E9 205 – Machine Learning For Signal Processing

Practice Midterm Exam
Date: Feb 23, 2022, 3:30pm

Instructions

1. This exam is open book. However, computers, mobile phones and other handheld devices are not allowed.
2. Any reference materials that are used in the exam (other than materials distributed in the course channel) should be pre-approved with the instructor before the exam.
3. No additional resources (other than those pre-approved) are allowed for use in the exam.
4. Academic integrity and ethics of highest order are expected.
5. Notation - bold symbols are vectors, capital bold symbols are matrices and regular symbols are scalars.
6. Answer all questions.
7. Name your scanned copy of answer file in pdf format as FirstName-LastName-midterm.pdf and follow the upload to the Teams channel.
8. All answer sheets should contain your name and SR number in the top.
9. Question number should be clearly marked for each response.
10. Total Duration - **90 minutes including answer upload**
11. Total Marks - **100 points**

1. **MLSP Exam and grading** - Prof. Raj is evaluating the midterm exam of the MLSP course which was taken by N students. The exam had Q questions. From the answers provided by students, he finds the assignment variable x_{nq} where $(x_{nq} = 1)$ indicates that the answer for student n and question q was correct and $(x_{nq} = 0)$ indicates answer for student n and question q was incorrect. Here $n \in \{1, .., N\}$ and $q \in \{1, .., Q\}$. Each question is assigned a latent difficulty δ_q and each student is associated with a latent ability α_n. Prof. Raj uses a sigmoidal model for the conditional probability of the assignment variable $(x_{nq} = 1)$ given the latent ability vector $\boldsymbol{\alpha} = [\alpha_1, .., \alpha_N]^T$ and latent difficulty vector $\boldsymbol{\delta} = [\delta_1, ..., \delta_Q]^T$. Specifically,

$$p(x_{nq} = 1|\boldsymbol{\alpha}, \boldsymbol{\delta}) = \sigma(\alpha_n - \delta_q)$$

where σ is the sigmoidal nonlinearity function. He plans to estimate the deterministic latent parameters in the model given the binary data matrix $\boldsymbol{X}$ of dimension $N \times Q$ containing elements $[x_{nq}]$ (assuming that variables x_{nq} are i.i.d.).

   (a) Find the total data likelihood under the given model for the MLSP exam.

   (b) How can Prof. Raj apply gradient based learning to estimate the latent ability of students α_n and latent difficulty of questions δ_q which maximize the total log-likelihood ?

(**Points 15**)

2. **Bayesian Machine Learning** - Varada is a budding stock market analyst. At the outset of her job, she attempts modeling the market data denoted as $\mathbf{X} = \{\mathbf{x}_i, i = 1, ..., N\}$ using a GMM $\lambda = \{\alpha_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}_{c=1}^C$. Given only a small number of samples ($N = 30$), she decides on using a simple covariance matrix $\boldsymbol{\Sigma}_c = \mathbf{I}$. At this point, her colleague Vikas asks her to apply Bayesian techniques instead of ML for parameter estimation of the Gaussian mixture model. For mixture component means, Vikas suggests using a prior density given by Gaussian density,

$$p(\boldsymbol{\mu}_c) \propto \exp\{-\frac{\rho_c}{2}(\boldsymbol{\mu}_c - \boldsymbol{m}_c)^*(\boldsymbol{\mu}_c - \boldsymbol{m}_c)\}$$

where $\rho_c > 0$ and $\boldsymbol{m}_c$ are the hyper parameters of the Gaussian distribution. Vikas also suggests to not use any prior distribution for weight parameters and to assume the hyper-parameters as fixed quantities. Let $\boldsymbol{\Theta} = \{\alpha_c, \boldsymbol{\mu}_c\}_{c=1}^C$ denote the parameters of interest for estimating the model. While the ML rule maximizes $\arg\max_{\boldsymbol{\Theta}} \; p(\mathbf{X}|\boldsymbol{\Theta})$, the Bayesian estimation suggested by Vikas maximizes the MAP rule $\arg\max_{\boldsymbol{\Theta}} \; p(\boldsymbol{\Theta}|\mathbf{X})$ which is equivalent to maximizing $\arg\max_{\boldsymbol{\Theta}} \; p(\mathbf{X}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})$.

   (a) The first challenge before Varada is modify the EM algorithm (which maximizes the ML objective) to optimize the MAP rule for Bayesian estimation. How would you derive an iterative algorithm for Bayesian estimation. Show that your algorithm

consistently improves the objective function at each iteration. (**Points 10**)

(b) Varada has managed to develop the EM algorithm for Bayesian estimation. Now she does some mathematical analysis and to her delight finds that her choice of prior distributions for $\boldsymbol{\mu}_c$ obeys the conjugate density property (the EM style lower bound for the posterior distribution is also a Gaussian distribution like the prior distribution). How did she arrive at this property ? What are the mean and covariance of the EM style posterior Gaussian distribution. (**Points 15**)

(c) Using the above problem solutions, Varada proceed to derive the iterative update rules for $\boldsymbol{\Theta} = \{\alpha_c, \boldsymbol{\mu}_c\}_{c=1}^{C}$. Can you perform the same derivation ? (**Points 10**)

3. **Revisiting LDA** - The LDA is the problem of finding projection matrix $\mathbf{W}$ which maximizes $Tr(\frac{\mathbf{W}^T\mathbf{S}_b\mathbf{W}}{\mathbf{W}^T\mathbf{S}_w\mathbf{W}})$ for data $\mathbf{x}_1, ..\mathbf{x}_N \in \mathcal{R}^D$, where

$$\begin{aligned}\mathbf{S}_w &= \sum_{c=1}^{C}\sum_{\mathbf{x}_k \in c}(\mathbf{x}_k - \boldsymbol{\mu}_c)(\mathbf{x}_k - \boldsymbol{\mu}_c)^T \\ \mathbf{S}_T &= \sum_{k=1}^{N}(\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T\end{aligned}$$

with C denoting the number of classes, $\boldsymbol{\mu}_c$ being the within class mean for class c, $\boldsymbol{\mu}$ being the sample mean of the entire data and between class scatter matrix $\mathbf{S}_b = \mathbf{S}_T - \mathbf{S}_w$.
Show that LDA problem can be equivalently expressed using all the pairwise scatter matrices $\mathbf{S}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$ for all $i, j = 1, ..., N$ and the affinity matrix $\mathbf{R}$ of size $N \times N$ whose (i, j)th element r_{ij} defined as
$r_{ij} = \frac{1}{N_k}$ if $\mathbf{x}_i, \mathbf{x}_j$ belong to the kth class
$r_{ij} = 0$ otherwise.
Here, N_k is the number of data points belonging to class k. Specifically, show that,

$$\begin{aligned}\mathbf{S}_w &= \sum_{i=1}^{N}\sum_{j=1}^{N} r_{ij}\mathbf{S}_{ij} \\ \mathbf{S}_b &= \sum_{i=1}^{N}\sum_{j=1}^{N}(\frac{1}{N} - r_{ij})\mathbf{S}_{ij}\end{aligned}$$

(**Points 20**)

4. **Line Mixture Model** - A line mixture model is the problem of fitting a mixture of lines on a 2-D dataset. Let $\mathbf{z}_i = [x_i \ y_i]^T$ denote a set of 2-D data $i = \{1, .., N\}$. Each mixture component in the LMM is defined using a line $f_k(x_i) = a_k x_i + b_k, \ k = \{1, ..., K\}$, where K is the number of mixtures and a_k, b_k are the parameters of the line for the kth mixture component. The pdf of z_i is modeled as,

$$p(z_i|\lambda) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(y_i; f_k(x_i), \sigma_k^2)$$

where σ_k is the variance of the k-th mixture component and the model parameters $\lambda = \{a_k, b_k, \sigma_k\}_{k=1}^{K}$. Given a set of N data points,

(a) Write down the Q function which will allow the EM estimation of the λ.

(b) Find the iterative maximization steps for all the parameters in the model.

(**Points 15**)

5. **Robust Principal Components** - Vani is interested in applying PCA to a set of pure data $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ where each $\mathbf{x}_n$ is D dimensional. She has read in wiki that applying PCA finds the principal components as the eigenvectors of sample covarince matrix $\mathbf{S_{xx}}$ with the largest eigenvalues. However in measuring the data $\mathbf{x}_n$, she realizes that an additive noise $\boldsymbol{\epsilon}_n$ has also been introduced, i.e., the measured data with her is $\mathbf{y}_n$ where $\mathbf{y}_n = \mathbf{x}_n + \boldsymbol{\epsilon}_n$ with $\boldsymbol{\epsilon}_n$ having a sample mean of $\mathbf{0}$ and sample covariance of $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$ which is full rank. The pure data $\mathbf{x}_n$ and the noise $\boldsymbol{\epsilon}_n$ are also uncorrelated. She realizes that application of PCA on the set of noisy data $\{\mathbf{y}_1, ..., \mathbf{y}_N\}$ will not recover the principal components of the pure data. With this concern, she approaches Ryan who is a student of MLSP course. Ryan solves this problem using the Cholesky decomposition $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathbf{A}\mathbf{A}^T$. Specifically, using a data transformation $\mathbf{z}_n = \mathbf{A}^{-1}\mathbf{y}_n$, he is able to recover the prinicipal components of the pure data. How will you go about finding the prinicipal components of the pure data if you were Ryan. Further, for a PCA of K dimensions ($K < D$), how can Ryan estimate the reconstruction error in PCA using the data transformation approach. (**Points 15**)