CSE/ISyE 6740-A Homework 1

Name: TBD
GTID: TBD

Deadline: Sep 15th 2024 11:59pm ET

- There are 2 sections in grade scope: Homework 1 and Homework 1 Programming. Submit your answers as a PDF file to Homework 1 (including report for programming) and also submit your code in a zip file to Homework 1 Programming.
- Late homework incurs a penalty of 20% for each 24 hours that it is late. Thus, right after the deadline it will only be worth 80% credit and after five days it will not be worth any credit.
- We recommend the use of LaTeX for typing up your solutions. No credit will be given to unreadable handwriting.
- List explicitly with whom in the class you discussed which problem, if any. Cite all external ressources that you were using to complete the homeworks. For details, consult the collaboration policy in the class syllabus on canvas.
- Recommended reading: PRML[1] Section 9.1, 12.1

1 Foundations in Probability, Linear Algebra, and Matrix Calculus [20 pts]

(a) [5 pts] Probability
Consider two variables x and y with joint distribution $p(x, y)$. Prove the following two results

$$\mathbb{E}[x] = \mathbb{E}_y[\mathbb{E}_x[x|y]] \tag{1}$$
$$var[x] = \mathbb{E}_y var_x[x|y] + var_y[\mathbb{E}_x[x|y]] \tag{2}$$

(b) [5 pts] Pairwise Independence We say that two random variables are pairwise independent if

$$p(X_2|X_1) = p(X_2) \tag{3}$$

and hence

$$p(X_2, X_1) = p(X_1)p(X_2|X_1) = p(X_1)p(X_2) \tag{4}$$

We say that n random variables are mutually independent if

$$p(X_i|X_S) = p(X_i)\ \forall S \subseteq \{1, ..., n\} \setminus i \tag{5}$$

[1] Christopher M. Bishop, Pattern Recognition and Machine Learning, 2006, Springer.

and hence

$$p(X_{1:n}) = \prod_{i=1}^{n} p(X_i) \tag{6}$$

Show that pairwise independence between all pairs of variables does not necessarily imply mutual independence. It suffices to give a counter example.

(c) [5 pts] Linear Algebra

For $A \in \mathbb{R}^{m\times n}$, a pseudoinverse of A is defined as a matrix $A^+ \in \mathbb{R}^{n\times m}$ satisfying all of the following four criteria, known as the Moore–Penrose conditions:

a. AA^+ need not be the general identity matrix, but it maps all column vectors of A to themselves:

$$AA^+A = A.$$

b. A^+ acts like a weak inverse:

$$A^+AA^+ = A^+.$$

c. AA^+ is Hermitian:

$$(AA^+)^* = AA^+.$$

d. A^+A is also Hermitian:

$$(A^+A)^* = A^+A.$$

1. **Show that A^+A and AA^+ are idempotent operators.**

2. **When A has linearly independent columns, show that you can write A^+ as $(A^TA)^{-1}A^T$ and that this is a left inverse to A. Can you find a right inverse too and why?**

3. **Suppose you know the SVD of A, write A^+ using the SVD expression.**

(d) [5 pts] Matrix and Vector Calculus

Please answer the following questions about matrix calculus. If needed, please read "Linear Algebra Review and Reference" by Zico Kolter (`https://www.cs.cmu.edu/~zkolter/course/15-884/linalg-review.pdf`).

1. If $A(x) \in \mathbb{R}^{n\times n}$ is a function of $x \in \mathbb{R}$ and it is differentiable. Suppose that $A(x)$ is invertible at x. Please compute $\frac{d}{dx}\left(A^{-1}\right)$.

2. If $x \in \mathbb{R}^m$ and $f(x) \in \mathbb{R}^n$ differentiable at x, please compute $\frac{df(x)}{dx}$?

3. If $x \in \mathbb{R}^m$, $f : \mathbb{R}^m \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^l$, please compute $\frac{d\,g(f(x))}{dx}$ and explain why it works? Please specify the dimensionality of all the gradients.

2 Maximum Likelihood [15 pts]

Suppose we have n i.i.d (independent and identically distributed) data samples $D = \{x_1, x_2, ..., x_n\}$ from the following probability distribution. This problem asks you to build a log-likelihood function, and find the maximum likelihood estimator of the parameter(s).

(a) Exponential distribution [5 pts]

The probability density function of Exponential distribution is given by

$$f(x|\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

What is the maximum likelihood estimator of λ?

(b) Pareto distribution [5 pts]

The Pareto distribution has been used in economics for a density function with a slowly decaying tail:

$$f(x|x_0, \theta) = \theta x_0^{\theta} x^{-\theta-1}, \quad x \geq x_0, \theta > 0.$$

Assume that x_0 is given. What is the maximum likelihood estimator of θ?

(c) Poisson distribution [5 pts]

The Poisson distribution is defined as

$$P(x = k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad \forall k = \{0, 1, 2....\}.$$

What is the maximum likelihood estimator of λ?

3 Eigenvalues and Eigenvectors [15 pts]

We talk about how to find the largest eigenvalue/eigenvector by maximizing the variance in that direction. However, we have not covered how to find the remaining eigenvalues/eigenvectors.

Please read Section 2 of Spectral and Algebraic Graph Theory by Spielman :
`http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf`

- **(a) [10 pts]** Work on Exercise 2.4.
- **(a) [5 pts]** Can you design an algorithm to find eigenvalues/eigenvectors based on this?

4 Clustering [20 pts]

[a-b] Given N data points $\mathrm{x}^n (n = 1, \ldots, N)$, K-means clustering algorithm groups them into K clusters by minimizing the distortion function over $\{r^{nk}, \mu^k\}$

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r^{nk} \|\mathrm{x}^n - \mu^k\|^2,$$

where $r^{nk} = 1$ if x^n belongs to the k-th cluster and $r^{nk} = 0$ otherwise.

(a) [5 pts] Prove that using the squared Euclidean distance $\|\mathbf{x}^n - \mu^k\|^2$ as the dissimilarity function and minimizing the distortion function, we will have

$$\mu^k = \frac{\sum_n r^{nk} \mathbf{x}_n}{\sum_n r^{nk}}.$$

That is, μ^k is the center of k-th cluster.

(b) [5 pts] Prove that K-means algorithm converges to a local optimum in finite steps.

[c-d] In class, we discussed bottom-up hierarchical clustering. For each iteration, we need to find two clusters $\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_m\}$ and $\{\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_p\}$ with the minimum distance to merge. Some of the most commonly used distance metrics between two clusters are:

- Single linkage: the minimum distance between any pairs of points from the two clusters, i.e.

$$\min_{\substack{i=1,\ldots,m \\ j=1,\ldots,p}} \|\mathrm{x}_i - \mathrm{y}_j\|$$

- Complete linkage: the maximum distance between any parts of points from the two clusters, i.e.

$$\max_{\substack{i=1,\ldots,m \\ j=1,\ldots,p}} \|\mathrm{x}_i - \mathrm{y}_j\|$$

- Average linkage: the average distance between all pair of points from the two clusters, i.e.

$$\frac{1}{mp} \sum_{i=1}^{m} \sum_{j=1}^{p} \|\mathrm{x}_i - \mathrm{y}_j\|$$

(c) [5 pts] When we use the bottom up hierarchical clustering to realize the partition of data, which of the three cluster distance metrics described above would most likely result in clusters most similar to those given by K-means? (Suppose K is a power of 2 in this case).

(d) [5 pts] Is the time complexity $O(n^3)$ and space complexity $O(n^2)$ of the hierarchical clustering algorithm optimal (assuming single linkage)? Can you design an algorithm that has a more efficient time and space complexity? Does the same algorithm work for complete linkage and average linkage? Please also analyze your algorithm.

5 Programming: Image compression [Report + Code 30 pts]

In this programming assignment, you are going to apply clustering algorithms for image compression. Before starting this assignment, we strongly recommend reading PRML Section 9.1.1, page 428 – 430.

To ease your implementation, we provide a skeleton code containing image processing part. `homework1.ipynb` is designed to read an RGB bitmap image file, then cluster pixels with the given number of clusters K. It shows converted image only using K colors, each of them with the representative color of centroid. To see what it looks like, you are encouraged to run the notebook, for example.

Your task is implementing the clustering parts with two algorithms: *K-means* and *K-medoids*. We learned and demonstrated K-means in class, so you may start from the sample code we distributed.

The file you need to edit is Juptyer notebook, inside the notebook there are the function `mykmeans` and `mykmedoids`, provided with this homework. In the files, you can see it calls python function `kmeans` initially. Comment this line out, and implement your own in the files. You would expect to see similar result with your implementation of K-means, instead of `kmeans` function in python.

K-medoids

In class, we learned that the basic K-means works in Euclidean space for computing distance between data points as well as for updating centroids by arithmetic mean. Sometimes, however, the dataset may work better with other distance measures. It is sometimes even impossible to compute arithmetic mean if a feature

is categorical, e.g, gender or nationality of a person. With K-medoids, you choose a representative data point for each cluster instead of computing their average.

Given N data points $\mathrm{x}^n (n = 1, ..., N)$, K-medoids clustering algorithm groups them into K clusters by minimizing the distortion function $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r^{nk} D(\mathrm{x}^n, \mu^k)$, where $D(\mathrm{x}, \mathrm{y})$ is a distance measure between two vectors x and y in same size (in case of K-means, $D(x, y) = \|\mathrm{x} - \mathrm{y}\|^2$), μ^k is the center of k-th cluster; and $r^{nk} = 1$ if x^n belongs to the k-th cluster and $r^{nk} = 0$ otherwise. In this exercise, we will use the following iterative procedure:

- Initialize the cluster center μ^k, $k = 1, ..., K$.
- Iterate until convergence:
  - Update the cluster assignments for every data point x^n: $r^{nk} = 1$ if $k = \arg\min_j D(\mathrm{x}^n, \mu^j)$, and $r^{nk} = 0$ otherwise.
  - Update the center for each cluster k: choosing another representative if necessary.

There can be many options to implement the procedure; for example, you can try many distance measures in addition to Euclidean distance, and also you can be creative for deciding a better representative of each cluster. We will not restrict these choices in this assignment. You are encouraged to try many distance measures as well as way of choosing representatives.

Formatting instruction

Both function `mykmeans` and `mykmedoids` take input and output format as follows. You should not alter this definition, otherwise your submission will print an error, which leads to zero credit.

Input

- `pixels`: the input image representation. Each row contains one data point (pixel). For image dataset, it contains 3 columns, each column corresponding to Red, Green, and Blue component. Each component has an integer value between 0 and 255.
- `K`: the number of desired clusters. Too high value of K may result in empty cluster error. Then, you need to reduce it.

Output

- `class`: cluster assignment of each data point in pixels. The assignment should be 1, 2, 3, etc. For $K = 5$, for example, each cell of class should be either 1, 2, 3, 4, or 5. The output should be a column vector with `size(pixels, 1)` elements. Start from 0 if you are using python.
- `centroid`: location of K centroids (or representatives) in your result. With images, each centroid corresponds to the representative color of each cluster. The output should be a matrix with K rows and 3 columns. The range of values should be [0, 255], possibly floating point numbers.

Hand-in

Both of your code and report will be evaluated. Submit the notebook after completing the functions `mykmeans` and `mykmedoids` files as a zip to Homework 1 Programming. In your report, answer to the following questions:

1. Within the K-medoids framework, you have several choices for detailed implementation. Explain how you designed and implemented details of your K-medoids algorithm, including (but not limited to) how you chose representatives of each cluster, what distance measures you tried and chose one, or when you stopped iteration.
2. Attach a picture of your own. We recommend size of 320×240 or smaller.

3. Run your K-medoids implementation with the picture you chose above, with several different K. (e.g, small values like 2 or 3, large values like 16 or 32) What did you observe with different K? How long does it take to converge for each K?

4. Run your K-medoids implementation with different initial centroids/representatives. Does it affect final result? Do you see same or different result for each trial with different initial assignments? (We usually randomize initial location of centroids in general. To answer this question, an intentional poor assignment may be useful.)

5. Repeat question 3 and 4 with K-means. Do you see significant difference between K-medoids and K-means, in terms of output quality, robustness, or running time?

Note

- You may see some error message about empty clusters even with Matlab implementation, when you use too large K. Your implementation should treat this exception as well. That is, do not terminate even if you have an empty cluster, but use smaller number of clusters in that case.
- We will grade using test pictures which are not provided. We recommend you to test your code with several different pictures so that you can detect some problems that might happen occasionally.
- If we detect copy from any other student's code or from the web, you will not be eligible for any credit for the entire homework, not just for the programming part. Also, directly calling Matlab function `kmeans` or other clustering functions is not allowed.