CSCI 5521: Introduction to Machine Learning (Fall 2021)[1]

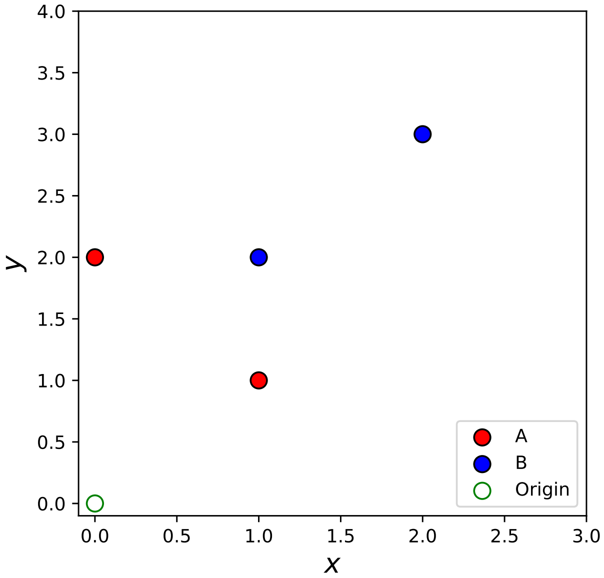
Homework 2

Due date: Oct 20, 2021 11:59pm

1. (**30 points**) Consider the following 2 sets of points in the plane:

$$A : \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ with mean } \boldsymbol{\mu}_A = \begin{pmatrix} \frac{1}{2} \\ \frac{3}{2} \end{pmatrix}$$

$$B : \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix} \text{ with mean } \boldsymbol{\mu}_B = \begin{pmatrix} \frac{3}{2} \\ \frac{5}{2} \end{pmatrix}$$



(a) What is the first principal component $\boldsymbol{w}_1$ (use the **unbiased** estimation of covariance)? Draw the first principal component direction $\boldsymbol{w}_1$ on the plot, anchored at the origin.

(b) Consider a more general case (not specific to the aforementioned samples): PCA performs linear dimensionality reduction with $\mathbf{z}^t = \mathbf{W}^T\mathbf{x}^t$, where $\mathbf{x}^t \in \mathbb{R}^D$ is the original data for the t-th sample, $\mathbf{z}^t \in \mathbb{R}^d$ is the low-dimensional projection ($d < D$), $\mathbf{W} \in \mathbb{R}^{D\times d}$ is the PCA projection matrix (each column is a principal component). Professor HighLowHigh claims that we can reconstruct the original data with $\mathbf{v}^t = \mathbf{W}\mathbf{z}^t$, so that $\forall_t \ \mathbf{v}^t = \mathbf{x}^t$. Is the claim correct? Explain your answer with necessary details (you can use formulations if it helps explain).

(c) Compute the Within-Class Scatter matrix $\boldsymbol{S_W}$ and Between-Class Scatter matrix $\boldsymbol{S_B}$.

(d) What is the Fisher projection direction $\boldsymbol{w}$ found by the Fisher Linear Discriminant Analysis(LDA)? Normalize $\boldsymbol{w}$ to have unit length, that is $||\boldsymbol{w}||_2 = 1$, and draw such $\boldsymbol{w}$ on the plot, anchored at the origin.

[1] Instructor: Catherine Qi Zhao. TA: Shi Chen, Xianyu Chen, Helena Shield, Jinhui Yang, Yifeng Zhang. Email: csci5521.f2021@gmail.com

Note: You need to show all your calculations in order to get full credits.

2. (**30 points**) Given the following data points in 1D: $x_1 = 2, x_2 = 4, x_3 = 5, x_4 = 6, x_5 = 8, x_6 = 10, x_7 = 12, x_8 = 14$, perform k-means clustering algorithm for $k = 2$.

   (a) Start from initial cluster centers $c_1 = 1$, $c_2 = 10$. Show your steps for all iterations: (1) the cluster assignments $y_1, \cdots, y_8$; (2) the updated cluster centers at the end of that iteration.

   (b) How many iterations does it take for k-means algorithm to converge (*i.e.,* number of iterations includes all iterations you perform to find convergence)? What is the reconstruction error (*i.e.*, distortion measure J, equation 9.1 of the Bishop's textbook) at the end of that iteration?

   (c) Repeat the above steps with initial cluster centers $c_1 = 6, c_2 = 12$.

   (d) How many iterations does it take for k-means algorithm to converge in this case? What is the reconstruction error at the end of that iteration?

   (e) Comparing (a) with (c), which solution is better? Why?

3. (**40 points**) In this programming exercise you will implement k-means and Principal Component Analysis algorithms:

   (a) You will first apply k-means algorithm ($K = 8$) to the provided dataset `Digits089.csv`. The dataset contains 3000 samples, where each sample has 784 features. The first column of the data file contains flags that are not used in this assignment, the second column includes class labels (*i.e.,* 0, 8, 9), and the rest of the columns store features. Your program should initialize the centers with a set of pre-selected samples (see the template for detailed initialization), iteratively update the center of each cluster based on the input samples, and record the reconstruction error after each iteration. After reaching convergence, it should measure the quality of clustering results by computing the information entropy for each cluster with its distribution of class labels:

$$H(X_k) = -\sum_{c=0,8,9} P(X_k^c) \log_2 P(X_k^c) \tag{1}$$

   where X_k is the distribution of labels for the k_{th} cluster, $P(X_k^c)$ denotes the proportion of samples from class c.

   Report the number of iterations for convergence and the average information entropy across different clusters. Plot the history of reconstruction errors. Is the plot shape following what you expect? The code for plotting is included in `hw2.py`, and you do not need to modify the file.

   (b) Repeat the above, but use low-dimensional data obtained from PCA. Your PCA algorithm should reduce the original samples to dimensions needed to capture

$> 95\%$ of the variance. **How many dimensions are necessary in this case? Does PCA help clustering? Explain.** (Hint: Consider both the information entropy and the number of iterations for convergence.)

(c) Repeat (b), but using only the first principal component. **Are the results better? Explain.**

We have provided the skeleton code `Mykmeans.py` and `MyPCA.py` for implementing the algorithms. To verify your implementation, call the main function `hw2.py`, which automatically generates plots for the reconstruction errors.

Submission

- **Things to submit:**

  1. hw2_sol.pdf: a document containing all your answers for the written questions (including answers and plots for problem 3).
  2. `Mykmeans.py`: a Python source file containing your implementation of the k-means algorithm with header `class Kmeans`. Use the skeleton file `Mykmeans.py` found with the data on the class web site, and fill in the missing parts. The `run_kmeans` function should take the samples and class labels as inputs, initialize the centers, update the model parameters and record the reconstruction errors. It should also return the number of iterations for convergence, the history of reconstruction errors, and the average information entropy. The `compute_error` function should take the samples and cluster assignment (a vector specifying the cluster ID assigned to each sample) as inputs, and return the reconstruction error for the current assignment.
  3. `MyPCA.py`: a Python source file containing your implementation of the Principal Component Analysis algorithm with header `def PCA`. Given the original samples and an optional parameter *num_dim* as inputs, the function should return the low-dimensional projection and the dimension of features. If *num_dim* is specified, it should project the samples to d dimensional data, where d=*num_dim*. Otherwise, it will use the dimensions that capture $> 95\%$ of the variance.

- **Submit**: All material must be submitted electronically via Gradescope. **Note that there are two entries for the assignment, *i.e.*, Hw2-Written (for hw2_sol.pdf) and Hw2-Programming (for a zipped file containing the Python code), please submit your files accordingly.** We will grade the assignment with vanilla Python, and code submitted as iPython notebooks will not be graded.