CS589: Machine Learning - Fall 2025

Homework 7: Mixture Models and Dimensionality Reduction

Assigned: Friday, November 21st, 2025

Getting Started: This assignment consists of written and coding problems. Download the assignment archive from Canvas and unzip the file. Starter code for select coding problems is provided in the code directory. For general clarification questions, please submit public posts to Campuswire. For questions related to your specific solutions, please submit private posts on Campuswire. In-person help is available through regularly scheduled office hours.

Due Date and Late Work: This assignment is due at 8:00pm ET on December 3rd, 2025. Students can submit up to 11:59pm on the due date with no penalty. Work submitted up to 11:59pm on one day after the assignment is due is subject to a penalty of 10%. Work submitted up to 11:59pm two days after the assignment is due is subject to a penalty of 20%. Gradescope will close at 11:59pm two days after the assignment is due and work can not be submitted for credit after this point.

How to Submit: Your written report must be submitted to Gradescope as a PDF file. You must select the page on which each answer appears. You are encouraged to typeset your PDF solutions using LaTeX. The source of this assignment is provided to help you get started. You may also submit a PDF containing scans of *clear* hand-written solutions. Work that is illegible will not count for credit. For this assignment, code should be submitted to Gradescope as Python 3.10+ Jupyter Notebooks. Autograding will not be used for this assignment. You may submit both the code and the report as many times as you like before Gradescope closes. Only your final submission will be graded. Any late penalties will be based on the timestamp of your final submission as determined by Gradescope.

Academic Honesty Reminder: Homework assignments are individual work. Being in possession of another student's solutions, code, code output, or plots/graphs for any reason is considered cheating. Sharing your solutions, code, code output, or plots/graphs with other students for any reason is considered cheating. Copying solutions from external sources (books, web pages, etc.) is considered cheating. Collaboration indistinguishable from copying is considered cheating. Posting your code to public repositories like GitHub (during or after the course) is not allowed. Manual and algorithmic cheating detection are used in this class. Any detected cheating will result in a grade of 0 on the assignment for all students involved, and potentially a grade of F in the course.

Generative AI Use Reminder: The use of generative AI tools to help with coding is permitted for programming problems in this course. The use of generative AI for all other work is considered cheating.

1. (*35 points*) **Bernoulli Mixture Inference.** In this question, you will develop formulas and inference code for the Bernoulli mixture model. Assume that the data cases $\mathbf{x} \in \{0,1\}^D$ and the mixture model has K components. Use π_k as the parameter for mixture proportion for cluster k. Use ϕ_{dk} as the Bernoulli parameter for data dimension d and cluster k.

a. (*5 pts*) Provide an equation for the joint probability of the mixture model $P(\mathbf{X} = \mathbf{x}, Z = k)$ in terms of the data $\mathbf{x}$ and the model parameters. Explain your answer.

b. (*5 pts*) Provide Numpy code for computing $\log P(\mathbf{X} = \mathbf{x}, Z = k)$ given an array `x` of shape (1,D), an array of mixture proportions `pi` of shape (K,1), and a array of Bernoulli parameters `phi` of shape (K,D). Add your code to `bernoulli_mixture.ipynb` and to your report.

c. (*5 pts*) Run your code for computing $\log P(\mathbf{X} = \mathbf{x}, Z = k)$ using the provided parameters and data case for each value of k. Provide the values you find for $\log P(\mathbf{X} = \mathbf{x}, Z = k)$ in your report.

d. (*5 pts*) Provide an equation for the conditional probability of the mixture model $P(Z = k|\mathbf{X} = \mathbf{x})$ in terms of the data $\mathbf{x}$ and the model parameters. Explain your answer.

e. (*10 pts*) Provide Numpy code for computing $\log P(Z = k|\mathbf{X} = \mathbf{x})$ given an array `x` of shape (1,D), an array of mixture proportions `pi` of shape (K,1), and a array of Bernoulli parameters `phi` of shape (K,D). Add your code to `bernoulli_mixture.ipynb` and to your report.

f. (*5 pts*) Run your code for computing $\log P(Z = k|\mathbf{X} = \mathbf{x})$ using the provided parameters and data case for each value of k. Provide the values you find for $P(Z = k|\mathbf{X} = \mathbf{x})$ for each value of k in your report.

2. (*35 points*) **Gaussian Mixture Clustering**. In this problem, you will explore clustering global climate data using a Gaussian mixture model. Each data case contains a time series of daily temperature data for a different weather station. Data are included for 12,120 weather stations covering the globe. Each time series is 366 days long, covering all of 2024. Add your code for this question to `gaussian_mixture.ipynb`.

a. (*10 pts*) To begin, add code to learn the model on the training set using the `sklearn.mixture` class `GaussianMixture` with `covariance_type="diag"`, `random_state=589`, `max_iter=1000` and 16 clusters. Once the model is learned, make a bar chart showing the learned mixture proportions and include it in your report. Make sure to label axes. You will need to read the API documentation[1] for the model to extract the parameters.

b. (*10 pts*) Continuing from the model learned in (a), you will next add code to visualize the parameters associated with each cluster. Each cluster has a learned mean vector of length 366 and a learned vector of marginal variances of length 366 representing the daily mean temperature and variability around the daily mean temperature. To display the parameters for each cluster, make a filled area plot showing the 95% confidence around the daily mean temperature, and overlay it with a line plot showing the daily mean temperature. Make one such plot per cluster. Include all of the plots in your report. You will need to refer to

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html`

the API documentation for the model to extract the parameters. Useful plotting functions in `matplotlib` include `subplot` for making a 4×4 grid of plots and `fill_between` to make a filled area plot.

c. (*5 pts*) Based on the output from part (b), what structures has the model found in the data?

d. (*5 pts*) Add code to use the `predict` function of the model fit in (a) to cluster the data cases in the test set. Use the provided plotting code to plot each test weather station on a map colored by the cluster it belongs to. Include an image of the resulting map in your report as the answer to this question.

e. (*5 pts*) Describe the patterns that you see in the results from part (d). How has the model clustered the weather stations in the test set?

3. (*30 points*) **Dimensionality Reduction for Image Compression**. In this problem, you will explore using dimensionality reduction methods for image compression. To get started, select a natural image of your choosing with a resolution close to 1080p. You will apply dimensionality reduction to non-overlapping 8×8 pixel blocks of the image to compress the image representation. Add your code for this question to `image_compression.ipynb`.

a. (*5 pts*) To begin, write code to crop your image so the width and height are multiples of 8 pixels. Next, write code to convert your image from an array `I` of shape $(H, W, 3)$ to an patch image array `P` of shape $((H/8) \times (W/8), 8, 8, 3)$ containing all non-overlapping 8×8 pixel patches from `I`. In this representation, $P[i]$ corresponds to an image patch of shape $(8, 8, 3)$. Now, write code to re-shape the patch array `P` of shape $((H/8) \times (W/8), 8, 8, 3)$ to a patch vector array `V` of shape $((H/8) \times (W/8), 8 \times 8 \times 3)$. In this representation, `V[i]` represents an 8×8 pixel patch as a flat vector. As your answer to this question, apply your code to extract all patches from your image. Run the function `show_patches` on your patches and include the resulting image in your report along with your selected image.

b. (*10 pts*) Now, using the `sklearn.decomposition` class `PCA`, write code to fit PCA using your patch vector array `V` with values of K from 1 to 64. For each value of K, use the learned model to project the patch vector array `V` into K-dimensional space using PCA's `transform` method. The result is a low-dimensional representation `Z` of shape $((H/8) \times (W/8), K)$. We can determine the quality of this representation by computing a reconstruction error. The reconstruction `R` of `V` is obtained by applying PCA's `inverse_transform` method to the low-dimensional representation `Z`. As a reconstruction error, we will use the root mean squared error of the difference between `V` and `R`. As your answer to this question, include a line plot in your report showing the reconstruction error versus K. Make sure to label axes.

c. (*5 pts*) Building on part (b), we can also compute the total storage cost s for the dimensionality reduced representation of `I`. To do so, we need to add the number of elements in `Z` to the total number of parameters learned by PCA. The compression ratio achieved for each value of K is equal to the number of elements in the original image array `I` divided by s. As your answer to this question, include a line plot in your report showing the compression ratio as a function of K. Make sure to label axes.

d. (*5 pts*) Next, write code that takes an array of patch vectors `V` of shape $((H/8) \times (W/8), 8 \times 8 \times 3)$ and the values of H and W, and converts the array of patch vectors back into an image `I` of shape $(H, W, 3)$. Verify your code by converting the patch vector array for your image back into an image. There is nothing to include in your report for this question.

e. (*5 pts*) Lastly, use your code for converting patch vector arrays to images to visually inspect the reconstructions obtained using different values of K. Select the smallest value of K that you think results in a reconstruction that is perceptually indistinguishable from the original image. As your answer to this question, include in your report the reconstructed images using $K = 1$, $K = 10$, and the reconstructed image using your selected value of K. Also report the value of K you selected, the reconstruction loss it achieves, and the compression ratio that it achieves.

4. (*0 points*) If you used generative AI tools to help complete any programming questions on this assignment, please briefly describe which tools you used, how you used them, and for which problems you used them. If you did not use generative AI tools, please indicate that as your response to this question. (Note: Not answering this question will result in a deduction of 2 points.)