HOMEWORK 2:
UNSUPERVISED AND SUPERVISED LEARNING

ELEC 400M @ UBC
TAs: Chun-Yin Huang (Primary), Wenlong Deng, Sadegh Mahdavi

Instructions

- **Homework Submission:** Submit your code and report to Canvas. You will use Co-lab to implement the coding tasks. Please check Piazza for updates about the homework.
  - Upload a zip file containing two files: Your report in .PDF format and your notebook in .Ipynb format.
  - To ensure the reproducibility of your results: (1) set a seed for numpy and python random modules on top of your Colab notebook (2) restart and run all the cells of your notebook once before submission.
- **Collaboration policy**: The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes (including code) are shared, or are taken at that time, and provided learning is facilitated, not circumvented. The actual solution must be done by each student alone. The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved.
- **Description:** In Assignment 2, you will conduct some calculation and proofs related to Unsupervised Learning (PCA for this assignment) using elementary linear algebra and calculus knowledge. Furthermore, you will practice on programming on Unsupervised Learning (PCA) and Supervised Learning (SVM, Random Forest, etc) methods.
- **Clarification:** The questions colored in blue indicates to be answered by hand calculation. Example code and data are provided in the HW folder.

1 PCA [5 pts]

Recall that the geometric formulation of PCA for dimensionality reduction is the the optimization problem as shown below:

$$U^* = \underset{U \in \mathbb{R}^{D\times d}, U^\top U = I}{\arg\min} \|\mathbf{X} - UU^\top \mathbf{X}\|_F,$$

where $\mathbf{X} \in \mathbb{R}^{D\times M}$ is a zero-mean data matrix with $D-$dimensional features and M samples. We have $d \ll D$ is the target lower dimension. When $d = 1$, we say U^* is the first principal axis and $U^{*\top}\mathbf{X}$ is the first principal component.

Suppose we have data matrix: $\mathbf{X} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$, which contains 4 samples with 3-dimensional features. We can observe there are some redundant information in the data matrix. In this regard, we want to perform PCA on the data matrix to reduce the dimensionality of the original data.

Note: No need to centralize features to 0s in this toy example as we mentioned 'zero-mean data matrix.' But if you have centralized it, it is okay. Please do remember normalize your eigenvector it a unit vector.

(a) Find all eigenvalues of $\mathbf{X}\mathbf{X}^\top$. Please show all of your calculation steps.

(b) Find the first principal axis and the first principal component of $\mathbf{X}$.

(c) If we have another data matrix

$$\mathbf{X}' = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

which is obtained by exchanging the last two columns of $\mathbf{X}$. Do you think the first principal axis of $\mathbf{X}'$ will differ from the first principal axis of $\mathbf{X}$? If you think they are different, find the first principal axis of $\mathbf{X}'$. Otherwise, prove they are identical and state a benefit of this property.

2 Coding practice [10 pts]

There are two problems in this coding practice. In the first problem, you will study how different numbers of principal components represent the images visually. For the second problem, we will solve a price classification problem using Support Vector Machine (SVM), Random Forest, etc. Please find the details and hints in the question description and 'HW2.ipynb'.

2.1 Unsupervised dimension reduction - PCA (3pts)

For this question, we will explore the MNIST handwriting digits loaded from tensorflow. Codes are provided in 'HW2.ipynb' Problem 2.1.

(a) For $k = 0, 10, 20, 30, 40, 50$, use k-th principal components ONLY, namely, $\hat{x}_i = \bar{X} + (\phi_k^\top x_i)\phi_k$, where $\hat{x}_i$ is the reconstructed image, $\bar{X}$ is the averaged image, and ϕ_k is the k-th principle axis for MNIST 0's to approximately reconstruct the image selected above. Note that we index from 0, namely 0-th principal component is the first one. Display the reconstruction for each value of k. To display the set of images compactly, you may want to use the 'plot_images' function defined in 'HW2.ipynb' Problem 2.2.

2.2 Supervised classification (7pts)

In this question, we will be working on the dataset of model price classification. **Please use the data we provided in the 'data' folder.** The task is to classify the level of cost given the features of mobiles. In machine learning terms, it is a classification problem. The data are provided in the data folder. The training and testing data are in the different files. You can use `numpy` or `pandas` libraries to load the csv files.

The targets in the data have values:

- 0 (low cost)
- 1 (medium cost)
- 2 (high cost)
- 3 (very high cost)

which can be read from column `price_range`.

The features are the things like:

- battery_power: Total energy a battery can store in one time measured in mAh
- blue: Has bluetooth or not
- clock_speed: speed at which microprocessor executes instructions
- dual_sim: Has dual sim support or not
- fc: Front Camera mega pixels
- four_g: Has 4G or not
- ...

which can be read from the columns after dropping column `price_range`. Before feeding the features in the machine learning models, you need to do zero-mean unit variance normalization: https://en.wikipedia.org/wiki/Feature_scaling.

(a) Train a linear kernel SVM using `sklearn.svm.LinearSVC`. Please report the accuracy on testing data when choosing C in the range of $(10^{-5}, 10^{-4}, 10^{-3}, \ldots, 10^{5})$ using `matplotlib.pyplot` where x-axis is C and y-axis is accuracy (range from 0 to 1). The accuracy can be calculated using `sklearn.metrics.accuracy_score`.

(b) Train a RBF (Gaussian) kernel SVM using `sklearn.svm.SVC`. Please report the accuracy on testing data when choosing γ in the range of $(10^{-1}, 10^{0}, \ldots, 10^{4})$ using `matplotlib.pyplot` where x-axis is γ and y-axis is accuracy (range from 0 to 1). The accuracy can be calculated using `sklearn.metrics.accuracy_score`.

(c) Train a Random Forest Classifier using `RandomForestClassifier`. Please report the accuracy on testing data when choosing number of trees (`n_estimator`) in the range of $(10, 100, 500, 1000)$ using `matplotlib.pyplot` where x-axis is `n_estimator` and y-axis is accuracy (range from 0 to 1). The accuracy can be calculated using `sklearn.metrics.accuracy_score`.

(d) Please implement 5-fold cross-validation for hyper-parameter selection using `sklearn` on the training set and select the best parameters for problem 2.(a-c) separately and report the corresponding testing accuracy.

Note

1. Remember to submit your assignment by 11:59pm of Nov 03. Late submission will affect your scores.
2. If you submit multiple times, ONLY the content and time-stamp of the latest one would be considered.
3. We strictly follow the rules of UBC Academic Misconduct.