Assignment 3 (Updated March 15 2022)

Deadline: Sunday, March 20, at 11:59pm.

Submission: You need to submit the final PDF file, and any Python scripts, as a compressed folder (.zip or .tgz) on Quercus. If you used Google Colab, including a link is sufficient.

Neatness Point: One point will be given for neatness. You will receive this point as long as we don't have a hard time reading your solutions or understanding the structure of your code.

Late Submission: 10% of the marks will be deducted for each day late, up to a maximum of 3 days. After that, no submissions will be accepted.

Assignments are individual work. See the Course Information handout[1] for detailed policies.

1. [**2pts**] **Revisiting single-cell RNA-seq**

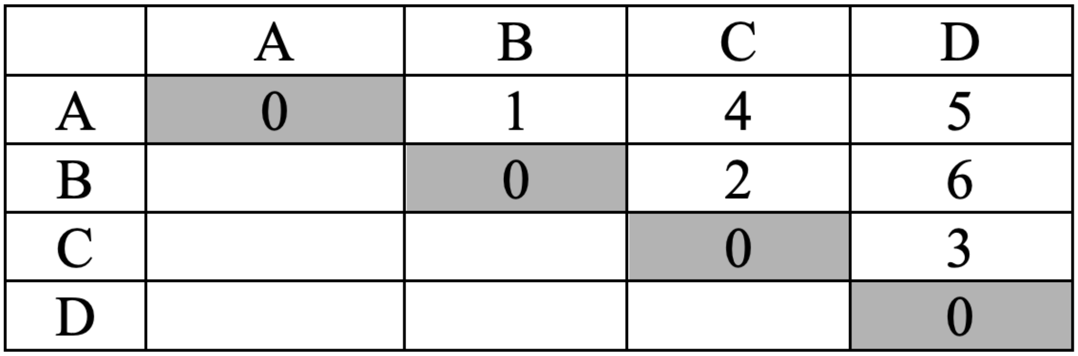
   In this question, we will re-use the single-cell RNA-seq data in A2. You can download the data from the course website.

   (a) [**1pts**] Run K-means and GMM with different K values [2,3,4,5] and report the AMI values [2].

   (b) [**1pts**] Comment on the differences between K-means and GMM in the above exercise in terms of speed and accuracy.

2. [**1 pts**] **Hierarchical Clustering**

   Use single and complete link agglomerative clustering to group the data described by the following distance matrix. SHOW EVERY STEP OF THE COMPUTATIONS AND THE FINAL DENDROGRAMS.



	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

3. [**1 pts**] **AdaBoost Theory** The goal of this question is to show that the AdaBoost algorithm changes the weights in order to force the weak learner to focus on difficult data points.

[1] https://lmp1210-uoft.github.io/2022/assets/misc/syllabus.pdf

[2] you can use the function in https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html

Here we consider the case that the target labels are from the set $\{-1, +1\}$ and the weak learner also returns a classifier whose outputs belongs to $\{-1, +1\}$ (instead of $\{0, 1\}$). Consider the t-th iteration of AdaBoost, where the weak learner is

$$h_t \leftarrow \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{N} w_i h(x^{(i)}) \neq t^{(i)},$$

the w-weighted classification error is

$$\text{err}_t = \frac{\sum_{i=1}^{N} w_i h_t(x^{(i)}) \neq t^{(i)}}{\sum_{i=1}^{N} w_i},$$

and the classifier coefficient is $\alpha_t = \frac{1}{2} \log \frac{1-\text{err}_t}{\text{err}_t}$. (Here, log denotes the natural logarithm.) AdaBoost changes the weights of each sample depending on whether the weak learner h_t classifies it correctly or incorrectly. The updated weights for sample i is denoted by w_i' and is

$$w_i' \leftarrow w_i \exp\Big(-\alpha_t t^{(i)} h_t(x^{(i)})\Big).$$

Show that the error w.r.t. $(w_1', \ldots, w_N')$ is exactly $\frac{1}{2}$. That is, show that

$$\text{err}_t' = \frac{\sum_{i=1}^{N} w_i' h_t(x^{(i)}) \neq t^{(i)}}{\sum_{i=1}^{N} w_i'} = \frac{1}{2}.$$

Note that here we use the weak learner of iteration t and evaluate it according to the new weights, which will be used to learn the $t+1$-st weak learner. What is the interpretation of this result?

Hints:

(a) Start from err_t' and divide the summation to two sets of $E = \{i : h_t(x^{(i)}) \neq t^{(i)}\}$ and its complement $E^c = \{i : h_t(x^{(i)}) = t^{(i)}\}$.

(b) Note that

$$\frac{\sum_{i \in E} w_i}{\sum_{i=1}^{N} w_i} = \text{err}_t.$$

4. **[2 pts] Adaboost Implementation**

   Please complete this Colab notebook.

5. **[4 pts] PCA**

   For this question we will also use the RNA-seq dataset from Q1. You do not need to split the data.

   (a) **[1 pt]** Run Principal Component Analysis (PCA) on your data with 10 components, and report their variance explained. Which component is most important for representing the data?

   (b) **[1 pt]** Make a scatter plot the largest principal component (y-axis) vs the second largest (x-axis). Colour the points based on their class label. Do you notice a pattern? Repeat this process with the bottom two principal components.

   Note: see the Lecture 7 Colab book for hints

(c) **[1 pt]** Train a logistic regression classifier on the top two principal components, and add this line to your plot. Do the same for the bottom two principal components. Comment on the accuracy of each model.

Note: this tutorial may be of use

(d) **[1 pt]** For k from 1 to 10, train a logistic regression model that uses the top k principal components (i.e. the first model uses the top 1, the second uses the top 2, and so on). Make a line plot of each model's accuracy, where the x-axis is the k value and the y-axis is the accuracy of the model that uses the top k principal components. What pattern do you observe?

6. **[4 pts] Multi-Omics Analysis for Cancer Subtyping** You are given a simulated multi-omic dataset for 832 cancer patients with 10 subtypes. The dataset consists of bulk RNA-seq (see file "A3RNAseq.csv") and DNA Methylations (see file "A3Methylation.csv"). The groundtruth can be found in file "label.csv".

(a) **[1 pts]** Use three different visualization methods (PCA, tSNE, and UMAP) to project the RNA-seq data into a 2D space in which each dot represents a patient color-coded with the cancer subtypes. Further, run k-means with $k = 10$ and report the AMI value.

(b) **[1 pts]** Use three different visualization methods (PCA, tSNE, and UMAP) to project the Methylation data into a 2D space in which each dot represents a patient color-coded with the cancer subtypes. Further, run k-means with $k = 10$ and report the AMI value.

(c) **[2 pts]** Develop a unsupervised multi-modal machine learning algorithm which takes both RNA-seq and Methylation data and learn a latent representation of patients. Once the patient representations are learned, use UMAP to visualize them in 2-D space in which each dot represents a patient color-coded with the cancer subtypes. Further, run k-means with $k = 10$ and report the AMI value. The higher your final AMI is, the more marks you will get. **Note: this is an open-ended question. You can use any external package you can possibly find to solve this question. Please submit your code too with a Colab link so that we can verify the result.**