Topics in Probabilistic Modeling and Inference (CS698X)
Homework 2 (Due date: March 11, 2018, 11:59pm)

Instructions

- We will only accept electronic submissions and the main writeup must be as a PDF file. If you are handwriting your solutions, please scan the hard-copy and convert it into PDF. Your name and roll number should be clearly written at the top. In case you are submitting multiple files, all files must to be zipped and **submitted as a single file** (named: your-roll-number.zip). Please do not email us your submissions. Your submissions have to be uploaded at the following link: `https://tinyurl.com/yd3lkftz`).
- Each late submission will receive a 10% penalty per day for up to 3 days. No submissions will be accepted after the 3rd late day.

Problem 1 (10 marks)

For any generative model $p(\boldsymbol{x}|\theta)$ where $\boldsymbol{x}$ denotes data and θ denotes the distribution parameters, it is possible to define a natural notion of "similarity" (i.e., a kernel) between any two inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$ under $p(\boldsymbol{x}|\theta)$ as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{g}(\boldsymbol{x}, \theta)^\top \mathbf{F}^{-1} \boldsymbol{g}(\boldsymbol{x}', \theta)$$

where $\boldsymbol{g}(\boldsymbol{x}, \theta) = \nabla_\theta \log p(\boldsymbol{x}|\theta)$ and $\mathbf{F} = \mathbb{E}[\boldsymbol{g}(\boldsymbol{x}, \theta)\boldsymbol{g}(\boldsymbol{x}, \theta)^\top]$ and the expectation is w.r.t. the distribution $p(\boldsymbol{x}|\theta)$.

Now let's assume $p(\boldsymbol{x}|\theta)$ to be a Gaussian distribution with mean vector μ and *fixed* covariance matrix Σ (since the covariance is fixed, $\theta = \mu$). For this Gaussian distribution, compute the expression $k(\boldsymbol{x}, \boldsymbol{x}')$ for similarity between two inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$. Does the expression make intuitive sense? What would this be equal to if the covariance matrix Σ is an identity matrix?

Problem 2 (20 marks)

Consider the following model for the joint distribution of input $\boldsymbol{x}$ and output y for a regression problem

$$p(\boldsymbol{x}, y) = \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{x} - \boldsymbol{x}_n, y - y_n)$$

where $f(\boldsymbol{x}, y)$ denotes another joint probability density function over an input-output pair, and $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ is any arbitrary set of N such input-output pairs. Assume $f(\boldsymbol{x}, y)$ to be a Gaussian joint distribution over $[\boldsymbol{x}, y]$ with zero mean vector and spehrical covariance matrix $\sigma^2\mathbf{I}$.

- Derive the expression for $p(y|\boldsymbol{x})$ and show that it can be written as a weighted mixture of univariate Gaussians. Give expressions for the mixture weights and the mean/variance of these Gaussians. (12 marks)
- Using the predictive distribution $p(y|\boldsymbol{x})$ obtained in the above part, compute the mean $\mathbb{E}[y|\boldsymbol{x}]$ and variance $\text{var}[y|\boldsymbol{x}]$. If your derivations are correct, these quantities will be somewhat similar (may not be exactly) in form to what we obtained for Gaussian Process regression model seen in the class. (8 marks)

In your derivations, feel free to use properties of multivariate Gaussians (e.g., conditional, marginal, etc.)

Problem 3 (20 marks)

The gamma p.d.f. is defined as $\text{Gamma}(x|a,b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}$, where a and b are the shape and rate parameters, respectively, and $\Gamma(a)$ denotes the gamma function.

- Approximate this gamma distribution by a Gaussian using Laplace approximation.
- How does the Laplace approximation compare with approximating $\text{Gamma}(x|a,b)$ by a Gaussian whose mean and variance are equal to the mean and variance, respectively, of $\text{Gamma}(x|a,b)$. Under what condition would these two approximations be roughly the same?
- Using the Laplace approximation, approximate the gamma function $\Gamma(a)$. Plot this approximation as well as the true $\Gamma(a)$ as function of a and comment on the accuracy of this approximation.

Problem 4 (10 marks)

Consider a supervised learning problem with training data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, with each input $\boldsymbol{x}_n \in \mathbb{R}^D$ and each response being a count-valued non-negative integer, i.e., $y_n \in \mathbb{Z}_+$. Since Poisson distribution is a common choice for modeling count data, let us use a Poisson distribution to model the conditional distribution of y_n given $\boldsymbol{x}_n$. Suggest the exact form of this model (i.e., the likelihood model) with the necessary parameter(s).

Is maximum likelihood estimation (MLE) possible to do here in closed form (as in linear regression)? Justify your answer and sketch the procedure to obtain the MLE solution, giving the required expressions.

Problem 5 (20 marks)

Consider binary classification with training data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, with $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \{0,1\}$. Assume each binary label to be generated as $y_n = \mathbb{I}[z_n > 0]$, where z_n is a Gaussian latent variable with $p(z_n|\boldsymbol{w}, \boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n, 1)$, $\boldsymbol{w} \in \mathbb{R}^D$, and $\mathbb{I}[.]$ returns 1 if the condition is true, and 0 otherwise. The model parameter here is $\boldsymbol{w}$ and your goal here to do point estimation for $\boldsymbol{w}$.

Develop an EM algorithm to do this. In particular, write down the expected CLL expression, and derive the expressions for all the relevant quantities that are estimated in the E step (i.e., posteriors for latent variables, the expectations you would need in the M step, etc.) and the M step update equations (MLE expression for $\boldsymbol{w}$). Sketch the overall EM algorithm as well.

Note: In deriving the EM algorithm for this model, you would most likely need to compute the expectation of a truncated normal distribution. You may visit the Wikipedia entry to look up the desired expression.

Problem 6 (20 marks)

Given N observations $\{\boldsymbol{x}_n\}_{n=1}^N$, where $\boldsymbol{x}_n \in \mathbb{R}^D$, consider a latent factor model for each observation $\boldsymbol{x}_n = \mathbf{W}\boldsymbol{z}_n + \epsilon_n$. Here $\mathbf{W}$ is a a $D \times K$ matrix of reals and noise ϵ_n to be Gaussian, i.e., $p(\epsilon_n) = \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I})$. Unlike standard latent factor models such as Probabilistic PCA which assume $\boldsymbol{z}_n$ to be a real-valued vector of length K, here we will assume $\boldsymbol{z}_n \in \{0,1\}^K$ and assume each entry of $\boldsymbol{z}_n$ to have a prior $p(z_{nk}|\pi_k) = \text{Bernoulli}(\pi_k)$.

Note that this model is representing an observation $\boldsymbol{x}_n$ as a "subset sum" of the K columns of $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$, plus the added Gaussian noise. The subset to be added up depends on the binary vector $\boldsymbol{z}_n$. Such models are quite common in sparse dictionary learning, recommender systems, etc. (later during the semester, we will look at this model again and see how we can even learn K for such a model using a fully Bayesian setting).

Derive an EM algorithm to estimate the latent variables and parameters of the model. For this model, the latent variables are $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]$ and the parameters are $\{\mathbf{W}, \{\pi_k\}_{k=1}^K, \beta\}$. In your solution, you must clearly write down the expression for the expected CLL and simplify it similar to how we did for GMM, PPCA, MoE, etc. For the latent variables, you should give the expression for their posterior distribution, their required expectations, etc. (the procedure should be similar to what we used for GMM, PPCA, MoE, etc), and for the parameters, you should derive the MLE updates. Finally, also give a brief sketch of the overall EM algorithm.