

## HOMEWORK 4 - V0

STA414/2104 WINTER 2021

*University of Toronto*

VERSION HISTORY: V0  $\rightarrow$  V1:

- **Deadline:** Apr. 5, at 13:59.
- **Submission:** You need to submit your solutions through Crowdmark, including all your derivations, plots, and your code. You can produce the file however you like (e.g. L<sup>A</sup>T<sub>E</sub>X, Microsoft Word, etc), as long as it is readable. Points will be deducted if we have a hard time reading your solutions or understanding the structure of your code. For this assignment, you should submit the filled out starter code right after your answers.

**1. Unsupervised Learning, 30 pts.** In this problem, we will experiment with two clustering algorithms: (i) k-means, and (ii) EM algorithm for Gaussian mixtures. In what follows,  $N$  denotes the number of data points,  $D$  denotes the dimension of each data point,  $\mathbf{x}_i \in \mathbb{R}^D$  denotes a single data point,  $K$  denotes the number of clusters, and  $\mathbf{x}_i \sim p(\mathbf{x}|z = k)$  for  $k = 1, \dots, K$  denotes the data generating distribution, and  $p(z = k)$  is the prior on the class (cluster in this case).

- (a) First, we will generate some data for this problem. Set the number of points  $N = 400$ , their dimension  $D = 2$ , and the number of clusters  $K = 2$ , and generate data from the distribution  $p(\mathbf{x}|z = k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ . Sample 200 data points for  $k = 1$  and 200 for  $k = 2$ , with

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} 6.0 \\ 0.1 \end{bmatrix} \text{ and } \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix} 10 & 7 \\ 7 & 10 \end{bmatrix}$$

Here,  $N = 400$ . Since you generated the data, you already know which sample comes from which class. Run the cell in the IPython notebook to generate the data.

Make a scatter plot of the data points showing the true cluster assignment of each point either using different color codes or shape or both.

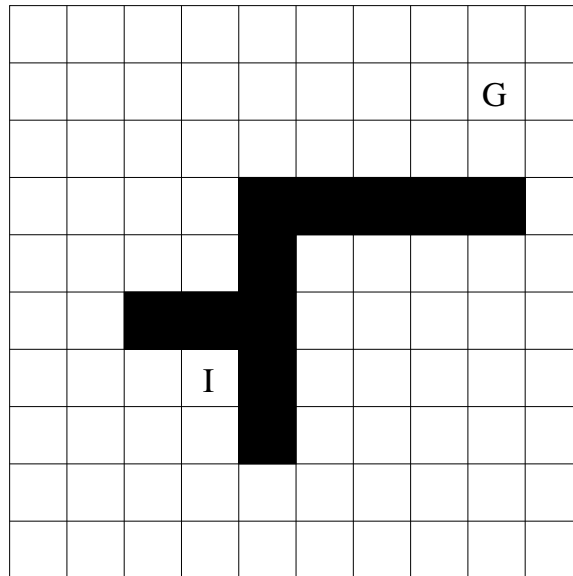
- (b) Now, we assume that the true class labels are not known. Implement the k-means algorithm for this problem. Write **two** functions: `km_assignment_step`, and `km_refitting_step` as given in the lecture (Here, `km_` means k-means). Identify the correct arguments, and the order to run them. Initialize the algorithm with

$$(1.1) \quad \hat{\boldsymbol{\mu}}_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}, \hat{\boldsymbol{\mu}}_2 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}$$

and run it until convergence. Show the resulting cluster assignments on a scatter plot either using different color codes or shape or both. Also plot the cost vs. the number of iterations. Report your misclassification error. Hint: you generated the data, you know the correct labels. Your unsupervised learning algorithm doesn't.

- (c) Next, implement the EM algorithm for Gaussian mixtures. Write **three** functions: `log-likelihood`, `gm_e_step`, and `gm_m_step` as given in the lecture. Identify the correct arguments, and the order to run them. Initialize the algorithm with means as in (1.1), covariances with  $\hat{\Sigma}_1 = \hat{\Sigma}_2 = I$ , and  $\hat{\pi}_1 = \hat{\pi}_2$ . Run the algorithm until convergence and show the resulting cluster assignments on a scatter plot either using different color codes or shape or both. Also plot the log-likelihood vs. the number of iterations. Report your misclassification error.
- (d) Comment on the results:
- Compare the performance of k-Means and EM based on the resulting cluster assignments.
  - Compare the performance of k-Means and EM based on their convergence rate. What is the bottleneck for which method?
  - Experiment with 5 different data realizations (generate new data), run your algorithms, and summarize your findings. Does the algorithm performance depend on different realizations of data?

**2. Reinforcement Learning, 65 pts.** In this portion of the assignment, you will write software to implement a simulated environment and build a reinforcement learning agent that discovers the optimal (shortest) path to a goal. The agent’s environment will look like:



Each cell is a state in the environment. The cell labeled “I” is the initial state of the agent. The cell labeled “G” is the goal state of the agent. The black cells are barriers—states that are inaccessible to the agent. At each time step, the agent may move one cell to the left, right, up, or down. The environment does not wraparound. Thus, if the agent is in the lower left corner and tries to move down, it will remain in the same cell. Likewise, if the agent is in the initial state and tries to move up (into a barrier), it will remain in the same cell.