Name: _____

Roll No.: _____ **Dept.**: _____

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022

Instructions: *Total:* **100 marks**

1. Total duration: **3 hours**. Please write your name, roll number, department on **all pages**.
2. This booklet has 10 pages (9 pages + 1 page for rough work). No part of your answers should be on pages designated for rough work. Additional rough sheets may be provided if needed.
3. Write/mark your answers clearly in the provided space. Please keep your answers precise and concise.
4. Avoid showing very detailed derivations (you may use the rough sheet for that). In some cases, you may directly use the standard results/expressions provided on page 6 of this booklet.

Section 1 (True or False: 15 X 1 = 15 marks)**.** For each of the following simply write **T** or **F** in the box.

1. [**F**] The frequentist approach cannot provide an estimate of the parameter uncertainty.
2. [**T**] SGLD sampling based inference usually converges faster than Metropolis-Hastings.
3. [**T**] In Metropolis-Hastings sampling for posterior inference, the cost of calculating the acceptance probability depends on the dataset size.
4. [**F**] The predictive variance of a regression model when using the MAP estimate of the weights depends on the inputs.
5. [**F**] When using VI to approximate the posterior, the posterior predictive distribution does not require an approximation.
6. [**T**] A non-probabilistic autoencoder cannot be used to generate new data.
7. [**T**] A standard generative adversarial network (GAN) cannot be used to compress/encode data.
8. [**T**] If a node c is the common parent of two nodes a and b and there are no other edges between nodes, then conditioned on c, nodes a and b are independent.
9. [**F**] A generative model for classification, in general, has fewer parameters to estimate than a discriminative model.
10. [**F**] A generalized linear model (GLM) is guaranteed to have a closed form solution for its parameters when doing MLE/MAP estimation.
11. [**T**] Storing the VI approximation of a posterior distribution requires less space as compared to its MCMC approximation.
12. [**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 always be ignored.
13. [**F**] Active Learning is based on querying the labels of those unlabeled examples whose predicted label distribution has the smallest entropy.
14. [**T**] The posterior predictive distribution of Bayesian linear regression with Gaussian likelihood, Gaussian prior on weights, and rest of the hyperparameters being unknown, is not tractable.
15. [**T**] A large value of the concentration parameter α in a Dirichlet Process based nonparametric Bayesian model for clustering will lead to large inferred number of clusters.

Section 2 (15 short answer questions: 15 x 3 = 45 marks)**.** .

1. How is Bayesian ML different from"conventional" ML? Specifically, state the difference in terms of: (a) How parameter estimation is done? (b) How predictions are made?

   Answer: In Bayesian ML, parameter estimation is done by inferring the posterior distribution over the unknowns instead of by doing point estimation. Predictions are made by marginalizing over the posterior distribution of the unknowns (i.e, computing the PPD), instead of using their point estimate (i.e., computing the plug-in predictive distribution).

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022

---

2. Given two models, $\mathcal{M}_1$ and $\mathcal{M}_2$ with their average accuracies and average confidence scores being a_1, c_1 and a_2, c_2, respectively, in a rough sense how would be find out which model has a better calibration?

   Answer: The model with less gap between the accuracy and confidence scores will have better calibration.

3. Suppose you have learned the latent Dirichlet allocation (LDA) model on some text data. Given a topic vector ϕ_k, how would you find out what topic (in terms of its semantic meaning) ϕ_k represents?

   Answer: Each entry ϕ_{kv} is the vector ϕ_k denotes the probability of observing word v in topic k. We can find the top few (say 5-10) largest entries in ϕ_k and see which words they correspond to. These words (which will roughly be about the same concept/topic) can tell us which topic ϕ_k represents.

4. Briefly explain how a shrinkage based approach to nonparametric Bayesian modeling helps learn the right model size. You may use an example of a model discussed in the class.

   Answer: The shrinkage based approach overspecifies the number of parameters. After inference, only a finite/small number of parameters will have significant value, and this number will indicate the right model size. For example, in the nonparametric Bayesian clustering, we can assume an infinite-sized vector $\pi_1, \pi_2, \pi_2, \ldots$. After inference, we can see how many of the π_k's have significant values, and that will tell us the right number of clusters.

5. Why would a method like Rejection Sampling not be an ideal choice for doing posterior inference for a model like Bayesian linear regression or logistic regression or various other Bayesian ML models?

   It is because these models usually have parameters that are high-dimensional (e.g., the weight vector) and Rejection Sampling is not ideal in such situations.

6. Laplace approximation can be expensive for Bayesian neural networks due to the enormous number of parameters. How is the Laplace approximation used in a hybrid Bayesian neural network?

   We can perform point estimation for the weights of the hidden layers and use compute a Laplace approximation based posterior for the output layer weights.

7. In what way is the deep ensemble approach better than inference methods such as sampling or VI in terms of the quality of inferred posterior?

   Deep ensembles can capture the different modes of the posterior (because different runs of the optimizer would potentially capture different optima) whereas methods like sampling and VI are typically limited to exploring regions/uncertainty around one of the modes (although they can explore multiple modes, it is usually more difficult and requires more expressive q distribution in VI, or carefully designed MCMC samplers).

8. (Bayesian) Active Learning and Bayesian Optimization use an acquisition function to decide which observation(s) to acquire next. Briefly describe the basic difference between the goals of the respective acquisition functions used in both these methods?

   Active learning acquisition functions are designed to select inputs at which the current estimate of the function is most uncertain about the label. BO acquisition functions are design to select inputs at which the current estimate of the function is the most uncertain about the function's value as well as at which the function has small (or large if doing maximization) value.

9. Can you combine a standard (not Bayesian) deep neural network and Bayesian linear regression model to do Bayesian *nonlinear* regression? If yes, how? If no, why not?

   Yes. We can use a deep neural net to extract nonlinear features, and then train a Bayesian linear regression using these extracted features.

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022

10. Briefly explain why VI which is based on minimizing $\text{KL}[q(Z)||p(Z|X)] = \int q(Z) \log \frac{q(Z)}{p(Z|X)} dZ$ tends to underestimate the variance of the true posterior $p(Z|X)$.

    It is because, to keep the KL small, the optimizer would avoid taking $q(Z)$ to the regions where $p(Z|X)$ is very small (for example, the low-probability region where $p(Z|X)$ is transitioning from one mode to the other mode). So $q(Z)$ would mostly be limited in regions where $p(Z|X)$ is large, thereby the variance of $p(Z|X)$ being underestimated

11. State 3 advantages (should be distinct from each-other) of Gaussian Process based models over standard, kernelized supervised learning models, such as SVM, or methods such as nearest neighbors.

    (1) GPs provide an estimate of the model/function uncertainty, (2) GPs provide an estimate of the predictive uncertainty. (3) When using GPs, we can estimate the model hyperparameters (e.g., kernel's hyperparameters) from data.

12. 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?

    Note that the prior, as a function of θ, has the same functional form as the likelihood (both are quadratic functions of θ). Thus the prior is conjugate to the likelihood.

13. Suppose you have a coin with bias (probability of a head) π having a prior $\text{Beta}(\pi|a, b)$. You toss the coin until you see a total of k heads (assume i.i.d. tosses). Note that the number of heads is fixed here but the total number (call it n) of tosses is stochastic. Give the expression for the likelihood, and also the posterior of π (no need to derive it; just the final answer is fine if you can find it by inspection/intuition).

    We basically are looking at a scenario where there were $k-1$ heads on the first $n-1$ tosses and the n-th toss was a heads. The first condition can be modeled by a binomial and the second condition by a Bernoulli. Thus the likelihood for the number of tosses X being n will be $p(X = n|k, \pi) = \binom{n-1}{k-1}\pi^{(k-1)}(1-\pi)^{(n-k)} \times \pi$. Since this has the same form as a binomial, the posterior will be $\text{Beta}(a + k, b + n - k)$.

14. What is the benefit of using amortized inference for inferring the local latent variables?

    Amortized inference learns a single neural network $\phi_n = \text{NN}(x_n; W)$ which can be used to predict the parameters ϕ_n of the posterior $p(z_n|x_n)$ for each local latent variable z_n. In contrast, if we estimate each ϕ_n separately then it makes the update of global variables slow since we would need to estimate all the ϕ_n's first. Moreover, amortized inference also makes it fast at test time when we want to infer the posterior of the local latent variable z_* for a new data point x_*.

15. What is the meaning of "collapsing" in context of Bayesian inference? What is the benefit of collapsing?

    Collapsing refers to marginalizing/integrating out some of the unknowns of the model. As a result, we need to infer fewer variables which makes inference faster.

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022

Section 3 (4 not-so-short answer questions: 10+8+12+10 = 40 marks). .

1. Suppose you are given N data-points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ where each data-point $\boldsymbol{x}_n \in \mathbb{R}^D$ has some missing features. Denote it as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{(o)}, \boldsymbol{x}_n^{(m)}]$, where $\boldsymbol{x}_n^{(o)}$ denotes the observed features and $\boldsymbol{x}_n^{(m)}$ denotes the missing features. The set of features missing could be different in different data-points. Assume each data-point $\boldsymbol{x}_n$ is generated from a Gaussian $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Give an EM algorithm to compute the MLE of the parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of this Gaussian (in the box below, you should only mention the key steps/equations in the EM algorithm and skip detailed derivations) .
   Hint: Treat the missing features $\boldsymbol{x}_n^{(m)}$ of each data-point as latent variables.
   Note: You may use the standard result that the MLE of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for a Gaussian when there are no missing features is given by $\boldsymbol{\mu}_{MLE} = \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n$, and $\boldsymbol{\Sigma}_{MLE} = \frac{1}{N}\sum_{n=1}^N (\boldsymbol{x}_n - \boldsymbol{\mu}_{MLE})(\boldsymbol{x}_n - \boldsymbol{\mu}_{MLE})^\top$

   We are assuming that we have arranged the features as $\boldsymbol{x}_n = [\boldsymbol{x}_n^{(o)}, \boldsymbol{x}_n^{(m)}]$ such that the observed features are written first followed by all the missing features. Let's denote $\boldsymbol{\mu}$ as $\boldsymbol{\mu} = [\boldsymbol{\mu}_o, \boldsymbol{\mu}_m]$ and $\boldsymbol{\Sigma}$ as $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{oo} & \boldsymbol{\Sigma}_{om} \\ \boldsymbol{\Sigma}_{mo} & \boldsymbol{\Sigma}_{mm} \end{bmatrix}$ (note that this partitioning structure will be different for each observation since the set of missing features will be different for each $\boldsymbol{x}_n$)

   Using Gaussian's properties, we can get the conditional distribution of $\boldsymbol{x}_n^{(m)}$ given $\boldsymbol{x}_n^{(o)}$, i.e., $p(\boldsymbol{x}_n^{(m)}|\boldsymbol{x}_n^{(o)}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ which will also be a Gaussian. For this problem, this is essentially the CP that we compute in the E step.

   In the M step, we compute MLE of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. The equations will be identical to the case when there are no missing features, except that the missing parts of $\boldsymbol{x}_n$ will be replaced by the corresponding expectations. Basically, in these equations, terms like $\boldsymbol{x}_n^{(m)}$ and $\boldsymbol{x}_n^{(m)}\boldsymbol{x}_n^{(m)\top}$ will have to be replaced by the corresponding expectations, which are easy to compute since $p(\boldsymbol{x}_n^{(m)}|\boldsymbol{x}_n^{(o)}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is Gaussian. For example, $\boldsymbol{\mu}_{MLE} = \frac{1}{N}\sum_{n=1}^N [\boldsymbol{x}_n^{(o)}, \mathbb{E}[\boldsymbol{x}_n^{(m)}]]$. Likewise, you can compute $\boldsymbol{\Sigma}_{MLE}$.

2. 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. Also, a particular type of clustering algorithm arises in this special case. What is that algorithm?

   Ans: The posterior will be $p(\boldsymbol{z} = k|\boldsymbol{x}, \Theta) \propto p(\boldsymbol{z} = k)p(\boldsymbol{x}|\boldsymbol{z} = k, \Theta) = \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \sigma_k^2\mathbf{I})$. Normalizing this expression will give us the posterior probability of $\boldsymbol{z}$ taking value k

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

   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).

   Since, in this situation, we are basically making a hard assignment of each point to its closest cluster, this special variant of GMM is equivalent to the K-means clustering algorithm.

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022

3. Consider modeling an $N \times K$ binary matrix $\mathbf{Z}$ with a prior such that its binary entries are assumed to be generated i.i.d. as follows

$$\begin{aligned} \pi_k &\sim \text{Beta}(\alpha/K, 1) \qquad k = 1, \ldots, K \\ Z_{nk}|\pi_k &\sim \text{Bernoulli}(\pi_k) \qquad n = 1, \ldots, N, k = 1, \ldots, K \end{aligned}$$

- Derive the expression for the marginal prior $p(\mathbf{Z}|\alpha)$ after integrating out the π_k's.
- Derive the expression for $p(Z_{nk}|Z_{-nk})$ where Z_{-nk} denotes all the entries of $\mathbf{Z}$, except Z_{nk}. What will $p(Z_{nk}|Z_{-nk})$ be as $K \to \infty$? What does the result mean intuitively?
- (Bonus: 5 marks) As a function of α, what will be the expected number of ones in each column of $\mathbf{Z}$, and in all of $\mathbf{Z}$?

- To get $p(\mathbf{Z}|\alpha)$, we need to marginalize out π_k. Since the entries in $\mathbf{Z}$ are i.i.d. given π_k,

$$p(\mathbf{Z}|\alpha) = \prod_{k=1}^{K} \int \prod_{n=1}^{N} p(Z_{nk}|\pi_k) p(\pi_k|\alpha) d\pi_k = \prod_{k=1}^{K} \int \pi_k^{\sum_{n=1}^{N} Z_{nk}} (1-\pi_k)^{(N-\sum_{n=1}^{N} Z_{nk})} \times \frac{\pi_k^{\alpha/K-1}}{\mathcal{B}(\alpha/K, 1)} d\pi_k$$

  where $\mathcal{B}$ denotes the beta function. Simplifying the above and using the observation that the integral in the numerator is the normalization constant of a beta distribution, we get the following

$$p(\mathbf{Z}|\alpha) = \prod_{k=1}^{K} \frac{\mathcal{B}(\alpha/K + \sum_{n=1}^{N} Z_{nk}, N + 1 - \sum_{n=1}^{N} Z_{nk})}{\mathcal{B}(\alpha/K, 1)} = \prod_{k=1}^{K} \frac{\mathcal{B}(\alpha/K + m_k, N + 1 - m_k)}{\mathcal{B}(\alpha/K, 1)}$$

  where m_k denotes the number of 1s in the k-th column of $\mathbf{Z}$.

- To get $p(Z_{nk}|Z_{-nk})$, we can use one of the two ways: (1) Express $p(Z_{nk}|Z_{-nk})$ as a ratio using the Bayes rule, i.e., $p(Z_{nk}|Z_{-nk}) = \frac{p(Z_{nk}, Z_{-nk})}{Z_{-nk}} = \frac{p(\mathbf{Z})}{p(Z_{-nk})}$ where $p(\mathbf{Z})$ is given by the expression in the previous part (ratio of two beta functions), and $p(Z_{-nk})$ also has an identical expression except leaving out the entry Z_{nk} from the expression of $p(\mathbf{Z})$.
  (2) Another way to get $p(Z_{nk}|Z_{-nk})$ would be to use marginalization as follows

$$p(Z_{nk}|Z_{-nk}) = \int p(Z_{nk}|\pi_k) p(\pi_k|Z_{-nk}) d\pi_k$$

  Note that $p(Z_{nk}|\pi_k)$ is Bernoulli(π_k) and $p(\pi_k|Z_{-nk}) = p(\pi_k|Z_k)$ where Z_k denotes the k-th column's entries of the matrix $\mathbf{Z}$ but excluding Z_{nk}. It is easy to see that $p(\pi_k|Z_{-nk})$ is also a beta posterior given by Beta$(\alpha/K + m_k, N + 1 - m_k)$ (think of this as computing the posterior of a coin's probability of heads π_k, given N observations represented by the entries of the k-th column of $\mathbf{Z}$. With either of the appraoches (1 or 2), you can verify that $p(Z_{nk} = 1|Z_{-nk}) = \frac{\alpha/K + m_{-n,k}}{\alpha/K + N}$, where $m_{-n,k}$ is the number of 1s in the k-th column of $\mathbf{Z}$, excluding the entry Z_{nk}. As $K \to \infty$, $p(Z_{nk} = 1|Z_{-nk}) = \frac{m_{-n,k}}{N}$, which means that the prior probability that $Z_{nk} = 1$ given all other entries of the k-th column is proportional to how many other entries in the k-th column are nonzero.

- For the bonus part, you need to compute $\mathbb{E}[\sum_{n=1}^{N} Z_{nk}]$ and $\mathbb{E}[\sum_{n=1}^{N} \sum_{k=1}^{K} Z_{nk}]$. Use the linearity of expectation and the result of the expectation of a Bernoulli random variable with the probability parameter π_k integrated out. The results will be $\frac{N\alpha/K}{\alpha/K+1}$ and $\frac{N\alpha}{\alpha/K+1}$, respectively (try as an exercise).

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022

4. Consider a regression problem where we are modeling count-valued responses using a Poisson distribution $p(y_n|\boldsymbol{x}_n, \boldsymbol{w}) = \text{Poisson}(y_n|\exp(\boldsymbol{w}^\top \boldsymbol{x}_n))$, $n = 1, \ldots, N$. Assume a Gaussian prior on the weights, i.e., $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}\mathbf{I})$ with λ known. The goal is to infer the posterior of $\boldsymbol{w}$. Given two choices for the inference algorithm, MH sampling and SGLD, which one would you prefer for this problem and why? Give a sketch of your algorithm of choice and give the necessary update equations.

   Answer: Note that the model does not have conjugacy (since Poisson and Gaussian are not conjugate to each other). Although you can, in principal use MH sampling, it will require a careful choice of proposal distribution, and computing acceptance probabilities in each MCMC iteration.

   A better and more efficient choice for this model would be SLGD. As we know, doing SGLD is almost as straightforward as using SGD to find the MAP estimate of $\boldsymbol{w}$. Doing SGLD would require writing down log-posterior which is the sum of the log likelihood (log of $\text{Poisson}(y_n|\exp(\boldsymbol{w}^\top \boldsymbol{x}_n))$ in this case) and the log prior (log of $\mathcal{N}(\boldsymbol{w}|0, \lambda^{-1}\mathbf{I})$ in this case), and taking derivatives of the resulting expression. This will give us the gradients needed by SGLD. Once we have the gradient, you can plug these into the SLGD update equations.

   The gradient of the log-posterior distribution for this model would be (skipping the calculations which are straightforward)

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

   When using SGLD, we will only use a minibatch of the data for computing the above gradient.

   Using these gradient expressions, we can easily get the SGLD updates.

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022

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), and $\text{Gamma}(x|a,b) = \frac{1}{\Gamma(a)b^a} x^{a-1} \exp(-\frac{x}{b})$ (shape and scale 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: $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\Big\{-\frac{1}{2}\text{trace}\left[\boldsymbol{\Sigma}^{-1}\mathbf{S}\right]\Big\}$, $\mathbf{S} = (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top$.
  Information form: $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-D/2}|\boldsymbol{\Lambda}|^{1/2} \exp\left[-\frac{1}{2}\left(\boldsymbol{x}^\top\boldsymbol{\Lambda}\boldsymbol{x} + \boldsymbol{\xi}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi} - 2\boldsymbol{x}^\top\boldsymbol{\xi}\right)\right]$ where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$
- 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 $\boldsymbol{x}$, $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{x}]^\top + \text{cov}[\boldsymbol{x}]$

Name:

Roll No.: **Dept.**:

IIT Kanpur
CS772A (PML)
End-sem Exam
Date: November 21, 2022