Applications of Machine Learning in Remote Sensing
Homework 2

John Smith – johnsmith@rit.edu

https://github.com/johnsmith/repo.git

- In your submission, include **explanation**, **results**, and **the code** for the problem in the same PDF file in form of a Jupyter Notebook Results. Also *separately*, attach solution's codes so I can replicate your results.
- Show your understanding of the problem by providing **explanation**.
- Provide sufficient commenting in your code.
- Ensure all text/images are legible and organized.
- Ensure that your code can reproduce the submitted results.

Create a directory in your repository and name it `ml`, if you already do not have. The workflows and the scripts created in this homework would go under `ml`.

Do not leave the file in your github repository for your homework as the data file is large.

Problem 1: Hyperspectral Data

Last week, you worked with Sentinel-2 multispectral data. This week, you will analyze hyperspectral data collected by the RIT MX1 drone over the RIT Tait Reserve area. The dataset is provided as `taitlabsphere` and `taitlabsphere.hdr`.

This is an ENVI headered file, which includes a corresponding metadata file (.hdr). The header file describes various aspects of the dataset, such as the coordinate system, number of lines and columns, number of spectral bands, and more. Read through the .hdr file to familiarize yourself with the metadata structure. You can also find the wavelengths for each spectral band within the header file.

(1.a)(10 points) Load the data using the Spectral Python (SPy) and envi module. What range in the electromagnetic wave this dataset covers? Select and show bands blue ($\approx 450nm$), green ($\approx 550nm$), and red ($\approx 650nm$) regions of the spectrum in one plot. Additionally, create a pseudocolor BGR image using bands from the green ($\approx 550nm$), red ($\approx 650nm$), and near-infrared (NIR; $\approx 800nm$) regions. What pops up in the pseudo color image? Lastly, plot 3 bands in 900, 950, and 1000 nm in 3 separate plots, how are these bands different visually compared to the other plots you created?

(1.b)(10 points) Compute and display the correlation matrix as an image (from the previous homework). Analyze the correlation between spectral bands—what patterns do you observe? How are the bands correlated with each other?

Problem 2: Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is often explained using **eigendecomposition or spectral decomposition** of the covariance matrix. However, due to **finite-precision arithmetic**, this approach can be numerically unstable on a computer. Instead, in practice, Singular Value Decomposition (SVD) is used to compute PCA more robustly.

(2.a)(10 points) Write a function that takes an array of size $m \times d$, (where m is the number of samples and d is the number of features). Your function should will perform PCA and should return 1) all principal components as an array, 2) eigenvalues, and 3) standardized

data (per independent variable) .

Note:

- Mean-center your data prior to passing to SVD on each feature separately.
- You may use a library for SVD calculation.
- Are the eigenvalues and corr. eigenvectors are sorted?
- Provide sufficient explanation with your answer.

```
pcs, eigenvalues, mean_arr = principal_component_analysis(array)
```

(2.b)(10 points) Using PCA as a dimensionality reduction technique, apply PCA to the provided hyperspectral dataset. Extract the first 10 principal components, use the 10 PCs as a transformation matrix to transform your data to the reduced dimension, reshape the observation dimension to match the image size, and plot first 10 PCs separately. Describe your observations—what patterns or features do you notice? Discuss what each principal component represents in the context of the hyperspectral data. Do the same with the last 10 PCs and report and explain on what you observe on the plots.

(2.c)(10 points) On the same hyperspectral dataset, plot the mean reconstruction error using the L2 distance (between the mean-centered and reconstructed data) as a function of the number of principal components. Given the large size of the image dataset, compute and visualize the error for a selected number of principal components: 1, 10, 50, 100, and d (the total number of features). Note that the reconstructed data uses the first K eigenvectors as a transformation matrix to create a reduced dimension in the PC space - the forward transform, then uses the transpose of the same transformation matrix on the lower dimensional data in the PC space to reconstruct the data in the original space using only K PCs - backward transform.

(2.d) (10 points) As you may have observed from part 2.b, PC with lower eigenvalues primarily capture noise rather than meaningful variability. PCA can be also be used as a noise reduction approach in for high dimensional data. This problem walks you through the process.

On the hyperspectral dataset, using the eigenvalues and their corresponding PCs, compute the explained variance ratio for each PC (eigenvalue of a PC divided by the sum of all eigenvalues). Keep the PCs that collectively explain 99% of the total variance, and set the remaining PCs to zero. Note that we are not performing dimensionality reduction in this practice, so the your transformation matrix should be $band \times band$. Then, apply the backward transform to reconstruct the data using the modified $band \times band$ PC matrix. In this practice, no PCs are removed—only their contributions in the eigenvector space are manipulated.

Select five interesting pixels from your image and compare their spectral signals before and after reconstruction; show this as a plot for your five pixels, make sure to name what materials you picked. Additionally, calculate and report the mean signal-to-noise ratio (SNR) $\frac{\mu}{\sigma}$ across all bands, before and after reconstruction for the entire image. Make sure to mask out no-data pixels.

(2.BONUS)(10 points) Implement the concept of impact plot discussed in Peter Bajorski's book provided as optional reading material in Week 2. Visualize how variability changes relative to the mean signal for the first five principal components in terms of positive and negative impacts.

Problem 3: K-Means Clustering

(3.a)(15 points) We discussed K-Means clustering as an unsupervised clustering technique in class. Implement K-Means from scratch `without using a library`. Below is a pseudo code:

1. initialize K cluster centers
2. Iterate until convergence or max number of iterations; new cluster centers will not change.
   (a) Assign each point's class to the Nearest Centroid
   (b) Compute new cluster centeroids

Keep in mind that K-Means calculates L2 distance across dimensions and iteratively updates cluster centers to the mean of assigned points. Since L2 distance is sensitive to scale, make sure to standardize your data before applying K-Means.

To debug your approach, use the provided test image `jellybeans.tiff`, which is smaller and easier to analyze. Based on the colors in the image, what is an appropriate choice for K?

(3.b)(15 points) In this exercise, we will use the Sentinel-2 data from Assignment 1. Provided Sentinel-2 has 12 spectral bands, but L2 distance in high-dimensional spaces (many features) can become less meaningful as we discussed. This raises the question of which features should be used for unsupervised clustering?

One approach is to select specific bands relevant to the clusters of interest based on our application. Another approach is to use PCA as a feature extraction technique.

- Apply PCA to the Sentinel-2 data using your PCA function.
- Extract the first 3, 4, 5, and 6 principal components and transform the data into these lower-dimensional representations in the PC space.

- Apply your K-Means clustering algorithm to the lower dimensional data.

(3.c)(10 points) Now that you have used PCA for feature extraction and K-Means for clustering, we will apply the same workflow to hyperspectral data from Problem 2. In this case, we have hundreds of bands, making feature selection non-trivial. Additionally, since this dataset is much larger in terms of number of samples, K-Means will be computationally expensive due to repeated distance calculations. To improve efficiency, use MiniBatch K-Means from `sklearn` library, which speeds up clustering for dataset of large sizes.

For this problem:

- Extract a 250×250 patch from the hyperspectral dataset that includes calibration panels, roads, and vegetation (regions with variability).
- Perform dimensionality reduction using PCA, transforming the data to 2, 5, 10, 50, and 100 features.
- Apply K-Means clustering to both the **lower-dimensional data** and the **original dataset** (all bands).
- Report and compare your results. How does performance change with different numbers of features? How is the performance different between lower dimensional data and the original data?