# Intro to Machine Learning (CS771A, Autumn 2020)
# Bonus Assignment
# Due Date: December 15, 2020 (11:59pm)

## Instructions:

- Only electronic submissions will be accepted. Your main PDF writeup must be typeset in LaTeX (please also refer to the "Additional Instructions" below).
- The PDF writeup containing your solution has to be submitted via Gradescope `https://www.gradescope.com/`
- We have created your Gradescope account (you should have received the notification). Please use your IITK CC ID (not any other email ID) to login. Use the "Forgot Password" option to set your password.

## Additional Instructions

- We have provided a LaTeX template file `bonus.tex` to help typeset your PDF writeup. There is also a style file `ml.sty` that contain shortcuts to many of the useful LaTeX commends for doing things such as boldfaced/calligraphic fonts for letters, various mathematical/greek symbols, etc., and others. Use of these shortcuts is recommended (but not necessary).
- Your answer to every question should begin on a new page. The provided template is designed to do this automatically. However, if it fails to do so, use the `\clearpage` option in LaTeX before starting the answer to a new question, to *enforce* this.
- While submitting your assignment on the Gradescope website, you will have to specify on which page(s) is question 1 answered, on which page(s) is question 2 answered etc. To do this properly, first ensure that the answer to each question starts on a different page.
- Be careful to flush all your floats (figures, tables) corresponding to question $n$ before starting the answer to question $n+1$ otherwise, while grading, we might miss your important parts of your answers.
- Your solutions must appear in proper order in the PDF file i.e. solution to question $n$ must be complete in the PDF file (including all plots, tables, proofs etc) before you present a solution to question $n+1$.

## Problem 1 (20 marks)

**(PCA using an alternate way)** Suppose we wish to do PCA for an $N \times D$ matrix $\mathbf{X}$ and assume $D > N$. The traditional way to do PCA is to compute the eigenvectors of the covariance matrix $\mathbf{S} = \frac{1}{N}\mathbf{X}^\top\mathbf{X}$ (assuming centered data). Show that, if someone instead gives you an eigenvector $\boldsymbol{v} \in \mathbb{R}^N$ of the matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^\top$, you can use it to get an eigenvector $\boldsymbol{u} \in \mathbb{R}^D$ of $\mathbf{S}$. What is the advantage of this way of obtaining the eigenvectors of $\mathbf{S}$?

## Problem 2 (40 marks)

**(EM for Poisson Mixture Model)** Recall that a Poisson distribution is a distribution over positive count values; for a count $k$ with parameter $\lambda$, the Poisson has the form $p(k \mid \lambda) = \frac{1}{e^\lambda}\frac{\lambda^k}{k!}$. We saw (mid-sem exam problem) that the MLE for $\lambda$ given a sequence of counts $k_1, \ldots, k_N$ was simply $\frac{1}{N}\sum_n k_n$ – the mean of the counts.

Let's consider an generalization of this: the Poisson mixture model. Believe it or not, this is actually used in web server monitoring. The number of accesses to a web server in a minute typically follows a Poisson distribution.

Suppose we have $N$ web servers we are monitoring and we monitor each for $M$ minutes. Thus, we have $N \times M$ counts; call $k_{n,m}$ the number of hits to web server $n$ in minute $m$. Our goal is to *cluster* the web servers according to their hit frequency. Construct a Poisson mixture model for this problem and derive the EM algorithm by working out the expectation and maximization steps for this model. Note that here we are basically clustering data where each observation is an $M$ dimensional vector of counts.

Hint: Suppose we want to cluster the data into $L$ clusters; let $z_n$ be the latent variable telling us which cluster web server $n$ belongs to (from one to $L$). Let $\lambda_l$ denote the parameter for the Poisson for cluster $l$. Then, the complete data likelihood for each point $n$ should look pretty close to the Gaussian case, but with a product of Poissons, rather than a multivariate Gaussian. For all the observations, this looks something like:

$$p(\boldsymbol{k}, \boldsymbol{z} \mid \boldsymbol{\lambda}, \boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{l=1}^{L}\left[p(z_n = \ell)\prod_{m=1}^{M}\text{Poisson}(k_{n,m} \mid \lambda_l)\right]^{\mathbf{1}[z_n = l]}$$

Here, $\mathbf{1}[z_n = l]$ is one if $z_n = l$ and zero otherwise (if using the one-hot notation for $z_n$, we will have $z_{nl} = 1$ and all other components of the vector $z_n$ will be zero), $\boldsymbol{k} = \{\boldsymbol{k}_n\}_{n=1}^N$, is the observed data with the $n$-th observation denoted as $\boldsymbol{k}_n = \{k_{n1}, \ldots, k_{nM}\}$, $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ denotes the cluster ids of each of the $n$ observations, $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_L\}$ are the parameters of each of the $L$ Poisson distributions in the mixture, and $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_L\}$ denote the mixing proportions (these sum to 1). The goal is to estimate $\boldsymbol{z}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\pi}$ using EM. Using a similar recipe as we used for GMM, you are basically required to do the following:

- Write down the complete data log likelihood for the model.
- Estimate each $z_{nl}$ in the E step (unnormalized expression for $z_{nl}$ is fine).
- Estimate each $\pi_l$ and $\lambda_l$ in the M step by maximizing the expected complete data log-likelihood.

## Problem 3 (100 marks)

**(A Latent Variable Model for Regression)** Latent variable models can also be designed for supervised learning problems. Assume you are given $N$ training examples $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, with each $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \mathbb{R}$.

**(Part 1)** Assume the following generative story for each $(\boldsymbol{x}_n, y_n)$: (1) Generate $z_n \sim \text{multinoulli}(\pi_1, \ldots, \pi_K)$, (2) Generate the inputs $\boldsymbol{x}_n \sim \mathcal{N}(\mu_{z_n}, \Sigma_{z_n})$, and (3) Generate the outputs as $y_n \sim \mathcal{N}(\boldsymbol{w}_{z_n}^\top \boldsymbol{x}_n, \beta^{-1})$.

- Briefly explain (in 50-100 words, and may use figure(s) if needed) what this regression model is doing and in what ways it is different from a standard linear regression model we have studied. In particular, what type of input-output relationships do you expect this model to learn? **(10 marks)**

- Derive an EM algorithm for this model. In particular, the EM algorithm will compute the *conditional* posterior distribution of the latent variables $\mathbf{Z} = \{z_1, \ldots, z_N\}$ and point estimate (MLE) of the parameters $\Theta = \{\pi_k, \mu_k, \Sigma_k, \boldsymbol{w}_k\}_{k=1}^K$ of this model. Assume $\beta$ to be fixed. To do so, first write down the expression for the complete-data log-likelihood (CLL) for the model, and simplify it (ignore the constants). Now derive the necessary expressions that you would need for the EM algorithm for this model. If some of these derivations are obvious/familiar to you, you can skip those and only write down the key expressions. Also write down the overall EM algorithm. **(40 marks)**

- Briefly explain why the form of the update equation of each regression weight vector $\boldsymbol{w}_k$ makes intuitive sense. **(5 marks)**

- Assuming $\pi_k = 1/K, \forall k$, derive the ALT-OPT algorithm for this model (you may use the results from the above EM algorithm to get the ALT-OPT algorithm directly, without deriving from scratch). The ALT-OPT algorithm will compute point estimates for both $\mathbf{Z}$ and $\Theta$. Also give a brief sketch of the overall ALT-OPT algorithm. **(15 marks)**

**(Part 2)** Now let's consider a variation of the above model in which we will make two key changes: (1) We will not model the inputs $\boldsymbol{x}_n$ but will treat them simply as given; and (2) we will assume the probability vector of the multinoulli is input-specific, i.e., $z_n \sim \text{multinoulli}(\pi_1(\boldsymbol{x}_n), \ldots, \pi_K(\boldsymbol{x}_n))$, with each $\pi_k(\boldsymbol{x}_n)$ defined as a softmax function $\frac{\exp({\eta_k}^\top \boldsymbol{x}_n)}{\sum_{\ell=1}^K \exp({\eta_\ell}^\top \boldsymbol{x}_n)}$.

Derive an EM algorithm for this model. In particular, the EM algorithm will compute the *conditional* posterior distribution of the latent variables $\mathbf{Z} = \{z_1, \ldots, z_N\}$ and point estimate (MLE) of the parameters $\Theta = \{\eta_k, \boldsymbol{w}_k\}_{k=1}^K$ of this model. Assume $\beta$ to be fixed. Can you obtain closed form update for all parameters of this model in the M step? Justify your answer. **(30 marks)**