Introduction to ML (CS771), 2024-2025-Sem-I Quiz 3. October 24, 2024		Total Marks	25
		Duration	45 minutes
Name		Roll No.	

Instructions:

1.	Clearly write your name (in block letters) and roll number in the provided boxes above.
2.	Write your final answers concisely in the provided space. You may use blue/black pen.
3.	We won't be able to provide clarifications during the quiz. If any aspect of some question appears ambiguous/unclear to you, please state your assumption(s) and answer accordingly.

Question 1: Write **T** or **F** for True/False in the box next to each question given below, with a brief (1-2 sentences at most) explanation in the provided space in the box below the question. Marks will be awarded only when the answer (T/F) and explanation both are correct. (**3 x 2 = 6 marks**)

1.1	To predict the label using a generative classification model, comparing the probabilities $p(y = k\|\boldsymbol{x})$ for different values of k is equivalent to comparing the class-conditional probability densities $p(\boldsymbol{x}\|y = k)$ for different values of k	**F**
$p(y = k\|\boldsymbol{x}) \propto p(y = k)p(\boldsymbol{x}\|y = k)$ so it also incorporate the class prior (class marginal) distribution $p(y = k)$		

1.2	A Gaussian prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}\|\boldsymbol{w}_0, \lambda^{-1}\boldsymbol{I})$ on the weight vector $\boldsymbol{w} \in \mathbb{R}^D$ will cause a regularization effect and encourage the entries in $\boldsymbol{w}$ to take small values.	**F**
This prior corresponds to a regularizer of the form $\lambda\|\boldsymbol{w} - \boldsymbol{w}_0\|^2$ which will encourage each entry of the vector $\boldsymbol{w}$ to be close the the corresponding entry in the vector $\boldsymbol{w}_0$. Only when $\boldsymbol{w}_0$ is the zero vector, the statement above would be true but in general it would be false.		

1.3	Even though the MAP estimate is the mode of the posterior distribution, to compute the MAP estimate, it is not necessary to compute the posterior distribution.	**T**
Recall that the posterior is $p(\theta\|y) = \frac{p(\theta)p(y\|\theta)}{p(y)}$. Because of the denominator (marginal likelihood) is independent of θ, maximization of the posterior only requires maximization of the numerator $p(\theta)p(y\|\theta)$ (or $\log p(\theta) + \log p(y\|\theta)$) and we don't need to compute the full posterior for the maximization.		

Question 2: Answer the following questions concisely in the space provided below the question.

2.1	Consider the RBF kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$ where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are D dim inputs. Consider two cases: (1) when bandwidth hyperparameter γ is set as very-very large, and (2) when γ is set as very-very small. For each of these two cases, answer (with brief justification) whether the resulting kernel function would be practically useful. **(4 marks)**
(1) When γ is very-very large, $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ will be nonzero (will equal 1) only when $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are nearly identical. For all other pairs of inputs, the kernel will give 0 similarity. (2) When γ is very-very small, $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ will be close to 1 for all pairs of inputs, thus treating all pairs of inputs as equally similar to each other. Clearly, neither of these two extreme cases are desirable.

2.2	Briefly explain why using kernels with the landmarks approach or the random features approach is faster at test time than using kernels in the standard manner? **(3 marks)**

When using landmarks or random features approach, we use the kernel to construct an L dimensional feature representation $\psi(\boldsymbol{x}_n)$, and train a linear model on these representations to get a weight vector $\boldsymbol{w}$ that is L dimensional. Thus, for a test input $\boldsymbol{x}_*$ the prediction cost for computing $w^\top \psi(\boldsymbol{x}_*)$ is also $O(L)$. In contrast, when using the kernel in the standard manner, this cost of $O(N)$ which can be very high if the number of training inputs is very large

2.3	Given a dataset $\boldsymbol{X}$ as the $N \times D$ input matrix with N inputs and D features, write down the K-means hard-clustering problem for this dataset in form of an equivalent matrix factorization problem, clearly specifying the meanings of the variables involved in the matrix factorization, their dimensions, and constraints on them, if any. **(4 marks)**

$\{\widehat{\boldsymbol{Z}}, \widehat{\boldsymbol{\mu}}\} = \text{argmin}_{\boldsymbol{Z},\boldsymbol{\mu}} \|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{\mu}\|^2$. Here $\boldsymbol{Z}$ is the $N \times K$ matrix with row n ($\boldsymbol{z}_n$) being a one-hot vector denoting which cluster the input $\boldsymbol{x}_n$ belongs to, and $\boldsymbol{\mu}$ denotes the $K \times D$ matrix with row k ($\boldsymbol{\mu}_k$) denoting the mean of the k^{th} cluster.

Constraints on $\boldsymbol{z}_n$: Must be a one-hot vector

Constraints on $\boldsymbol{\mu}_k$: None

2.4	Why is it difficult to compute the predictive distribution of a logistic regression model which, by definition, is given by $p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y_* = 1|\boldsymbol{w}, \boldsymbol{x}_*)p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})d\boldsymbol{w}$. Suggest a method to approximate it and clearly show the necessary equations. **(3 marks)**

It is difficult because the integral here is not tractable (involves integrating $p(y_* = 1|\boldsymbol{w}, \boldsymbol{x}_*)$ which is a sigmoid function over the posterior $p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y})$ and even if the latter is Gaussian (like in Laplace approximation), the integral still is intractable. To approximate the integral, one way is to use Monte-Carlo approximation where we draw S i.i.d. samples $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \ldots, \boldsymbol{w}^{(S)}$ from the posterior and approximate the predictive distribution as

$$p(y_* = 1|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) \approx \frac{1}{S}\sum_{s=1}^{S} p(y_* = 1|\boldsymbol{w}^{(s)}, \boldsymbol{x}_*)$$

2.5	Show that, for generative classification with uniform class marginal and Gaussian class conditionals $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, the posterior probability of input $\boldsymbol{x}$ belonging to class k, i.e., $p(y = k|\boldsymbol{x}) \propto \exp(\boldsymbol{w}_k^\top \boldsymbol{x} + b_k)$, and write down the expressions for $\boldsymbol{w}_k$ and b_k **(5 marks)**

Since we have a uniform class marginal, the posterior probability will be

$$p(y = k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y = k)}{p(\boldsymbol{x})} \propto p(\boldsymbol{x}|y = k)$$

Since the class conditional $p(\boldsymbol{x}|y = k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, we have

$p(\boldsymbol{x}|y = k) \propto \exp\left(-\frac{1}{2}((\boldsymbol{x} - \boldsymbol{\mu}_k)\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\right) \propto \exp\left(\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}\boldsymbol{\mu}_k\right)$, where in the last expression (after the proportionality sign), we are ignoring any terms that are not specific to class k. Thus $p(y = k|\boldsymbol{x}) \propto p(\boldsymbol{x}|y = k) \propto \exp\left(\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}\boldsymbol{\mu}_k\right)$ which is clearly in the form of $\exp(\boldsymbol{w}_k^\top \boldsymbol{x} + b_k)$ where $\boldsymbol{w}_k = (\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1})^\top = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$ and $b_k = -\frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}\boldsymbol{\mu}_k$

Side note (not required for the answer): Note that the above implies that this generative classification model has a similar form as softmax classification model, although the weight vector is learned using a generative manner, and not using GD as we do in case of softmax classification