Homework 1

Question 1

In this problem we do a deeper examination of the Graphene Raman spectra from class (i.e., `mystery_design_matrix.npy` and `mystery_labels.npy`). We define the following variables for this question.

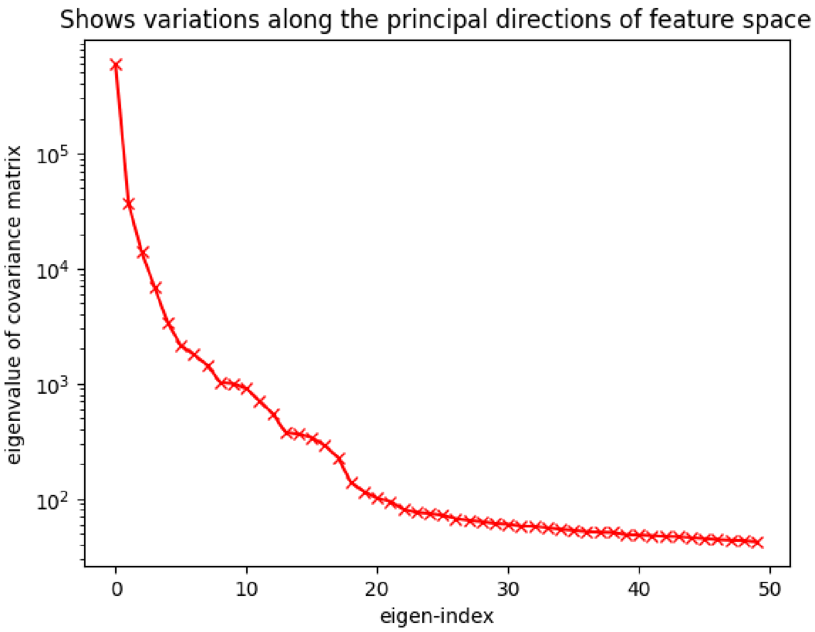
- The design matrix is $\mathbf{X} \equiv X_{nk}$;
- its scatter matrix is $\mathbf{S} \equiv S_{ij}$;
- its covariance matrix is $\mathbf{\Sigma} \equiv \mathbf{\Sigma}_{ij}$;
- its similarity matrix is $\mathbf{G} \equiv G_{nm}$.

Data provenance.

The sample was prepared and measured by Ph.D. candidate Zheng Yuntian from Professor Ariando's group at NUS Physics.

- This sample is multilayer graphene on a SiO_2/Si wafer, measured before thermal annealing.
- Although C–O peaks shouldn't appear in high-quality graphene, they often do due to residues from exfoliation or transfer processes (e.g., scotch tape, PMMA, PDMS). These C–O peaks stem from contamination, not the graphene itself.
- The oxidation level of graphene is best assessed by examining the C–O and D peaks together. Prolonged air exposure causes both peaks to emerge, indicating oxidation.
- Hydrogen annealing at high temperatures effectively suppresses the C–O peak and is typically performed before device fabrication.

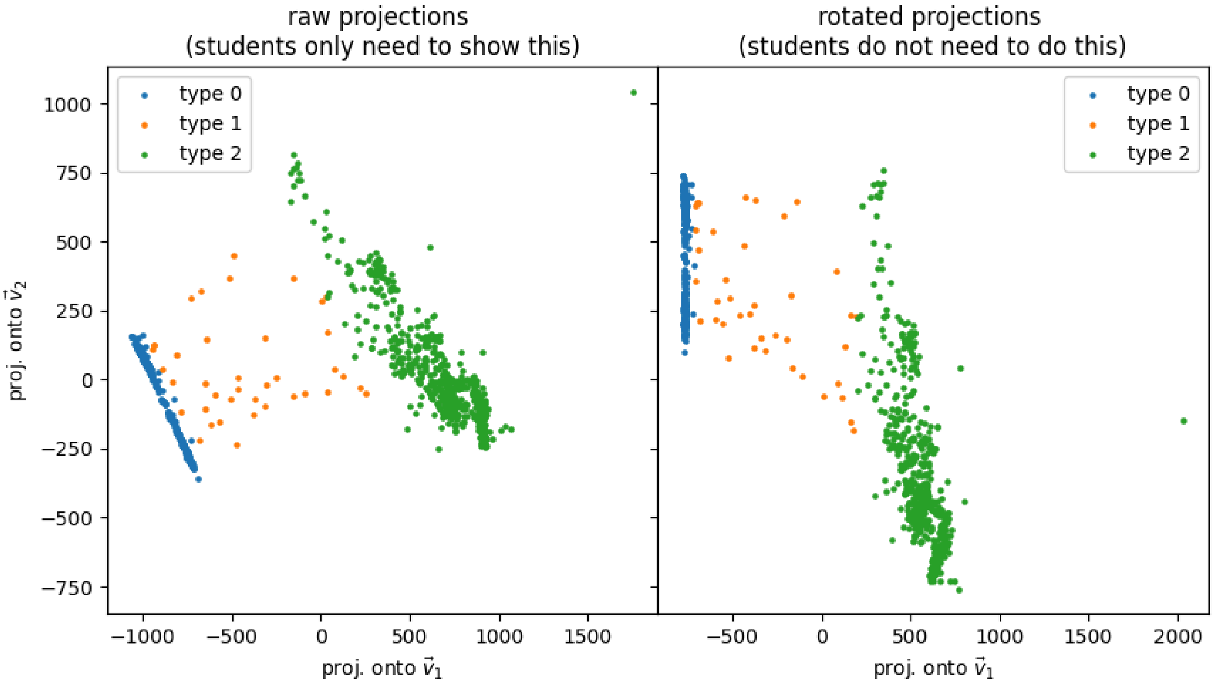
Question 1a (2 points)



The eigenvectors of the covariance matrix is closely related to the principal components of PCA (something we will learn later in the semester). Here we study the properties of the covariance matrix.

- What is the largest eigenvalue of the covariance matrix? Submit your answer on Canvas.

Question 1b (2 points)

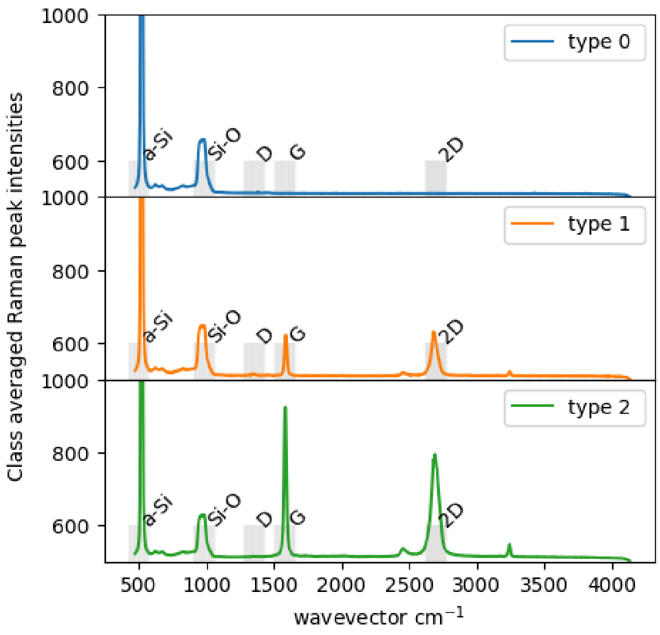


We project each Raman spectrum (i.e., row vector of design matrix) into the first two eigenvectors $\vec{v}_1$ and $\vec{v}_2$ of the covariance matrix.

Show (in the ipynb Jupyter notebook that you submit to canvas) the following:

- Plot of the projection of the mean-centered design matrix onto the first and second eigenvectors of the covariance matrix.
- How would you *approximately* cluster the spectra into three different types (as shown above). We don't expect your clustering to be identical to the solution (especially since we don't have the ground truth here). But your clustering should show approximately what is seen in the image above.

Question 1c (2 points)



Show (in the ipynb Jupyter notebook that you submit to canvas):

- The above plot of the average spectra of each of the three types that you've discovered above. You should label the **D, G, 2D** peaks.
- Again, the exact clustering of the spectra (i.e., which spectrum belongs to which class) is less important than getting approximate classes here.

Peak	Position (cm^{-1})	Activated By	Significance	Behavior in Multilayers
D	~1350	Defects	Crystallinity	Increases with disorder
G	~1580	Always (sp² C–C)	Graphitic content	Remains strong
2D	~2700	Double-resonance	Layer count, stacking	Broadens, weakens, and shifts

Question 1d (2 points)

- What are the approximate fractions of different types? Pick the closest ratios from the choices below Submit your answer on Canvas.

`type0: type1: type2`

- A) 0.19 : 0.38 : 0.42
- B) 0.04 : 0.39 : 0.57
- C) 0.12 : 0.38 : 0.50
- D) 0.33 : 0.33 : 0.33

Question 1e (2 points)

- What are the most likely samples in type 0, type 1, and type 2? Submit your answer on Canvas.

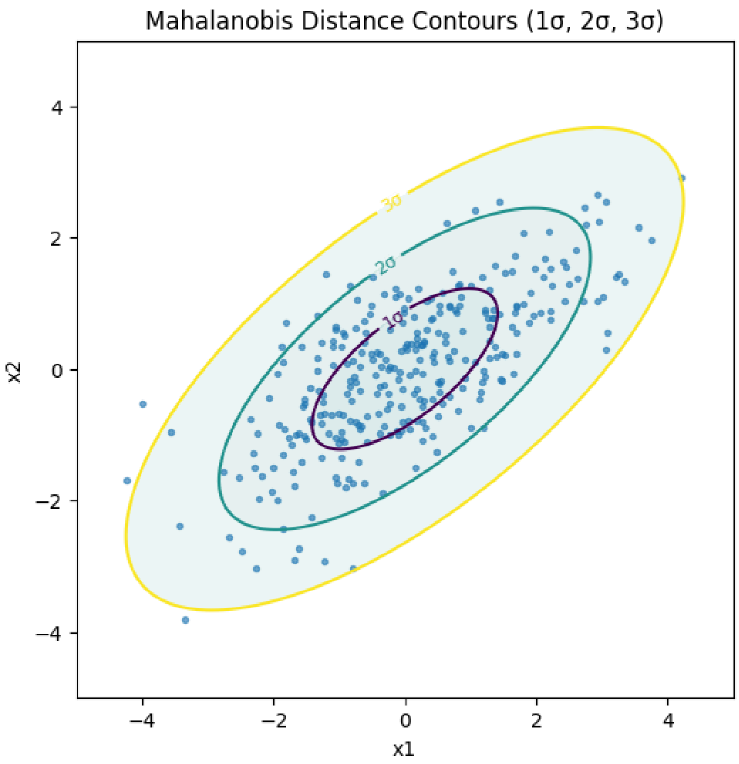
Question 2 (2 points)

Mahalanobis distance contours are ellipses aligned with the data covariance directions. **In other words, given a group of spectra (e.g., type 0 spectra) the Mahalanobis distance tells us how far each spectrum in this group is from the Gaussian that describes it.**

For a Gaussian distribution, Mahalanobis distance tells you how “many standard deviations away” a point is, taking into account correlations between variables.

In this spectroscopy example:

- You model a cluster of spectra (say, “type 0 spectra”) with a Gaussian (mean spectrum + covariance of variations).
- The Mahalanobis distance measures how far each individual spectrum is from that Gaussian “cloud.”



Setup

- You have a **design matrix** $X \in \mathbb{R}^{n \times p}$:
  each row is a measurement (observation), each column is a feature.
  So row $X_n \in \mathbb{R}^p$.
- You have the **covariance matrix** $\Sigma \in \mathbb{R}^{p \times p}$ of the data.

Mahalanobis Distance Definition

For a single observation X_n, the **Mahalanobis distance** to the mean vector $\langle X \rangle$ is:

$$d_M(X_n) = \sqrt{(X_n - \langle X \rangle)^T \, \Sigma^{-1} \, (X_n - \langle X \rangle)}$$

where:

- $\langle X \rangle = \mathbb{E}[X]$ or the sample mean,
- Σ^{-1} is the inverse covariance matrix.

Step-by-Step Computation

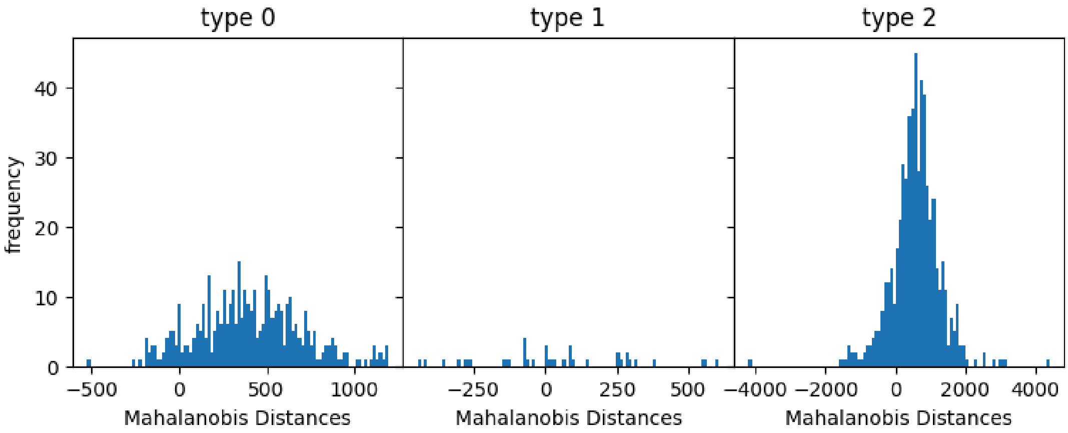
1. **Compute the sample mean.**
2. **Center the data.**
3. **Compute (or use given) covariance matrix, Σ.**
4. **Invert it (or pseudo-invert if singular), Σ^{-1}.**
5. **Compute Mahalanobis squared distances for all rows:**

$$D^2 = \operatorname{diag}\big(X_c \, \Sigma^{-1} \, X_c^T\big)$$

6. **Take square roots to get distances:**

$$d_n = \sqrt{D_n^2}$$

Question

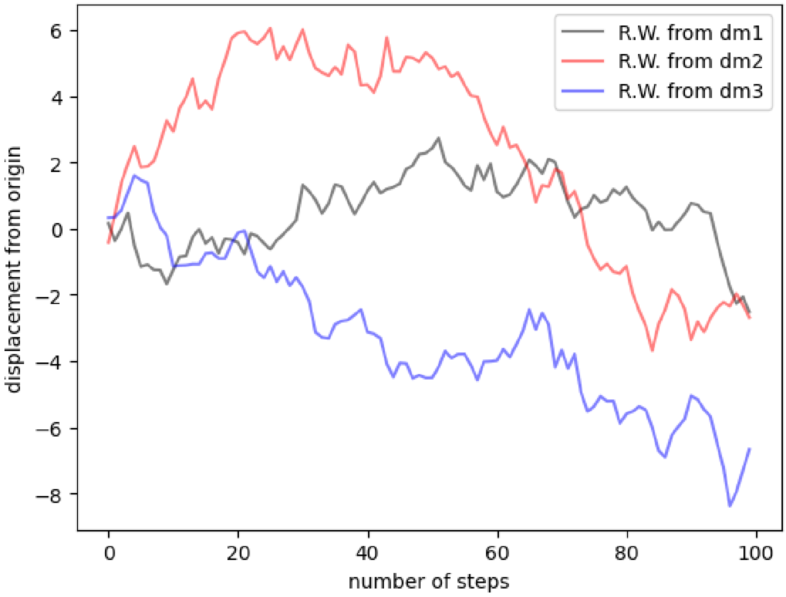


In this question, we continue from the cluster analyses in Question 1 based on the Raman spectra in `mystery_design_matrix.npy`.

Show (in the ipynb Jupyter notebook that you submit to canvas) the following:

- The histogram of the Mahalanobis distances of each spectrum from its cluster's Gaussian description.
- If each cluster's spectra belong to the same type of sample and are just random noisy versions of each other, then their squared Mahalanobis distances should follow a χ_d^2 distribution (with d = number of independent features). For large d, this distribution may appear approximately Gaussian.

Question 3 (2 points)



In class we learned that the **central limit theorem** is quite useful for deciding if two sets of observations belong to the same set of phenomena.

Here are three design matrices (i.e., `dm1.npy`, `dm2.npy`, `dm3.npy`). Each design matrix shows the 1D positions of 10,000 random walks performed by an autonomous walker. The 1D position of the object was recorded for each step it took, for a total of 100 steps. The walker would perform 10,000 such random walks (R.W.) which would be recorded into a single design matrix.

Some sample trajectories are shown above.

```
dm1 = np.load("dm1.npy")
dm2 = np.load("dm2.npy")
dm3 = np.load("dm3.npy")
```

Question.

Turns out the trajectories of all three design matrices were produced by only two unique walkers. Which of the two design matrices were made by the same walker?
Show your proof.