore the posterior will be $p(\boldsymbol{z} = k|\boldsymbol{x}, \Theta) \propto \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \sigma_k^2\mathbf{I})$. Normalizing this expression will be

$$\eta_k = p(\boldsymbol{z} = k|\boldsymbol{x}, \Theta) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \sigma_k^2\mathbf{I})}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \sigma_j^2\mathbf{I})} \qquad k = 1, \ldots, K$$

   Now let's see what happens to the probability vector $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_K]$, which denotes the posterior probability distribution $p(\boldsymbol{z}|\boldsymbol{x}, \Theta)$ of assignments of $\boldsymbol{x}$ to each of the K clusters, as the variances σ_k^2 tend to zero? Well, $\boldsymbol{\eta}$ will start looking like a one-hot vector (one very large entry and all other entries being very small), and consequently the entropy of this distribution will be close to zero (we have very little uncertainty about the cluster assignment of $\boldsymbol{x}$, hence close to zero entropy for $p(\boldsymbol{z}|\boldsymbol{x}, \Theta)$.

   Why this happens? Well, note that the expression of η_k has $\exp\left(-\frac{||\boldsymbol{x}-\mu_k||^2}{2\sigma_k^2}\right)$ in the numerator and the demoninator is a summation consisting of such terms (for all $j = 1, \ldots, K$). As σ_k^2 goes to zero, the term for which $||\boldsymbol{x} - \mu_k||^2$ is the smallest (the "closest" cluster) will dominate the summation and the corresponding η_k will be close to 1 (and all other η_k's will be close to zero as a consequence).

   A side-note: This phenomenon is popularly known as "small variance asymptotics" and in this case the GMM becomes equivalent to a K-means type algorithm which performs hard clustering. This idea has also been used to design fast non-probabilistic counterparts for many probabilistic models, notably for nonparametric Bayesian clustering (if interested, check out the paper "Revisiting k-means: New Algorithms via Bayesian Nonparametrics"), and even used for problems other than clustering.

2. Consider the following model for generating N count-valued observations $y_1, y_2, \ldots, y_N$:

$$\begin{aligned} y_n &\sim \text{Poisson}(\lambda_n), \quad n = 1, \ldots, N \\ \lambda_n &\sim \text{Gamma}(1, b), \quad n = 1, \ldots, N \\ b &\sim \text{Gamma}(1, 1) \end{aligned}$$

Show that the model has local conjugacy, derive the conditional posteriors for all the unknowns, and sketch a Gibbs sampler for sampling from the posterior $p(\lambda_1, \lambda_2, \ldots, \lambda_N, b|y_1, y_2, \ldots, y_N)$.
Ans: It is easy to see that the model has local conjugacy by looking at the local conditional posterior of each unknown. $p(\lambda_n|y_1, \ldots, y_N, \lambda_{-n}, b) = p(\lambda_n|y_n) \propto p(\lambda_n)p(y_n|\lambda_n)$. This is a product of gamma prior Gamma$(\lambda_n|1, b)$ and Poisson likelihood Poisson$(y_n|\lambda_n)$ and the local CP will also be a Gamma, namely Gamma$(\lambda_n|1 + y_n, b + 1)$. Likewise, the local CP of b will be $p(b|y_1, \ldots, y_N, \lambda_1, \ldots, \lambda_N) = p(b|\lambda_1, \ldots, \lambda_N) = p(b)\prod_{n=1}^{N} p(\lambda_n|b)$, which is a product of gamma priors and N gamma likelihoods whose shape parameters is known. Gamma is conjugate to itself in such a case, which you can easily see if you notice that the resulting product (the posterior) will again be a gamma, namely Gamma$(b|1, 1 + \sum_{n=1}^{N} \lambda_n)$. The Gibbs sampler will alternate between sampling each $\lambda_n, n = 1, \ldots, N$ at a time from the respective local CP followed by sampling b from its local CP, and repeating it until convergence.

3. Consider a binary classification problem given training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ with $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \{0, 1\}$. Assume each label y_n to be generated as follows

$$\begin{aligned} z_n &= \boldsymbol{w}^\top \boldsymbol{x}_n + \epsilon_n, \qquad \epsilon_n \sim \mathcal{N}(0, 1) \\ y_n &= \mathbb{I}[z_n \geq 0] \end{aligned}$$

where $\mathbb{I}[.]$ is the indicator function which returns 1 if the condition is true, and returns 0 otherwise.

Derive the expression for the conditional $p(y_n|\boldsymbol{w}, \boldsymbol{x}_n)$ (which will require you to integrate out $\boldsymbol{z}_n$). Your answer must be in closed form.

For the likelihood model $p(y_n|\boldsymbol{w}, \boldsymbol{x}_n)$, is closed form MLE possible for $\boldsymbol{w}$? Irrespective of your answer to this, briefly state a possible way to do MLE for this model, with necessary expressions.
Ans: This model is actually the well-known Probit Regression model for binary classification. The likelihood $p(y_n|\boldsymbol{w}, \boldsymbol{x}_n)$ can be obtained by taking the joint distribution $p(y_n, z_n|\boldsymbol{w}, \boldsymbol{x}_n)$ and integrating out z_n. This will be given by $p(y_n|\boldsymbol{w}, \boldsymbol{x}_n) = \int_{-\infty}^{\infty} p(y_n|z_n)p(z_n|\boldsymbol{w}, \boldsymbol{x}_n)dz_n$.

Note that $p(y_n = 1|\boldsymbol{w}, \boldsymbol{x}_n) = \int_{-\infty}^{\infty} p(y_n = 1|z_n)p(z_n|\boldsymbol{w}, \boldsymbol{x}_n)dz_n = \int_{0}^{\infty} p(z_n|\boldsymbol{w}, \boldsymbol{x}_n)dz_n$. Also note that $\int_{0}^{\infty} p(z_n|\boldsymbol{w}, \boldsymbol{x}_n)dz_n = \int_{0}^{\infty} \mathcal{N}(z_n|\boldsymbol{w}^\top \boldsymbol{x}_n, 1)dz_n = \int_{-\boldsymbol{w}^\top \boldsymbol{x}_n}^{\infty} \mathcal{N}(\epsilon_n|0, 1)d\epsilon_n$ where $\epsilon_n = z_n - \boldsymbol{w}^\top \boldsymbol{x}_n$. Thus

$$p(y_n = 1|\boldsymbol{w}, \boldsymbol{x}_n) = \int_{-\boldsymbol{w}^\top \boldsymbol{x}_n}^{\infty} \mathcal{N}(\epsilon_n|0, 1)d\epsilon_n = 1 - \int_{0}^{-\boldsymbol{w}^\top \boldsymbol{x}_n} \mathcal{N}(\epsilon_n|0, 1)d\epsilon_n = 1 - \Phi(-\boldsymbol{w}^\top \boldsymbol{x}_n) = \Phi(\boldsymbol{w}^\top \boldsymbol{x}_n)$$

where $\Phi()$ denotes the CDF of $\mathcal{N}(0, 1)$. Therefore $p(y_n|\boldsymbol{w}, \boldsymbol{x}_n) = [\Phi(\boldsymbol{w}^\top \boldsymbol{x}_n)]^{y_n}[1 - \Phi(\boldsymbol{w}^\top \boldsymbol{x}_n)]^{1-y_n}$.

Closed form estimation is not possible for this likelihood (just like logistic regression case). However, we can use gradient descent to iteratively estimate $\boldsymbol{w}$. The gradient of the log likelihood is given by

$$\begin{aligned} \nabla \log p(y_n = 1|\boldsymbol{w}, \boldsymbol{x}_n) &= \nabla[y_n \log \Phi(\boldsymbol{w}^\top \boldsymbol{x}_n)] = \frac{y_n \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n|0, 1)}{\Phi(\boldsymbol{w}^\top \boldsymbol{x}_n)} \times \boldsymbol{x}_n \\ \nabla \log p(y_n = 0|\boldsymbol{w}, \boldsymbol{x}_n) &= \nabla[(1 - y_n) \log[1 - \Phi(\boldsymbol{w}^\top \boldsymbol{x}_n)]] = \frac{(1 - y_n)\mathcal{N}(-\boldsymbol{w}^\top \boldsymbol{x}_n|0, 1)}{\Phi(-\boldsymbol{w}^\top \boldsymbol{x}_n)} \times -\boldsymbol{x}_n \end{aligned}$$

where we have used the fact that $1 - \Phi(\boldsymbol{w}^\top \boldsymbol{x}_n) = \Phi(-\boldsymbol{w}^\top \boldsymbol{x}_n)$.

Finally, note that encoding the label y_n as $\hat{y}_n = \{-1, +1\}$ instead of $\{0, 1\}$, we can write the above gradients more compactly as $\nabla \log p(\hat{y}_n|\boldsymbol{w}, \boldsymbol{x}_n) = \frac{\hat{y}_n \boldsymbol{x}_n \mathcal{N}(\hat{y}_n \boldsymbol{w}^\top \boldsymbol{x}_n|0,1)}{\Phi(\hat{y}_n \boldsymbol{w}^\top \boldsymbol{x}_n)}$.

4. Consider the probabilistic PCA model $\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n$, with $\boldsymbol{x} \in \mathbb{R}^D$ and $\boldsymbol{z}_n \in \mathbb{R}^K$. Suppose that we are also given a binary label $y_n \in \{0, 1\}$ for each observation $\boldsymbol{x}_n$. Suggest at least two ways to incorporate this information in this model. You only need to briefly describe the corresponding generative models and do not need to derive the inference.
Ans: Approach 1: Extend the PPCA model such that the latent factors $\boldsymbol{z}_n$ generate the binary label via a logistic regression model, i.e., $p(y_n|\beta, \boldsymbol{z}_n)$ where $\beta \in \mathbb{R}^K$ would be the weight vector of this regression model. This way the latent factors $\boldsymbol{z}_n$ generate both $\boldsymbol{x}_n$ and y_n and therefore both $\boldsymbol{x}_n$ and

y_n influence the posterior distribution of $\boldsymbol{z}_n$. Note that the two models are not learned separately (one after the other) but jointly. This is a very common way to construct a supervised PPCA modes.

Approach 2: We have have to PPCA models (defined by matrices $\mathbf{W}_0$ and $\mathbf{W}_1$, and depending on the value of y_n can invoke the appropriate one to generate $\boldsymbol{x}_n$.

Another approach would be assume that $\boldsymbol{z}_n$ is drawn from a mixture of two Gaussians (with the choice depending on y_n) rather than a single Gaussian $\mathcal{N}(0, \mathbf{I}_K)$ as done in standard PPCA.

5. Consider a Poisson GLM $p(y|\boldsymbol{w}, \boldsymbol{x}) = \text{Poisson}(y|\exp(\boldsymbol{w}^\top \boldsymbol{x}))$ with a prior on the weight vector $\boldsymbol{w} \in \mathbb{R}^D$ being $p(\boldsymbol{w}) = \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I})$. Suppose you are given training data in form of N examples $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$ and suppose λ is known. Your goal is to infer the posterior distribution of $\boldsymbol{w}$.

   Propose a sampling based inference algorithm for this model that uses the posterior's gradient information. Write down all the steps and the expressions involved in the posterior updates.
   Ans: Note that the model is not conjugate. However, we can use Langevin dynamics (or stochastic gradient Langevin dynamics which works with small minibatches of data) to perform inference in this model. These methods only require the gradient of the log-posterior distribution. The gradient of the log-posterior distribution for this model would be (skipping the calculations which are straightforward)

$$\boldsymbol{g} = \nabla \log p(\boldsymbol{w}|\mathbf{X}, \boldsymbol{y}) = \nabla[\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}) + \log p(\boldsymbol{w}|\lambda)] = \sum_{n=1}^{N}[y_n - \exp(\boldsymbol{w}^\top \boldsymbol{x}_n)]\boldsymbol{x}_n - \lambda \boldsymbol{w}$$

   (As an aside, note the form of the gradient, ignoring the part due to prior, is $(y_n - \exp(\boldsymbol{w}^\top \boldsymbol{x}_n)$, which is "true response minus the expected value of predict response, times the feature vector")

   Using these gradient expressions, we can easily derive a Langevin dynamics (or SGLD) sampling algorithm. When using (batch) LD, the algorithm works as follows

   - Initialize $\boldsymbol{w}^{(0)}$ randomly
   - For $t = 1, \ldots, T$
     - Compute $\boldsymbol{w}^*$ as $\boldsymbol{w}^* = \boldsymbol{w}^{(t-1)} + \frac{\eta}{2}\left(\sum_{n=1}^{N}[y_n - \exp({\boldsymbol{w}^{(t-1)}}^\top \boldsymbol{x}_n)]\boldsymbol{x}_n - \lambda \boldsymbol{w}^{(t-1)}\right)$
     - $\boldsymbol{w}^{(t)} \sim \mathcal{N}(\boldsymbol{w}^*, \eta\mathbf{I})$ and then accept/reject using MH (record $\boldsymbol{w}^{(t)}$ only if accepted)

5 Long Answer Problem (1 question, 15 marks)

Suppose you are given a collection of D labeled documents $\{\boldsymbol{w}_d, y_d\}_{d=1}^D$ where $\boldsymbol{w}_d = \{w_{d,n}\}_{n=1}^{N_d}$ denotes the set of N_d words in the document d and $y_d \in \{1, \ldots, K\}$ denotes the label of this document. Assume the vocabulary size (number of unique words across all the documents) to be V.

Consider the following generative story for each labeled document $\{\boldsymbol{w}_d, y_d\}$

- Choose a label $y_d \in \{1, \ldots, K\}$ from a categorical distribution with parameters $\pi = [\pi_1, \ldots, \pi_K]$

  $y_d \sim \text{multinoulli}(\pi)$

- Generate each word in this document i.i.d. from a categorical distribution with parameters $\phi^{(y_d)}$, where $\phi^{(y_d)}$ is a probability vector of size V.

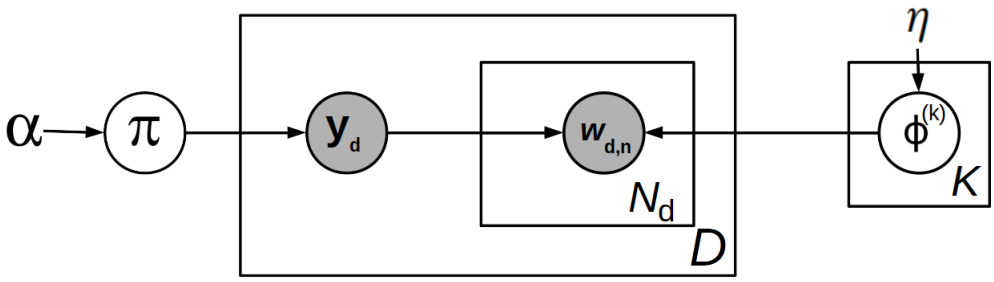
  $w_{d,n} \sim \text{multinoulli}(\phi^{(y_d)}) \qquad n = 1, \ldots, N_d$

Assume Dirichlet priors on all the multinoulli parameters

$$\begin{aligned} \pi &\sim \text{Dirichlet}(\alpha, \ldots, \alpha) \\ \phi^{(k)} &\sim \text{Dirichlet}(\beta, \ldots, \beta), \qquad k = 1, \ldots, K \end{aligned}$$

Assume the hyperparameters α and β to be known. Now answer the following questions:

- Draw the plate notation diagram for this model, clearly showing which parts are unknowns and which parts are observed (usual convention: white nodes = unobserved, shaded nodes = observed/known).
  Ans: The plate notation diagram is shown below



- How is this model different from a supervised variant of Latent Dirichlet Allocation (LDA) model that has access to class labels (which you considered in Homework 4)? In particular, is the above generative model less expressive or more expressive than the supervised LDA model?
  Ans: In the model being considered, given the class label y_d, all the words in the document are generated from a single distribution specific to that class. In contrast, LDA (and even supervised LDA which does have access to class label y_d) assumes that the document is a mixture of K **latent** topics and each word in a document can belong to one of those K possible topics). In that sense LDA or supervised LDA are more expressive than the model under consideration here. This model is actually a simple generative classification model, basically a naive Bayes classification model with multinoulli likelihood for each feature of an observation (though a fully Bayesian setting is considered here).

- Derive the posterior distribution for π and $\Phi = \{\phi^{(k)}\}_{k=1}^K$. Also write down the MAP solutions for π and $\{\phi^{(k)}\}_{k=1}^K$ (you don't need to re-derive the MAP solution from scratch; instead, you should be able to get these easily from the expression of their respective posterior distributions).
  Ans: The posterior distribution for π is straightforward. This will simply be $p(\pi|\{\boldsymbol{w}_d, y_d\}_{d=1}^D) = \text{Dirichlet}(\alpha + N_1, \ldots, \alpha + N_K)$ where N_k denotes the number of documents with label $y_d = k$.

  Likewise, estimating the posterior of each $\phi^{(k)}$ can be done in a straightforward manner by recognizing that we only need to consider documents belonging to class k. The posterior $p(\phi^{(k)}|\{\boldsymbol{w}_d, y_d\}_{d=1}^D)$ will simply be equal to $\text{Dirichlet}(\beta + N_{k1}, \ldots, \beta + N_{kV})$ where N_{kv} denotes the number of times word v appeared across all documents belonging to class k.

  The MAP estimates are straightforward as well. These are simply given by the mode of the above Dirichlet distributions. Note that, for $\text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$, the mode will be

$$\left[\frac{\alpha_1 - 1}{\sum_{k=1}^K (\alpha_k - 1)}, \ldots, \frac{\alpha_K - 1}{\sum_{k=1}^K (\alpha_k - 1)}\right]$$

  We can use this expression to compute the mode of both of the above Dirichlet distributions (which directly gives us the MAP estimates of π and $\phi^{(k)}$).

- How would you use this model for classification, i.e., given a new document $\boldsymbol{w}_* = \{w_{*,n}\}_{n=1}^{N_*}$, where N_* denotes the number of words in it, how would you predict the class label y_* of this document? In particular, give the expression for predictive distribution of y_* given the MAP estimates, i.e., $p(y_*|\boldsymbol{w}_*, \pi_{MAP}, \Phi_{MAP})$, as well as the posterior predictive distribution $p(y_*|\boldsymbol{w}_*, \{\boldsymbol{w}_d, y_d\}_{d=1}^D)$. For the posterior predictive, you can just write down the basic expression with all the distributions involved and leave it at that point if you don't know the required integrals to further simplify it.
  Ans: The predictive distribution is $p(y_* = k|\boldsymbol{w}_*, \pi_{MAP}, \Phi_{MAP}) \propto p(y_* = k|\pi_{MAP})p(\boldsymbol{w}_*|\Phi_{MAP}) = \pi_{k,MAP} \times \prod_{n=1}^{N_*} p(w_{*,n}|\phi_{MAP}^{(k)})$ where $\pi_{k,MAP}$ is the k^{th} component of π_{MAP} and $p(w_{*,n}|\phi_{MAP}^{(k)})$ is the $w_{*,n}^{th}$ component of V dimensional vector $\phi_{MAP}^{(k)}$.

The posterior predictive will require integrating out π and Φ using their posteriors, i.e.,

$$\begin{aligned} p(y_* = k|\boldsymbol{w}_*) &\propto \int p(y_* = k|\boldsymbol{w}_*, \pi, \Phi) p(\pi_k|\{\boldsymbol{w}_d, y_d\}_{d=1}^D) p(\phi^{(k)}|\{\boldsymbol{w}_d, y_d\}_{d=1}^D) d\pi_k d\phi^{(k)} \\ &= \int \pi_k \prod_{n=1}^{N_*} p(w_{*,n}|\phi^{(k)}) p(\pi_k|\{\boldsymbol{w}_d, y_d\}_{d=1}^D) p(\phi^{(k)}|\{\boldsymbol{w}_d, y_d\}_{d=1}^D) d\pi_k d\phi^{(k)} \end{aligned}$$

Despite the somewhat unwieldly-looking expression, it is actually easy to simplify. I'm lazy to do it here but the color-coding I have shown above should tell you how it should be done. Basically, the terms in red and green denote two simple expectations that are easy to compute (I leave it to you to think what these would be!). **Note:** For this part, I didn't expect you to show the expressions in great detail or give a closed form expression (which actually exists in this case!). Even if you have written the basic equations/mathematical definitions for predictive and posterior predictive distributions correctly, you will get full credit for this part.