Name:

Roll No.: **Dept.:**

IIT Kanpur
CS771A (IML)
Quiz-3
Date: October 18, 2023

Instructions: *Total:* **30 marks**

1. Duration is 60 minutes. Please write your name, roll number, department on **all pages**.
2. Write your answers clearly in the provided box. Keep your answer precise and concise.

Section 1 (Short/medium-length answer questions: 30 marks)**.** .

1. Given N coin-toss outcomes $\{y_1, y_2, \ldots, y_N\}$, assuming $p(y_n|\theta) = \text{Bernoulli}(y_n|\theta)$, with $y_n = 1$ denoting a heads outcome, the MLE solution is $\theta_{MLE} = \frac{\sum_{n=1}^N y_n}{N}$. Further assuming a $\text{Beta}(\theta|\alpha,\beta)$ prior on θ, the MAP solution is $\theta_{MLE} = \frac{\sum_{n=1}^N y_n + \alpha - 1}{N+\alpha+\beta-2}$. Comparing these results, briefly describe what the prior's hyperparameters α and β signify/denote in this problem? **(2 marks)**
   They denote the number of pseudo-observations ($\alpha - 1$ and $\beta - 1$ being the number of heads and tails, respectively that we have prior to getting the actual observations).

2. Assume a prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_0, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{w}_0)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{w}_0)\}$ on the weight vector of a linear model (assume prior's hyperparameters $\boldsymbol{w}_0, \boldsymbol{\Sigma}$ to be known). What does this specific prior say about $\boldsymbol{w}$? Also write down the expression of the corresponding regularizer on $\boldsymbol{w}$. **(2 marks)**
   It says that, *a priori*, we believe the weight vector $\boldsymbol{w}$ to be close to $\boldsymbol{w}_0$ with our belief having some uncertainty which is governed by the covariance matrix $\boldsymbol{\Sigma}$. The corresponding regularizer is the negative log of this prior, and focusing only on terms that contain $\boldsymbol{w}$, it is $\frac{1}{2}(\boldsymbol{w}-\boldsymbol{w}_0)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{w}-\boldsymbol{w}_0)$.

3. Is logistic regression a generative model or a discriminative model? Briefly explain. Also answer the same question for probabilistic linear regression with Gaussian likelihood for the responses. **(2 marks)**
   Both are discriminative models since they only model the distribution of y_n conditioned on $\boldsymbol{x}_n$ and some parameter $\boldsymbol{w}$, i.e., $p(y_n|\boldsymbol{w}, \boldsymbol{x}_n)$, and we do not assume any distribution for $\boldsymbol{x}_n$.

4. In generative classification model with Gaussian class-conditionals, what are the parameters that we need to estimate? Also write down the expression, in terms of these parameters, for probability of a test input $\boldsymbol{x}_*$ belonging to class $k \in \{1, 2, \ldots, K\}$ (your answer up to a proportionality constant is fine). **(2 marks)**
   We need to estimate $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ where $\pi_k = p(y_n = k)$ are the class priors/class marginals for class k, and μ_k, Σ_k denote the mean and covariance matrix of the k-th class-conditional (which is a Gaussian). The required expression is $p(y_* = k|\boldsymbol{x}_*, \Theta) \propto p(y_* = k)p(\boldsymbol{x}_n|y_* = k) = \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$, up to a proportionality constant.

5. Given the learned means $\mu_1, \mu_2, \ldots, \mu_K$ from a K-means clustering model, how would you obtain a soft clustering for input $\boldsymbol{x}_n$. Write your answer in terms of the precise mathematical expression. **(2 marks)**
   The probability of $\boldsymbol{x}_n$ belonging to cluster k will be $p(\boldsymbol{z}_n = k|\boldsymbol{x}_n) = \frac{\exp(-||\boldsymbol{x}_n - \mu_k||^2)}{\sum_{\ell=1}^K \exp(-||\boldsymbol{x}_n - \mu_\ell||^2}$.

6. If you learn a Gaussian mixture model (assuming K Gaussians) using the ALT-OPT algorithm, will the expression for the hard guess of the cluster id $\boldsymbol{z}_n$ for an input $\boldsymbol{x}_n$ be the same as what you would have in K-means clustering algorithm? Briefly explain your answer. **(2 marks)**
   No, it won't be the same as K-means because even the hard guess in GMM with ALT-OPT takes into account the distribution of each cluster (Gaussian in this case), not just the distance from the mean of the clusters. More specifically, this hard guess $\hat{\boldsymbol{z}}_n = \text{argmax}_{k\in\{1,2,\ldots,K\}} p(\boldsymbol{z}_n = k|\boldsymbol{x}_n, \Theta) = \text{argmax}_{k\in\{1,2,\ldots,K\}} p(\boldsymbol{z}_n = k)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \Theta) = \text{argmax}_{k\in\{1,2,\ldots,K\}} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$.

7. Given a Gaussian posterior distribution $\mathcal{N}(\theta|\mu, \sigma^2)$ for some parameter $\theta \in \mathbb{R}$, what is the MAP solution? Given this Gaussian posterior, can you obtain the MLE solution? Briefly explain your answer. **(2 marks)**
   MAP solution is simply the mode of the posterior. For a Gaussian, mode is equal to the mean, and therefore the MAP solution is μ. We can't obtain the MLE solution from the posterior because the MLE is the maxima of the likelihood function we are given the posterior distribution which has already combined the likelihood and the prior and we can't "extract" the likelihood out of it (unless the prior is uniform). :)

Name:

Roll No.: **Dept.:**

IIT Kanpur
CS771A (IML)
Quiz-3
Date: October 18, 2023

8. Briefly explain the basic idea of multi-dimensional scaling (MDS). Why is MDS not suitable for getting out-of-sample embedding? **(2 marks)**
   Given pairwise distances d_{ij} is large for two inputs indxed by i and j then MDS prefers their embeddings $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ to be far away (and near if d_{ij} is small). MDS is not suitable for out-of-sample embeddings because it directly optimizes for the embeddings of the training points and does not learn an explicit encoder function to give embeddings of new points that are not part of the training data.

9. Write down the expression for an appropriate distortion/reconstruction error function for a matrix factorization problem in which we wish to factorize an $N \times M$ matrix $\mathbf{X}$ as a product of an $N \times K$ matrix $\mathbf{Z}$ and a $K \times M$ matrix $\mathbf{W}$. Briefly explain how would you solve for $\mathbf{Z}$ and $\mathbf{W}$? **(2 marks)**
   The reconstruction error/loss can be the Frobenious norm $||\mathbf{X} - \mathbf{ZW}||^2$. We can learn $\mathbf{Z}$ and $\mathbf{W}$ by minimizing this error and use ALT-OPT to minimize this loss function (optimize for $\mathbf{Z}$ treating $\mathbf{W}$ as given, and vice-versa).

10. In PCA, what top-most eigenvector (i.e., which corresponds to the largest eigenvalue) means? **(2 marks)**
    The top-most eigenvector denotes the direction along which the data has the largest variance, i.e., the most important direction if we use the variance as the criterion to judge the importance..

11. Briefly explain why PCA might not be a suitable dimensionality reduction method if our eventual goal is to learn a classification model using the lower-dimensional inputs obtained by PCA? **(2 marks)**
    In some problems, e.g., classification, the directions along which the inputs have the largest variance, may not be the most important ones (e.g., projecting the inputs along the largest variance directions may destroy the class separation).

12. Briefly explain the basic idea of K-means++. **(2 marks)**
    K-means++ is a smart initialization scheme for K-means clustering. The basic idea is to choose the initial cluster centers as K of the inputs that are guaranteed to be reasonably far apart from each other. K-means++ uses a sequential scheme to do so such that the next mean to be initialized is one of the inputs that is the farthest away from the already initialized means.

13. Briefly explain what is semi-supervised learning and how can we use a latent variable model for semi-supervised learning. **(2 marks)**
    Semi-supervised learning uses a mix of labeled (usually available in small amounts) as well as unlabeled training data (usually available in large amounts) to learn a predictive model for output given the input. We can treat the label of each unlabeled input $\boldsymbol{x}_n$ as a latent variable $\boldsymbol{z}_n$, estimate it using ALT-OPT/EM so the unlabeled data can also be treated like labeled (or "pseudo"-labeled) data along with the actual labeled data, and both can now be combined to learn the predictive model.

14. For a latent variable model with likelihood $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)$, prior $p(\boldsymbol{z}_n|\phi)$ on the latent variables $\boldsymbol{z}_n$, write down the expression the incomplete data log-likelihood (ILL) $p(\boldsymbol{x}_n|\theta, \phi)$, assuming $\boldsymbol{z}_n$ is discrete. **(2 marks)**

    Denoting $\Theta = (\theta, \phi)$, and assuming $\boldsymbol{z}_n$ is discrete with K possible values, the ILL is

    $$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} p(\boldsymbol{x}_n, \boldsymbol{z}_n = k|\Theta) = \sum_{k=1}^{K} p(\boldsymbol{z}_n = k|\phi) p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \theta)$$

    .

15. For a dimensionality reduction method, briefly explain what the encoder and decoder components of the model are used for? **(2 marks)**
    Given an input $\boldsymbol{x}_n$, the encoder (say a function f) computes a new (usually lower-dimemsional) representation/embedding $\boldsymbol{z}_n$, so $\boldsymbol{z}_n = f(\boldsymbol{x}_n)$. A decoder (say a function g) takes the embedding $\boldsymbol{z}_n$ and reconstructs (usually an approximation of) the original input as $\hat{\boldsymbol{x}}_n = g(\boldsymbol{z}_n)$.