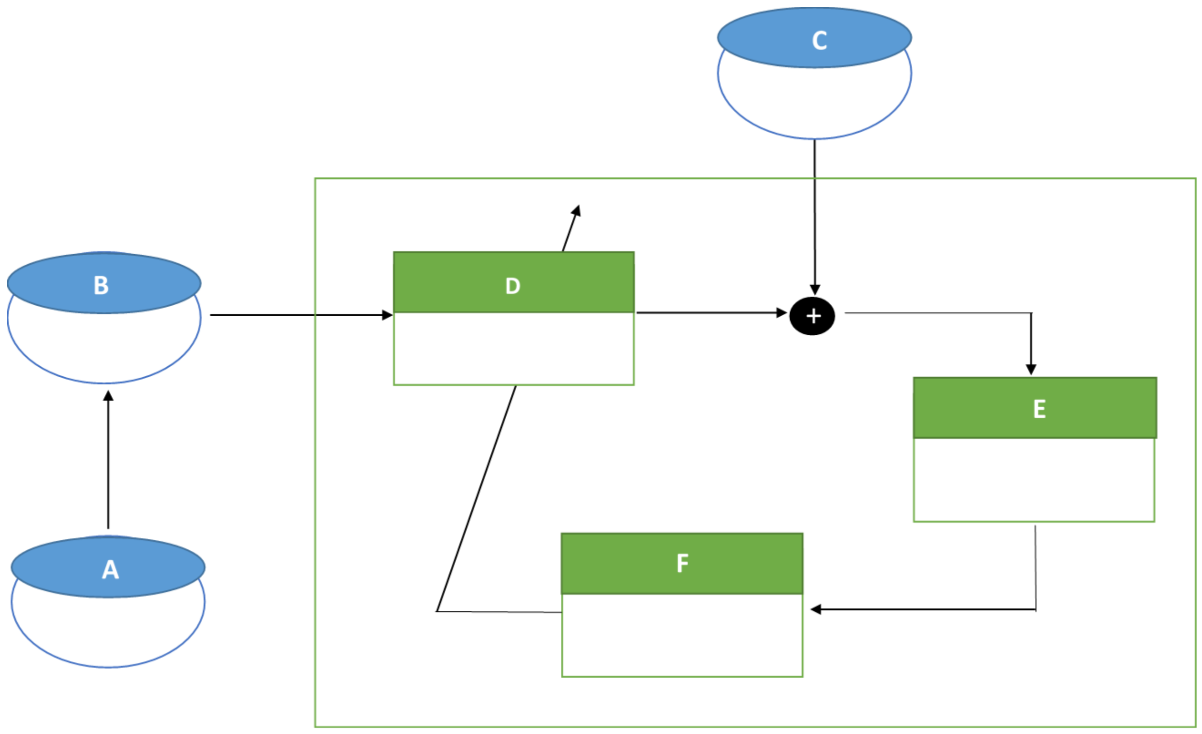
EEL5840/EEE4773 Fundamentals of Machine Learning **Summer 2022**
Midterm Exam **Solutions**

1. (10 points) **Consider the block diagram for a supervised learning system depicted below. Name each block and explain in words the function of each block.**



- A: Input Space, $\mathbf{X} = \{x_i\}_{i=1}^N$. This block includes data acquisition.
- B: Feature Space, $\phi(x)$. This block will select or extra features to create the feature space.
- C: Target, $\mathbf{t}$. This block will contain the target labels associated with each sample in order to compute an error value.
- D: Mapper or Model, $y = f(\phi(x), \mathbf{w})$. This block defines the mapper function as a function of the (unknown) parameters $\mathbf{w}$ and features $\phi(x)$.
- E: Objective function, $J(\mathbf{w})$. This block is responsible for evaluating the selected objective function given the mapper's output y and the target label t.
- F: Learning algorithm, $\mathbf{w}^{(t+1)}$. This block is responsible for updating the values of the parameters $\mathbf{w}$ in the mapper function such that the objective function is optimized.

2. (5 points) **In class, we saw that the Bayesian interpretation of an objective function with regularization is equivalent to maximizing a data likelihood distribution times a prior probability. Depending on the objective function**

we choose, we may end up with specific probabilistic models for the data likelihood and the prior.

Why is the Bayesian interpretation useful in machine learning? What advantages does this Bayesian interpretation bring when performing point/parameter estimation?

The Bayesian interpretation of the objective function provides a *different* perspective of the optimization problem. Put simply, it illustrates how the least squares objective function with a regularizer penalty term is equivalent to the Maximum A Posteriori (MAP) point estimation, where the data likelihood and prior probability have a specific probabilistic model.

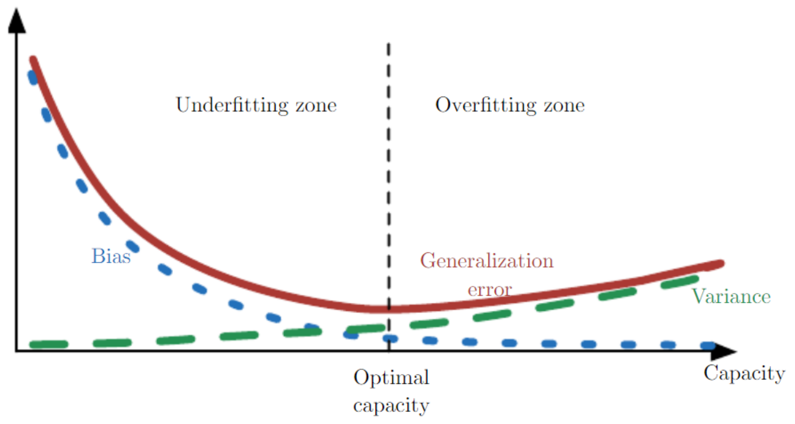
This interpretation allows us to *think* about the objective function in the Bayesian sense, thus now we can pick any probabilistic model for the data likelihood and prior distributions, those that fit the data better (alleviating Gaussianity assumptions) and encoding prior beliefs that fit the problem better as well.

Additionally, with the Bayesian interpretation, we are able to make online updates of the prior distributions provided that we are working with a conjugate prior relationship. This online update would not be possible with the "objective function interpretation", because it would require us to *hard-code* the penalty term. But in the Bayesian interpretation, we are able to start with a non-informative or incorrect prior and update it as we collect more data.

3. (10 points) **True (T) or False (F)?**

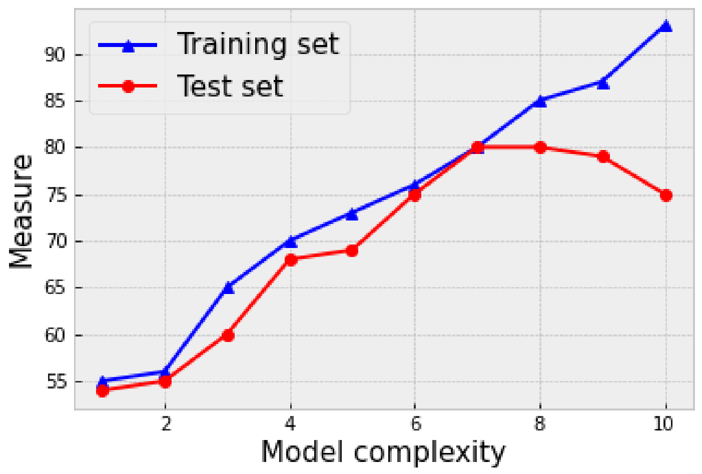
   F **Put simply, the curse of dimensionality states that we should never work with high-dimensional data.** The curse of dimensionality illustrates the difficulty of working with high-dimensional data, where samples are pushed to the corners of the features space with no longer a sense of *neighborhood*. It illustrates as well the need of exponentially more data as the dimensionality of the feature space increases. The curse of dimensionality motivates the use of feature selection or dimensionality reduction techniques to reduce the space in case the sample size is relatively small.

   T **When a model is overfitting, it has high variance but low bias. When a model is underfitting, it has low variance but high bias.**



[Image source: chapter 5, page 128 from ”Deep learning” by Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016.]

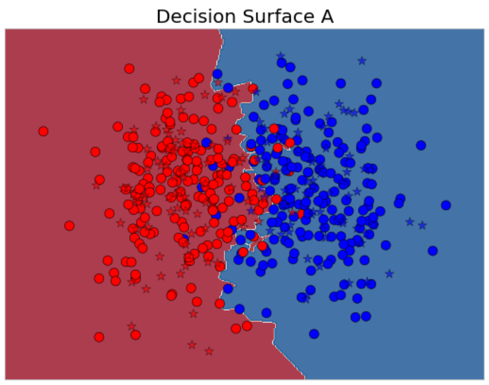
F **The figure below illustrates the experimental design carried over the training/test set in order to select the model complexity. Based on this figure, the performance measure in the y-axis must be a measure of the error.**

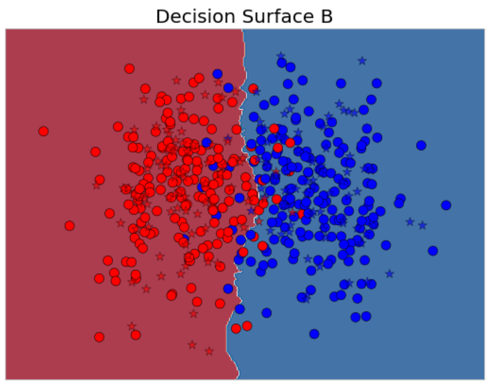


We see that, as the complexity increases, the measure in training increases steadily, where in test, the performance initially increases but then starts to decrease. This is the point where the model starts overfitting. Since the performance is increasing, it cannot be a measure of the error, otherwise the error would decrease for the training data as the model complexity increases.

T **The figure below shows two decision surfaces obtained from training a K-Nearest Neighbors classifier with the training set (circles) and evaluate prediction on test set (stars). One was obtained for $k = 20$ and the other for $k = 3$. Decision surface A used $k = 3$.**

A smaller number of neighbors, k, in KNN will produce a decision surface that is much more sensitive to the location of the training samples.





T **When performing clustering with the K-Means algorithm using Mahalanobis distance or with Gaussian Mixture Models with full covariance matrices, we should be able to learn elliptical-shaped clusters.**

The Mahalanobis distances models the covariance of groups, allowing for clusters to form elliptical-shaped clusters. Similarly, the GMM with full covariance matrices, will allow us to learn the variance spread along each feature and also between pairs of features. Thus also forming elliptical-shaped clusters.

F **The Naïve Bayes classifier is a type of a discriminative classifier.**

The Naïve Bayes classifier is a generative classifier. It models each class with a data likelihood density function instead of partitioning the feature space with a discriminant function.

F **A data likelihood described by the Bernoulli distribution with parameter ρ, $P(x|\rho) = \rho^x(1-\rho)^{1-x}$, and the Geometric prior distribution on the parameter ρ, $P(\rho|\lambda) = \lambda(1-\lambda)^\rho$, form a conjugate prior relationship.**

The posterior probability resulting from this data likelihood and prior setup is:

$$\begin{aligned}\mathcal{L} &\propto \left(\prod_{i=1}^{N} \rho^{x_i}(1-\rho)^{1-x_i}\right)\lambda(1-\lambda)^\rho \\ &= \rho^{\sum_{i=1}^N x_i}(1-\rho)^{N-\sum_{i=1}^N x_i}\lambda(1-\lambda)^\rho\end{aligned}$$

The posterior and the prior do not have the same parametric form, therefore Bernoulli-Geometric do not form a conjugate prior relationship.

T **A ROC (Receiver Operating Characteristic) curve is a binary (2-class) performance metric.**

A ROC curve is computed using FPR and TPR for a specific class in a one-vs-all approach.

F **Consider the following confusion matrix:**

The False Positive Rate (FPR) for class C_1 is $\frac{5}{150}$, for class C_2 is $\frac{6}{150}$ and for class C_3 is $\frac{9}{100}$.

		Predicted labels		
		C_1	C_2	C_3
True labels	C_1	96	3	1
	C_2	3	42	5
	C_3	6	2	42

		Predicted	
		C_1	$\overline{C_1}$
True	C_1	96	4
	$\overline{C_1}$	9	91

		Predicted	
		C_2	$\overline{C_2}$
True	C_2	42	8
	$\overline{C_2}$	5	115

		Predicted	
		C_3	$\overline{C_3}$
True	C_3	42	8
	$\overline{C_3}$	6	144

The binary confusion matrices for all classes are as follows:

Thus, the FPR for class C_1 is $\frac{9}{100}$, for class C_2 is $\frac{5}{150}$ and for class C_3 is $\frac{6}{150}$.

F **The Expectation-Maximization (EM) algorithm is implemented in three steps: initial guess for the hidden latent variables Z, expectation step and maximization step.**

The EM algorithm only has 2 steps: the E-step and the M-step.

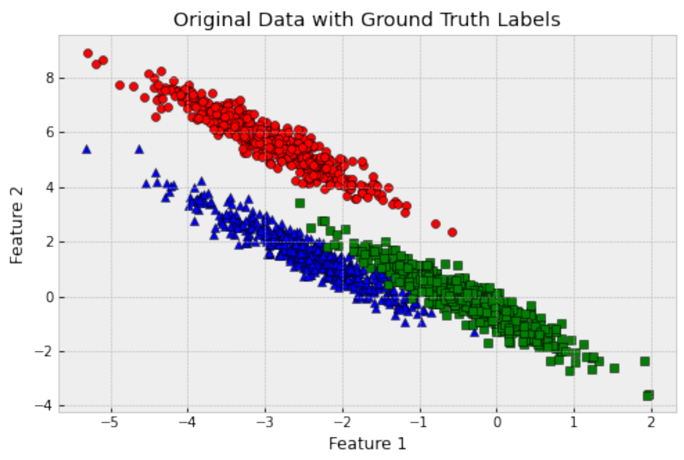
4. (10 points) **Write down the pseudo-code to implement the K-Nearest Neighbors classifier with a weighted distance voting scheme. Make sure to include all necessary steps.**

   The KNN algorithm is as follows:

   1. Normalize the data. This is an important step because KNN uses distances to measure similarity, and distances are sensitive to feature scaling.
   2. For every point in test x_i, compute the distance to every point in training. This will result in a distance matrix of size Ntest-by-Ntrain.
   3. For each test point, identify the K closest neighbors, $\{x_{ij}\}_{j=1}^{K}$, based on the distances computed earlier.
   4. Compute the score for each class using: $s_l = \sum_{j\in l} \frac{1}{d(x_i, x_{ij})}$, where l is a class label (e.g. for 3 classes, $l = \{1, 2, 3\}$).
   5. The test point x_i belongs to the class l for which its score s_l is largest.

5. (10 points) **Consider the following dataset with 3 classes:**

   Suppose that you will run three different clustering algorithms and hope to arrive at the ground truth labels shown in the figure above. Given the choices (1) K-Means with Euclidean distance, (2) Gaussian Mixture Models with full covariance or (3) Gaussian Mixture Models with isotropic covariance matrix, which one would you choose to perform clustering with this dataset? Justify your selection and elaborate why the other 2 would not be a good choice for this dataset.



This dataset contains 3 elliptical-shaped clusters. In order to converge to a clustering partition as close as the ground truth presented in the image above, we need to be able to learn 3 classes each with an elliptical shape.

The K-Means clustering algorithm with the Euclidean distance will force the clusters to be circular. For this dataset, the blue/triangles and green/squares cluster will most likely be split incorrectly to fit a circular cluster.

Similary, Gaussian Mixture Models (GMMs) with isotropic covariances are forced to learn clusters with circular shape since isotropic covariances are diagonal with the same element along the diagonal.

From these 3 options, the only algorithm that would work is GMM with full covariance matrices. When we allow GMM to learn a full covariance, we will be able to learn the variance for each feature and how pairs of features are covarying.

6. (20 points) **Suppose you have a training set with N data points $\{x_i\}_{i=1}^N$, where $x_i \in \mathbb{Z}_0^+$ (set of nonnegative integers - $0, 1, 2, 3, \dots$). Assume the samples are independent and identically distributed (i.i.d.), and each sample is drawn from a Geometric random variable with probability mass function:**

$$P(x|\rho) = \rho(1-\rho)^x$$

Moreover, consider the Beta density function as the prior probability on the success probability, ρ,

$$P(\rho|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\rho^{\alpha-1}(1-\rho)^{\beta-1}$$

Answer the following questions:

(a) (5 points) **Derive the maximum likelihood estimate (MLE) for the rate parameter ρ. Show your work.**

The observed data likelihood is:

$$\mathcal{L}^0 = \prod_{i=1}^{N} \rho(1-\rho)^{x_i} = \rho^N (1-\rho)^{\sum_{i=1}^N x_i}$$

The log-likelihood is given by:

$$\mathcal{L} = \ln \mathcal{L}^0 = N \ln(\rho) + \left(\sum_{i=1}^N x_i\right) \ln(1-\rho)$$

We can now find the MLE estimation for λ:

$$\frac{\partial \mathcal{L}}{\partial \rho} = 0 \iff \frac{N}{\rho} - \frac{\sum_{i=1}^N x_i}{1-\rho} = 0 \iff N - N\rho - \rho \sum_{i=1}^N x_i \iff \rho_{MLE} = \frac{N}{N + \sum_{i=1}^N x_i}$$

(b) (5 points) **Derive the maximum a posteriori (MAP) estimate for the rate parameter ρ. show your work.**

The observed data likelihood for MAP is:

$$\begin{aligned}\mathcal{L}^0 &\propto \left(\prod_{i=1}^N \rho(1-\rho)^{x_i}\right) \rho^{\alpha-1}(1-\rho)^{\beta-1} \\ &= \rho^N (1-\rho)^{\sum_{i=1}^N x_i} \rho^{\alpha-1}(1-\rho)^{\beta-1} \\ &= \rho^{N+\alpha-1}(1-\rho)^{\sum_{i=1}^N x_i + \beta - 1}\end{aligned}$$

The log-likelihood is given by:

$$\mathcal{L} = (N + \alpha - 1)\ln(\rho) + \left(\sum_{i=1}^N x_i + \beta - 1\right) \ln(1-\rho)$$

We can now find the MAP estimation for λ:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \rho} = 0 &\iff \frac{N+\alpha-1}{\rho} - \frac{\sum_{i=1}^N x_i + \beta - 1}{1-\rho} = 0 \\ &\iff N + \alpha - 1 - \rho\left(N + \alpha + \sum_{i=1}^N x_i + \beta - 2\right) = 0 \\ &\iff \rho_{MAP} = \frac{N+\alpha-1}{\sum_{i=1}^N x_i + N + \alpha + \beta - 2}\end{aligned}$$

(c) (5 points) **Is the Beta distribution a conjugate prior for the success probability, ρ, of the Geometric distribution? Why or why not?**

From the previous problem, we compute the posterior probability as:

$$\mathcal{L}^0 = \rho^{N+\alpha-1}(1-\rho)^{\sum_{i=1}^N x_i+\beta-1}$$

We see that this posterior is proportionally equal to the parametric form of the prior distribution $P(\rho|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\rho^{\alpha-1}(1-\rho)^{\beta-1}$, where

$$\alpha \longleftarrow \alpha + N$$
$$\beta \longleftarrow \beta + \sum_{i=1}^N x_i$$

(d) (5 points) **Suppose you would like to update the Beta prior distribution and the MAP point estimation in an online fashion, as you obtain more data. Write the pseudo-code for the online update of the prior parameters. In your answer, specify the new values for the parameters of the prior.**

The pseudo-code for online update of the prior is as follows:

1. Start at iteration $t = 0$. Initialize the prior parameters $\alpha^{(t)}$ and $\beta^{(t)}$.
2. Compute the parameter estimation for the current prior probability:

$$\lambda_{\text{MAP}} = \frac{N+\alpha-1}{\sum_{i=1}^N x_i + N + \alpha + \beta - 2}$$

3. Update the prior parameters

$$\alpha^{(t+1)} \leftarrow \alpha^{(t)} + N$$
$$\beta^{(t+1)} \leftarrow \beta^{(t)} + \sum_{i=1}^N x_i$$

4. Increment iteration counter

$$t \leftarrow t+1$$

7. (10 points) **Consider a training set containing nonnegative integer samples ($x \in \mathbb{Z}_0^+$, i.e., $x \in \{0,1,2,3,\dots\}$) for 3 classes, C_1, C_2 and C_3. The training set has 50 samples for class C_1, 100 for C_2 and 50 for C_3. Your goal is to train a Naïve Bayes Classifier with Geometric-distributed data likelihoods:**

$$
\begin{aligned}
P(x|C_1) &\sim \text{Geometric}\left(\rho_1 = \frac{1}{2}\right) \\
P(x|C_2) &\sim \text{Geometric}\left(\rho_2 = \frac{1}{7}\right) \\
P(x|C_3) &\sim \text{Geometric}\left(\rho_3 = \frac{3}{4}\right)
\end{aligned}
$$

This means that you can write each data likelihood using the following equation:

$$P(x|C_i) = \rho_i(1-\rho_i)^x$$

Answer the following questions:

(a) (3 points) **Compute the prior probability for each class.**

The prior probability for each class can be calculated as the relative frequency. Let $P(C_i)$ define the prior probability for class C_i, then we have:

$$
\begin{aligned}
P(C_1) &= \frac{50}{50+100+50} = \frac{1}{4} \\
P(C_2) &= \frac{100}{50+100+50} = \frac{1}{2} \\
P(C_3) &= \frac{50}{50+100+50} = \frac{1}{4}
\end{aligned}
$$

(b) (7 points) **Consider the test point $x = 2$. Which class will it be assigned to? Show your work.**

The Naïve Bayes Classifier uses the Bayes' rule to assign a label to every point in test, namely, point x will be assigned to class C_i if the posterior for class C_i is maximized. The posterior for class C_i is defined as $P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)}$ where $P(x)$ can be described with the Law of Total Probability as $P(x) = \sum_{j=1}^{3} P(x|C_j)P(C_j)$. Thus, we have:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{\sum_{j=1}^{3} P(x|C_j)P(C_j)}$$

We are given the test point $x = 2$. We can compute all of these terms:

$$
\begin{aligned}
P(x=2|C_1) &= \frac{1}{2} \times \left(\frac{1}{2}\right)^2 = \frac{1}{8} \\
P(x=2|C_2) &= \frac{1}{7} \times \left(\frac{6}{7}\right)^2 = \frac{36}{343} \\
P(x=2|C_3) &= \frac{3}{4} \times \left(\frac{1}{4}\right)^2 = \frac{3}{64}
\end{aligned}
$$

and

$$
\begin{aligned}
P(x=2) &= P(x=2|C_1)P(C_1) + P(x=2|C_2)P(C_2) + P(x=2|C_3)P(C_3) \\
&= \frac{1}{8} \times \frac{1}{4} + \frac{36}{343} \times \frac{1}{2} + \frac{3}{64} \times \frac{1}{4} \\
&= \frac{1}{32} + \frac{36}{686} + \frac{3}{256}
\end{aligned}
$$

Then, the posterior probabilities for each class are:

$$
\begin{aligned}
P(C_1|x=2) &= \frac{\frac{1}{8} \times \frac{1}{4}}{\frac{1}{32} + \frac{36}{686} + \frac{3}{256}} \approx 0.3274 \\
P(C_2|x=2) &= \frac{\frac{36}{343} \times \frac{1}{2}}{\frac{1}{32} + \frac{36}{686} + \frac{3}{256}} \approx 0.5498 \\
P(C_3|x=2) &= \frac{\frac{3}{64} \times \frac{1}{4}}{\frac{1}{32} + \frac{36}{686} + \frac{3}{256}} \approx 0.1228
\end{aligned}
$$

We verify that $\sum_{j=1}^{3} P(C_j|x=2) = 1$.
Since the posterior for class C_2 is largest (0.5498), we assign test point $x = 2$ to class C_2.

8. (25 points) **Use the Expectation-Maximization (EM) algorithm to solve for the parameters of an Exponential Mixture Model given a set of training data $\mathbf{X} = \{x_i\}_{i=1}^{N}$, where $x_i \geq 0, \forall i$. Recall the form of the Exponential probability density function is $P(x|\lambda) = \lambda e^{-\lambda x}$ for $x \geq 0$. Answer the following questions:**

   (a) (3 points) **Assuming your data is i.i.d., write down the observed data likelihood, $\mathcal{L}^0$.**

   The observed data likelihood is:

$$
\mathcal{L}^0 = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \lambda_k e^{-\lambda_k x_i}
$$

(b) (3 points) **Can you introduce hidden latent variables Z to this problem? Describe precisely what they are.**

Yes, we can introduce the hidden latent variable Z, where z_i corresponds to the Exponential component from which sample x_i was drawn from. Moreover, since we have a total of K Exponential components, $z_i \in \{1, 2, \dots, K\}$.

(c) (4 points) **For the hidden variables you defined above, write down the complete data likelihood, $\mathcal{L}^c$.**

The complete data likelihood is given by:

$$\mathcal{L}^c = \prod_{i=1}^{N} \pi_{z_i} \lambda_{z_i} e^{-\lambda_{z_i} x_i}$$

(d) (5 points) **Write down the EM optimization function, $Q(\Theta, \Theta^t)$, where $\Theta = \{\pi_k, \lambda_k\}_{k=1}^{K}$. Your final solution should contain the sum of simple (and simplified) log-terms.**

The EM optimization function is given by:

$$\begin{aligned}
Q(\Theta, \Theta^t) &= \mathbb{E}_z[\ln(\mathbf{L}^c)|\mathbf{X}, \Theta^t] \\
&= \sum_{z_i=1}^{K} \ln(\mathbf{L}^c) P(z_i|x_i, \Theta^t) \\
&= \sum_{k=1}^{K} \left[\sum_{i=1}^{N} (\ln(\pi_k) + \ln(\lambda_k) - \lambda_k x_i) \right] P(z_i = k|x_i, \Theta^t) \\
&= \sum_{k=1}^{K} \left[\sum_{i=1}^{N} (\ln(\pi_k) + \ln(\lambda_k) - \lambda_k x_i) \right] C_{ik}
\end{aligned}$$

where $C_{ik} = P(z_i = k|x_i, \Theta^t)$.

(e) (5 points) **Derive the update equations for the parameters λ_k.**

The solution for the parameter λ_k is:

$$\begin{aligned}
\frac{\partial Q(\Theta, \Theta^t)}{\partial \lambda_k} &= 0 \\
\sum_{i=1}^{N} \left(\frac{1}{\lambda_k} - x_i \right) C_{ik} &= 0 \\
\frac{\sum_{i=1}^{N} C_{ik}}{\lambda_k} &= \sum_{i=1}^{N} x_i C_{ik} \\
\lambda_k &= \frac{\sum_{i=1}^{N} C_{ik}}{\sum_{i=1}^{N} x_i C_{ik}}
\end{aligned}$$

(f) (5 points) **Derive the update equations for the parameters π_k.**

In order to find the solution for π_k, we must add the constraint $\sum_{k=1}^{K} \pi_k = 1$ to the optimization function:

$$Q_\pi(\Theta, \Theta^t) = Q(\Theta, \Theta^t) - a\left(\sum_{k=1}^{N} \pi_k - 1\right)$$

where $a \geq 0$ is the Lagrange multiplier. The solution for π_k can then be solved by optimizing Q_π:

$$\begin{aligned}\frac{\partial Q_\pi(\Theta, \Theta^t)}{\partial \pi_k} &= 0\\ \sum_{i=1}^{N} \frac{1}{\pi_k} C_{ik} - a &= 0\\ \pi_k &= \frac{1}{a}\sum_{i=1}^{N} C_{ik}\end{aligned}$$

Since $\sum_{k=1}^{K} \pi_k = 1$, we find that: $\sum_{k=1}^{K} \frac{1}{a} \sum_{i=1}^{N} C_{ik} = 1 \iff \frac{1}{a}\sum_{i=1}^{N}\sum_{k=1}^{K} C_{ik} = 1 \iff \frac{1}{a}\sum_{i=1}^{N} 1 = 1 \iff \frac{N}{a} = 1 \iff a = N$. Hence, the solution for π_k is:

$$\pi_k = \frac{\sum_{i=1}^{N} C_{ik}}{N}$$