Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning

Instructor: Purushottam Kar
Date: October 15, 2025
Total: 50 marks

MINOR ASSIGNMENT 2

---

1 What should I submit, where should I submit and by when?

Your submission for this minor assignment will be one PDF (.pdf) file only. Instructions on how to prepare and submit this file are given below.

Assignment Package:
`https://www.cse.iitk.ac.in/users/purushot/courses/ml/2025-26-a/material/assignments/minor2.zip`
Deadline for all submissions: October 26, 2025, 9:59PM IST
Code Submission: ***None needed – do not submit any code (ipynb, py, etc) files for this minor assignment.***
Report Submission: on Gradescope
There is no provision for "late submission" for this minor assignment.

1.1 How to submit the PDF report file

1. The PDF file must be submitted using Gradescope in the *group submission mode.*
2. Note that unregistered auditors cannot make submissions to this minor assignment.
3. Make only one submission per assignment group on Gradescope, not one submission per student. Gradescope allows you to submit in groups - please use this feature to make a group submission.
4. Link all group members in your group submission. If you miss out on a group member while submitting, Gradescope will think that person never submitted anything.
5. You may overwrite your group's submission as many times as you want before the deadline (submitting again on Gradescope simply overwrites the old submission).
6. Do not submit Microsoft Word or text files. Prepare your report in PDF using the style file we have provided (instructions on formatting given later).

Problem 2.1 (Melbo's Mesmerizing Mixtures)**.** Melbo finds generative models fascinating, especially how generative models using even something as simple as single Gaussians offer not just reasonable prediction accuracies, even in the presence of large number of missing pixels, but also how these simple models are able to reconstruct the missing pixels in a visually meaningful way. Melbo wishes to explore further and requests your help in doing so. For this task, you would need to refer to the following resources:

1. Calculations and code for the lecture "Generative Models" available at `https://www.cse.iitk.ac.in/users/purushot/courses/ml/2025-26-a/material/lectures/lec13.zip`

2. The MNIST dataset available at
`https://www.kaggle.com/datasets/hojjatk/mnist-dataset/data`

(50 + bonus marks)

Your Task. The following enumerates 7 parts to the question. All parts need to be answered in the PDF file containing your report. **Do not submit any code files for this minor assignment**. Thus, submit only a single PDF file as your submission. Notice that the lecture code performs generative classification with each class modeled as a single Gaussian i.e. $\mathbb{P}[\mathbf{x} \mid y = c] = \mathcal{N}(\boldsymbol{\mu}^c, \Sigma^c), c \in [C]$. We wish to make this model more powerful

1. Give detailed calculations on how to train a generative multi-classification model where each class-conditional distribution i.e. $\mathbb{P}[\mathbf{x} \mid y = c]$ is modeled as a mixture of K Gaussians
$$\mathbb{P}[\mathbf{x} \mid y = c] = \text{GMM}\left(\{\mathcal{N}(\boldsymbol{\mu}_k^c, \Sigma_k^c)\}_{k\in[K]}, \boldsymbol{\pi}^c\right), c \in [C]$$
where $\boldsymbol{\pi}^c \in \Delta^{K-1}$ is a K-dimensional histogram vector encoding the selection probability of each of the K components in the GMM for the c^{th} class. Note that when discussing GMMs in lecture slides, we often assumed the selection probability to be uniform i.e. $\boldsymbol{\pi}^c = \left[\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}\right]$. However, we should avoid making that assumption here. (10 marks)
2. Give detailed calculations on how can inference (testing) be performed with such a GMM-based generative multi-classification model on a new test point. (10 marks)
3. Extend the lecture code to perform generative classification on the MNIST dataset using GMMs for each digit instead of a single Gaussian. **Do not change the train/test splits.** Note that the lecture code is equivalent to using a GMM with $K = 1$ (i.e. a single component per GMM). **Describe major changes required to the code to do this implementation**. **Show a curve** on how **training accuracy** and **test accuracy** vary as K is varied to use small and larger values, specifically $K = 1, 2, 5, 10, 15, 20$ etc. The x-axis of the curve should be K and the y-axis should show train/test accuracy. For these experiments, we ensure that every class gets the same value of K i.e. if using $K = 5$, ensure that each one of the 10 GMMs corresponding to the 10 digit classes uses $K = 5$ components in its respective GMM. Note that the test accuracies on this part should be reported on uncensored test images. (15 marks)
4. Extend the lecture code to perform generative classification and reconstruction in the presence of missing pixels. **Do not change the manner in which pixels are omitted** i.e. do not change the method `censorImages` that sets the window of missing pixels. **Describe major changes required to the code to do this implementation**. **Show a curve** on how **test accuracy** on images with missing pixels vary as K is varied to use small and larger values, specifically $K = 1, 2, 5, 10, 15, 20$ etc. **Also show example images** of successful and unsuccessful reconstructions and correct/incorrect classifications in the presence of missing pixels. Note that only test images are censored. Training images are not censored i.e. you should reuse the models you learnt in the previous part and simply re-test on censored images. (15 marks)

Bonus Do certain classes need fewer GMMs? Do certain digits exhibit more variations than others? Explore this by fixing a reasonably large value of K and instead of giving each class the same number of K components in its GMM, see if accuracy remains the same or similar even if certain classes are given fewer than K components. Which classes seem to require a large number of components? Which classes seem to be content with fewer components? (TBD marks)

Using Internet Resources. You are allowed to refer to textbooks, internet sources, research papers to find out more about this problem and for specific derivations e.g. the arbiter-PUF problem. However, if you do use any such resource, cite it in your PDF file. There is no penalty for using external resources with attribution but claiming someone else's work (e.g. a book or a research paper) as one's own work without crediting the original author will attract penalties.

Restrictions on Code Usage. You should not use any library other than what is already imported in the Jupyter notebook in the code available at `https://www.cse.iitk.ac.in/users/purushot/courses/ml/2025-26-a/material/lectures/lec13.zip` (you do not need to anyway). Since this minor assignment does not require any code submission, we will trust you on this one. However, you may use fancier libraries like seaborn or plotly to plot graphs etc.

2 How to Prepare the PDF File

Use the following style file to prepare your report.
`https://media.neurips.cc/Conferences/NeurIPS2023/Styles/neurips_2023.sty`

For an example file and instructions, please refer to the following files
`https://media.neurips.cc/Conferences/NeurIPS2023/Styles/neurips_2023.tex`
`https://media.neurips.cc/Conferences/NeurIPS2023/Styles/neurips_2023.pdf`

You must use the following command in the preamble

```
\usepackage[preprint]{neurips_2023}
```

instead of `\usepackage{neurips_2023}` as the example file currently uses. Use proper LaTeX commands to neatly typeset your responses to the various parts of the problem. Use neat math expressions to typeset your derivations. Remember that all parts of the question need to be answered in the PDF file. All plots must be generated electronically - hand-drawn plots are unacceptable. All plots must have axes titles and a legend indicating what are the plotted quantities. Insert the plot into the PDF file using LaTeX `\includegraphics` commands.