CS 446 / ECE 449 — Homework 4

your NetID here

Version 1.0

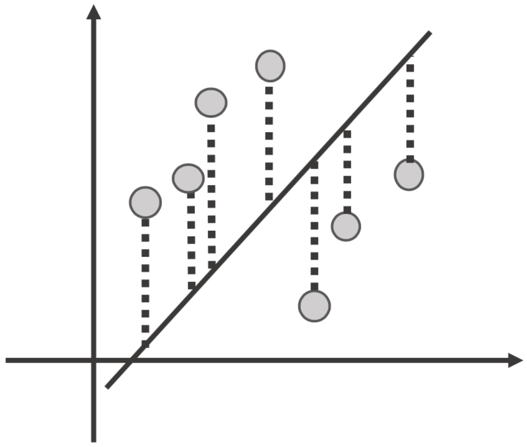
Instructions.

- Homework is due **Tuesday, April 6th, at noon CST**; no late homework accepted.
- Everyone must submit individually at gradescope under `hw4` and `hw4code`.
- The "written" submission at `hw4` **must be typed**, and submitted in any format gradescope accepts (to be safe, submit a PDF). You may use LaTeX, markdown, google docs, MS word, whatever you like; but it must be typed!
- When submitting at `hw4`, gradescope will ask you to mark out boxes around each of your answers; please do this precisely!
- Please make sure your NetID is clear and large on the first page of the homework.
- Your solution **must** be written in your own words. Please see the course webpage for full academic integrity information. Briefly, you may have high-level discussions with at most 3 classmates, whose NetIDs you should place on the first page of your solutions, and you should cite any external reference you use; despite all this, your solution must be written in your own words.
- We reserve the right to reduce the auto-graded score for `hw4code` if we detect funny business (e.g., your solution lacks any algorithm and hard-codes answers you obtained from someone else, or simply via trial-and-error with the autograder).
- When submitting to `hw4code`, only upload `hw4.py` and `hw4_utils.py`. Additional files will be ignored.

1. Principal Component Analysis

(a) For each of the following statements, specify whether the statement is true or false. If you think the statement is wrong, explain in 1 to 2 sentences why it is wrong.

- True or False: As shown in the figure below, PCA seeks a subspace such that the sum of all the vertical distance to the subspace (the dashed line) is minimized.



- True or False: PCA seeks a projection that best represents the data in a least-squares sense.
- True or False: PCA seeks a linear combination of variables such that the maximum variance is extracted from the variables.
- True or False: The principal components are not necessarily orthogonal to each other.

- True or False: Solving PCA using SVD might result in a solution which corresponds to a local minimum.

(b) Recall that PCA finds a direction w in which the projected data has highest variance by solving the following program:

$$\max_{w:||w||^2=1} w^T \Sigma w. \tag{1}$$

Here, Σ is a covariance matrix. You are given a dataset of two 2-dimensional points $(1,3)$ and $(4,7)$. Draw the two data points on the 2D plane. What is the first principal component w of this dataset?

(c) Now you are given a dataset of four points $(2,0)$, $(2,2)$, $(6,0)$ and $(6,2)$. Draw the four data points on the 2D plane. Given this dataset, what is the dimension of the covariance matrix Σ in Eq. (1)? Also, explicitly write down the values of Σ given the dataset.

(d) What is the optimal w and the optimal value of the program in Eq. (1) given

$$\Sigma = \begin{bmatrix} 12 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

Solution.

(a)
- False. Should be the orthogonal projection not vertical.
- True.
- True.

- False. The principal components are always orthogonal to each other.
- False. It finds a global optimum.

(b)

$$\frac{1}{5}[3, 4]^T$$

(c)

$$X = \begin{bmatrix} 2 & 2 & 6 & 6 \\ 0 & 2 & 0 & 2 \end{bmatrix}$$

First, center the data.

$$\bar{X} = \begin{bmatrix} -2 & -2 & 2 & 2 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

The covariance matrix Σ is a 2-by-2 matrix.

$$\Sigma = \frac{1}{4}\bar{X}\bar{X}^T = \frac{1}{4}\begin{bmatrix} 16 & 0 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

(d) By inspection, the eigenvector corresponding to the largest eigenvalue is:

$$[0, 0, 1, 0]^T,$$

and the largest eigenvalue is 20. Therefore, $w^* = [0, 0, 1, 0]^T$ and the optimal value is 20.

2. Gaussian Mixture Models

Consider a Gaussian mixture model with K components ($k \in \{1, \dots, K\}$), each having mean μ_k, variance σ_k^2, and mixture weight π_k. Further, we are given a dataset $\mathcal{D} = \{x_i\}$, where $x_i \in \mathbb{R}$. We use $z_i = \{z_{ik}\}$ to denote the latent variables.

(a) What is the log-likelihood of the data according to the Gaussian Mixture Model? (use μ_k, σ_k, π_k, K, x_i, and $\mathcal{D}$).

(b) Assume $K = 1$, find the maximum likelihood estimate for the parameters (μ_1, σ_1^2, π_1).

(c) What is the probability distribution on the latent variables, i.e., what is the distribution $p(z_{i,1}, z_{i,2}, \cdots, z_{i,K})$ underlying Gaussian mixture models. Also give its name.

(d) For general K, what is the posterior probability $p(z_{ik} = 1|x_i)$? To simplify, wherever possible, use $\mathcal{N}(x_i|\mu_k, \sigma_k)$, a Gaussian distribution over $x_i \in \mathbb{R}$ having mean μ_k and variance σ_k^2.

(e) How are k-Means and Gaussian Mixture Model related? (There are three conditions)

(f) Consider the modified Gaussian Mixture Model objective:

$$\min_{\mu} - \sum_{x_i \in \mathcal{D}} \epsilon \log \sum_{k=1}^{K} \exp\left(-(x_i - \mu_k)^2/\epsilon\right).$$

We denote by $F_k = (x - \mu_k)^2$. Show that:

$$\lim_{\epsilon \to 0} -\epsilon \log \sum_{k=1}^{K} \exp\left(-F_k/\epsilon\right) = \min_k F_k, \quad \epsilon \in \mathbb{R}^+$$

Hint: Use l'Hopital rule.

(g) Conclude that the objective for k-Means is the 0-temperature limit of Gaussian Mixture Model under the conditions you provided in part (e).

Solution.

(a)

$$\sum_{i \in \mathcal{D}} \log \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\Big(- \frac{(x_i - \mu_k)^2}{2\sigma_k^2} \Big)$$

(b)

$$\mu_1 = \frac{1}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}} x_i$$

$$\sigma_1^2 = \frac{1}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}} (x_i - \mu_1)^2$$

$$\pi_1 = 1$$

(c) Multinomial/categorical distribution;

$$p(z_{i,1}, z_{i,2}, \cdots, z_{i,K}) = \prod_{k=1}^{K} \pi_k^{z_{ik}}$$

(d)

$$p(z_{ik}=1|x_i)=\frac{\pi_k\mathcal{N}(x_i|\mu_k,\sigma_k)}{\sum_{\hat{k}=1}^{K}\pi_{\hat{k}}\mathcal{N}(x_i|\mu_{\hat{k}},\sigma_{\hat{k}})}$$

(e) 1) Same variance for all Gaussian mixtures.
2) $\pi_k > 0$, $\forall k$ (or $\pi_k = 1/K$, $\forall k$ is also correct)
3) Zero-temperature (hard-version) of the Gaussian Mixture Model. ($\epsilon \to 0$)

(f)

$$\min_{\mu} -\sum_{x_i\in\mathcal{D}}\epsilon\log\sum_{k=1}^{K}\exp -(x_i-\mu_k)^2/\epsilon$$

Apply L'Hospital's rule, let $F_k=(x-\mu_k)^2$

$$\lim_{\epsilon\to 0}\frac{-\log\sum_{k=1}^{K}\exp\left(\frac{-F_k}{\epsilon}\right)}{\frac{1}{\epsilon}}=\lim_{\epsilon\to 0}\frac{\partial_\epsilon(-\log\sum_{k=1}^{K}\exp\left(\frac{-F_k}{\epsilon}\right))}{\partial_\epsilon(\frac{1}{\epsilon})}=\lim_{\epsilon\to 0}\sum_{k=1}^{K}\frac{\exp\left(-F_k/\epsilon\right)}{\sum_{\hat{k}}\exp\left(-F_{\hat{k}}/\epsilon\right)}\cdot F_k$$

After moving the limit inside, we obtain

$$\lim_{\epsilon\to 0}\frac{-\log\sum_{k=1}^{K}\exp\left(\frac{-F_k}{\epsilon}\right)}{\frac{1}{\epsilon}}=\min_k F_k$$

(g) The GMM loss is of the form: $-\sum_{x_i\in\mathcal{D}}\epsilon\log\sum_{k=1}^{K}\exp\left(-(x_i-\mu_k)^2/\epsilon\right)$.
Computing the limit of the loss when $\epsilon\to 0$ results in:

$$\lim_{\epsilon\to 0}-\sum_{x_i\in\mathcal{D}}\epsilon\log\sum_{k=1}^{K}\exp\left(-(x_i-\mu_k)^2/\epsilon\right)=-\sum_{x_i\in\mathcal{D}}\lim_{\epsilon\to 0}\epsilon\log\sum_{k=1}^{K}\exp\left(-(x_i-\mu_k)^2/\epsilon\right).$$

We proved in part (f) that:

$$\lim_{\epsilon\to 0}-\epsilon\log\sum_{k=1}^{K}\exp\left(-(x-\mu_k)^2/\epsilon\right)=\sum_{k=1}^{K}\mathbb{1}_{\{k=\underset{k}{\operatorname{argmin}}(x-\mu_k)^2\}}(x-\mu_k)^2,\quad \epsilon\in\mathbb{R}^+$$

Combining the last two results above results in the k-means objective:

$$\lim_{\epsilon\to 0}-\sum_{x_i\in\mathcal{D}}\epsilon\log\sum_{k=1}^{K}\exp\left(-(x_i-\mu_k)^2/\epsilon\right)=\sum_{x_i\in\mathcal{D}}\sum_{k=1}^{K}\mathbb{1}_{[k=\operatorname{argmin}_k(x-\mu_k)^2]}(x_i-\mu_k)^2$$

3. Expectation Maximization

In this problem, you will implement an expectation-maximization (EM) algorithm to cluster samples $\mathcal{D} = \{x^{(i)}\}_{i=1}^n$, with $x^{(i)} \in \{0,1\}^D$ into groups. You will be using a mixture of Bernoullis model to tackle this problem.

(a) **Mixture of Bernoullis**.

i. Assume each variable x_d is drawn from a Bernoulli(q_d) distribution, $P(x_d = 1) = q_d$. Let $q = [q_1, \cdots, q_D] \in [0,1]^D$ be the resulting vector of Bernoulli parameters. Write an expression for $P(x|q)$ as a function of q_d and x_d.

ii. Now suppose we have a mixture of K Bernoulli distributions: each vector $x^{(i)}$ is drawn from some vector of Bernoulli random variables with parameters $p^{(k)} = [p_1^{(k)}, \cdots, p_D^{(k)}]$, that we call Bernoulli($p^{(k)}$). Let $\{p^{(1)}, \cdots, p^{(K)}\} = p$. Assume a distribution π over the selection of which set of Bernoulli parameters $p^{(k)}$ is chosen. Write an expression for $P(x^{(i)}|p, \pi)$, as a function of π_k and $P(x^{(i)}|p^{(k)})$. Here π_k denotes the probability associated with the k^{th} Bernoulli component.

iii. Using the above, write an expression for the log-likelihood of the data $\mathcal{D}$, $\log P(\mathcal{D}|\pi, p)$.

(b) **Expectation step**.

i. Let $z^{(i)} \in \{0,1\}^K$ be an indicator vector, such that $z_k^{(i)} = 1$ if $x^{(i)}$ was drawn from a Bernoulli($p^{(k)}$), and 0 otherwise. Let $Z = \{z^{(i)}\}_{i=1}^n$. Write down the expression of $P(z^{(i)}|\pi)$ as a function of π_k and $z_k^{(i)}$.

ii. Write down the expression of $P(x^{(i)}|z^{(i)}, p, \pi)$ as a function of $P(x^{(i)}|p^{(k)})$ and $z_k^{(i)}$.

iii. Using the two quantities above, derive the likelihood of the data and latent variables $P(Z, \mathcal{D}|\pi, p)$.

iv. Let $\eta(z_k^{(i)}) = \mathbb{E}[z_k^{(i)}|x^{(i)}, \pi, p]$. Show that

$$\eta(z_k^{(i)}) = \frac{\pi_k \prod_{d=1}^D (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^k)^{1-x_d^{(i)}}}{\sum_j \pi_j \prod_{d=1}^D (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^j)^{1-x_d^{(i)}}}$$

v. Let $\tilde{p}$, $\tilde{\pi}$ be the new parameters that we would like to maximize. p, π are from the previous iteration. Use this to derive the following final expression for the E-step in the EM algorithm:

$$\mathbb{E}[\log P(Z, \mathcal{D}|\tilde{p}, \tilde{\pi})|\mathcal{D}, p, \pi] = \sum_{i=1}^n \sum_{k=1}^K \eta(z_k^{(i)}) \Big[\log \tilde{\pi}_k + \sum_{d=1}^D (x_d^{(i)} \log \tilde{p}_d^{(k)} + (1 - x_d^{(i)}) \log(1 - \tilde{p}_d^{(k)})) \Big]$$

(c) **Maximization step**. In the following, we will find $\tilde{p}$ and $\tilde{\pi}$ that maximize the above expression.

i. Show that $\tilde{p}$ that maximizes the E-step is:

$$\tilde{p}^{(k)} = \frac{\sum_{i=1}^n \eta(z_k^{(i)}) x^{(i)}}{N_k},$$

where $N_k = \sum_{i=1}^n \eta(z_k^{(i)})$.

ii. Prove that the value of $\tilde{\pi}$ that maximizes the E-step is:

$$\tilde{\pi}_k = \frac{N_k}{\sum_{k'} N_{k'}}.$$

Solution.

(a) i. $P(x|q) = \prod_{d=1}^D q_d^{x_d} (1 - q_d)^{1-x_d}$

ii. Let $A_k^{(i)}$ be the event that $x^{(i)}$ was drawn from $p^{(k)}$. Then:

$$P(x^{(i)}|p,\pi) = \sum_k P(x^{(i)}, A_k^{(i)}|p,\pi) = \sum_k P(x^{(i)}|A_k^{(i)}, p, \pi) P(A_k^{(i)}|p,\pi) = \sum_k \pi_k P(x^{(i)}|p^{(k)})$$

iii.

$$\log P(\mathcal{D}|\pi, p) = \sum_{i=1}^n \log P(x^{(i)}|p,\pi) = \sum_{i=1}^n \log \sum_k (\pi_k P(x^{(i)}|p^{(k)}))$$

(b) i. $P(z^{(i)}|\pi) = \prod_{k=1}^K \pi_k^{z_k^{(i)}}$

ii. $P(x^{(i)}|z^{(i)}, p, \pi) = \prod_{k=1}^K [P(x^{(i)}|p^{(k)})]^{z_k^{(i)}}$

iii.

$$P(Z, \mathcal{D}|\pi, p) = \prod_{i=1}^n P(x^{(i)}, z^{(i)}|\pi, p) \tag{2}$$

$$= \prod_{i=1}^n P(x^{(i)}|z^{(i)}, \pi, p) P(z^{(i)}|\pi) \tag{3}$$

$$= \prod_{i=1}^n \Big[\prod_{k=1}^K P(x^{(i)}|p^{(k)})^{z_k^{(i)}}\Big]\Big[\prod_{k=1}^K \pi_k^{z_k^{(i)}}\Big]. \tag{4}$$

iv.

$$\eta(z_k^{(i)}) = \mathbb{E}[z_k^{(i)}|x^{(i)}, \pi, p] \tag{5}$$

$$= P[z_k^{(i)} = 1|x^{(i)}, \pi, p] \tag{6}$$

$$= \frac{P(x^{(i)}|z_k^{(i)} = 1, \pi, p) P(z_k^{(i)} = 1|\pi, p)}{\sum_{k'} P(x^{(i)}|z_{k'}^{(i)} = 1, \pi, p) P(z_{k'}^{(i)} = 1|\pi, p)} \tag{7}$$

$$= \frac{\pi_k \prod_{d=1}^D (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1 - x_d^{(i)}}}{\sum_{k'} \pi_{k'} \prod_{d=1}^D (p_d^{(k')})^{x_d^{(i)}} (1 - p_d^{(k')})^{1 - x_d^{(i)}}} \tag{8}$$

v.

$$\log P(Z, \mathcal{D}|\tilde{\pi}, \tilde{p}) = \sum_{i=1}^n \Big[\sum_{k=1}^K z_k^{(i)} \log[P(x^{(i)}|\tilde{p}^{(k)})]\Big] + \Big[\sum_{k=1}^K z_k^{(i)} \log \tilde{\pi}_k\Big] \tag{9}$$

$$= \sum_{i=1}^n \sum_{k=1}^K z_k^{(i)} \Big[\log P(x^{(i)}|\tilde{p}^{(k)}) + \log \tilde{\pi}_k\Big] \tag{10}$$

$$= \sum_{i=1}^n \sum_{k=1}^K z_k^{(i)} \Big[\log \tilde{\pi}_k + \log \prod_{d=1}^D (\tilde{p}_d^{(k)})^{x_d^{(i)}} (1 - \tilde{p}_d^{(k)})^{(1 - x_d^{(i)})}\Big] \tag{11}$$

$$= \sum_{i=1}^n \sum_{k=1}^K z_k^{(i)} \Big[\log \tilde{\pi}_k + \sum_{d=1}^D (x_d^{(i)} \log \tilde{p}_d^{(k)} + (1 - x_d^{(i)}) \log(1 - \tilde{p}_d^{(k)}))\Big] \tag{12}$$

Taking the expectation and using $\mathbb{E}[z_k^{(i)}] = \eta(z_k^{(i)})$, we have

$$= \sum_{i=1}^n \sum_{k=1}^K \eta(z_k^{(i)}) \Big[\log \tilde{\pi}_k + \sum_{d=1}^D (x_d^{(i)} \log \tilde{p}_d^{(k)} + (1 - x_d^{(i)}) \log(1 - \tilde{p}_d^{(k)}))\Big]$$

(c) i. Computing the derivative and setting it to 0 results in:

$$\frac{d}{dp_d^{(k)}}\mathbb{E}[\log P(Z,\mathcal{D}|\tilde{\pi},\tilde{p})] = \sum_{i=1}^{n}\eta(z_k^{(i)})\Big[\frac{x_d^{(i)}}{\tilde{p}_d^{(k)}} - \frac{1-x_d^{(i)}}{1-\tilde{p}_d^{(k)}}\Big] = 0$$

Solving for $\tilde{p}_d^{(k)}$ results in

$$\tilde{p}_d^{(k)} = \frac{\sum_{i=1}^{n}\eta(z_k^{(i)})x_d^{(i)}}{\sum_{i=1}^{n}\eta(z_k^{(i)})} = \frac{\sum_{i=1}^{n}\eta(z_k^{(i)})x^{(i)}}{N_k}.$$

ii. We only need to minimize $\sum_{i=1}^{n}\sum_{k=1}^{K}\eta(z_k^{(i)})\log\tilde{\pi}_k$ since the rest is not a function of π. In order to keep $\tilde{\pi}$ a distribution, we require $\sum_k \tilde{\pi}_k = 1$. Hence, the Lagrangian is:

$$L(\tilde{\pi},\lambda) = -\sum_{i=1}^{n}\sum_{k=1}^{K}\eta(z_k^{(i)})\log\tilde{\pi}_k + \lambda\Big(\sum_{k=1}^{K}\tilde{\pi}_k - 1\Big).$$

Taking the derivative with respect to $\tilde{\pi}_k$ results in:

$$\frac{d}{d\tilde{\pi}_k}L(\pi,\lambda) = -\sum_{i=1}^{n}\frac{\eta(z_k^{(i)})}{\tilde{\pi}_k} + \lambda = 0.$$

Solving for π_k, results in:

$$\tilde{\pi}_k = \frac{\sum_{i=1}^{n}\eta(z_k^{(i)})}{\lambda} = \frac{N_k}{\lambda}$$

Now, solve for λ:

$$L(\lambda) = -\sum_{i=1}^{n}\sum_{k=1}^{K}\eta(z_k^{(i)})(\log N_k - \log\lambda) + (\sum_{k=1}^{K}N_k - \lambda)$$

Computing the derivative with respect to λ, results in:

$$\frac{1}{\lambda}\sum_{i=1}^{n}\sum_{k=1}^{K}\eta(z_k^{(i)}) - 1 = 0.$$

Solving for λ results in:

$$\lambda = \sum_{i=1}^{n}\sum_{k=1}^{K}\eta(z_k^{(i)}) = \sum_{k=1}^{K}N_k$$

4. K-Means 1

(a) Mention if K-Means is a supervised or an un-supervised method and state the reason.

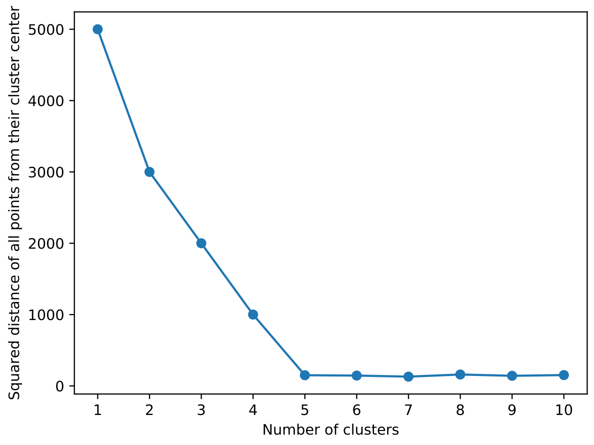
(b) Assume that you are trying to cluster data points x_i for $i \in \{1, 2, \ldots, D\}$ into K clusters each with center μ_k where $k \in \{1, 2, \ldots, K\}$. The objective function for doing this clustering involves minimize the euclidean distance between the points and the cluster centers. It is given by

$$\min_{\mu} \min_{r} \sum_{i \in D} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x_i - \mu_k\|_2^2.$$

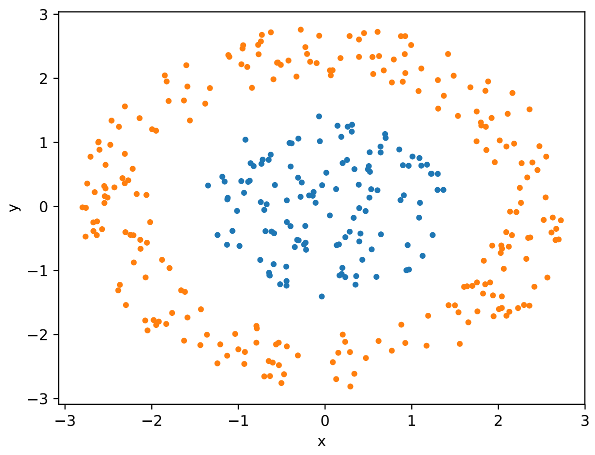
How do you ensure hard assignment of one data point to one and only one cluster at a given time?
Hint: By hard assignment we mean that you are 100 % sure that a point either belongs or doesn't belong to a cluster.

(c) How does your answer to part b change if we want to obtain a soft assignment instead?
Hint: By soft assignment we mean that a point belongs to a cluster with some probability.

(d) Looking at the following plot, what is the best choice for the number of clusters?



(e) Would K-Means be an efficient algorithm to cluster the following data? Explain your answer in a couple of lines.



Solution.

(a) Unsupervised method. Because we do not have labels.

(b) $r_{ik} \in \{0, 1\}, \sum_{k=1}^{K} r_{ik} = 1$

(c) $r_{ik} \in [0, 1], \sum_{k=1}^{K} r_{ik} = 1$

(d) 5. Because the number 5 is at the elbow and the larger number contribute less to the result.

(e) The original version of K-means is not a good method for clustering this data because this data is not linearly-separable. We need a non-linear kernel to transform data before implementing K-means.

5. K-Means 2

We are given a dataset $\mathcal{D} = \{(x)\}$ of 2d points $x \in \mathbb{R}^2$ which we are interested in partitioning into K clusters, each having a cluster center μ_k ($k \in \{1, \dots, K\}$) via the k-Means algorithm. This algorithm optimizes the following cost function:

$$\min_{\mu_k, r} \sum_{x \in \mathcal{D}, k \in \{1,\dots,K\}} \frac{1}{2} r_{x,k} \|x - \mu_k\|_2^2 \qquad \text{s.t.} \quad \begin{cases} r_{x,k} \in \{0,1\} & \forall x \in \mathcal{D}, k \in \{1,\dots,K\} \\ \sum_{k \in \{1,\dots,K\}} r_{x,k} = 1 & \forall x \in \mathcal{D} \end{cases} \tag{13}$$

(a) What is the domain for μ_k?

(b) Given fixed cluster centers μ_k $\forall k \in \{1, \dots, K\}$, what is the optimal $r_{x,k}$ for the program in Eq. 13? Provide a reason?

(c) Given fixed $r_{x,k}$ $\forall x \in \mathcal{D}, k \in \{1, \dots, K\}$, what are the optimal cluster centers μ_k $\forall k \in \{1, \dots, K\}$ for the program in Eq. 13?

Hint: Reason by first computing the derivative w.r.t μ_k.

(d) Using Pseudo-code, sketch the algorithm which alternates the aforementioned two steps. Is this algorithm guaranteed to converge and why? Is this algorithm guaranteed to find the global optimum? What is the reason?

Hint: you can provide a counter-example to invalidate a statement.

(e) Please implement the aforementioned two steps. For the given dataset, after how many updates does the algorithm converge, what cost function value does it converge to and what are the obtained cluster centers? Visualize clusters at each step and attach the plots here. Please at least report numbers with one decimal point.

Remark: how we count updates: when computing a set of new centroids from initilization, we call this one update.

Hint: You may find `hw4_utils.vis_cluster` useful.

Solution.

(a) $\mu_k \in \mathbb{R}^2$

(b) Assign data x to the closest cluster center μ_k, where closeness is measured using the 2-norm.

$$r_{x,k} = \begin{cases} 1 & \text{if } \ k = \arg\min_{k \in \{1,\dots,K\}} \|x - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

(c) Derivative:

$$\sum_{x \in \mathcal{D}} r_{x,k}(x - \mu_k) = 0$$

Solution:

$$\mu_k = \frac{\sum_{x \in \mathcal{D}} r_{x,k} x}{\sum_{x \in \mathcal{D}} r_{x,k}}$$

(d) Initialize μ_k and iterate:

```
i. $r_{x,k} = \begin{cases} 1 & \text{if } \ k = \arg\min_{k \in \{1,\dots,K\}} \|x - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$
ii. $\mu_k = \frac{\sum_{x \in \mathcal{D}} r_{x,k} x}{\sum_{x \in \mathcal{D}} r_{x,k}}$
```

Guaranteed to converge because we find the global minimum for each step which always decreases the cost function value. Meanwhile, since there are at most k^N ways to partition N points into k clusters, the domain is finite.

Not guaranteed to find the global optimum. Consider the following initialization for four points (width has length $2l$ and height has l):

$$\cdot \quad c \quad \cdot \tag{14}$$
$$\tag{15}$$
$$\cdot \quad c \quad \cdot \tag{16}$$
$$\tag{17}$$

where $\cdot$ is a point and c is the center. With such an initialization, the K-means will directly stop. However, the optimal solution is

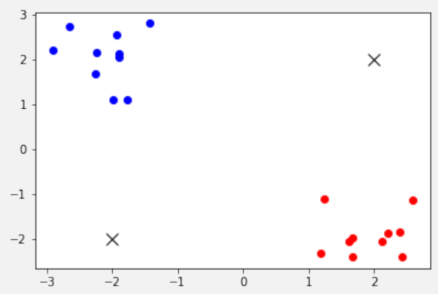
$$\cdot \qquad \cdot \tag{18}$$
$$c \qquad c \tag{19}$$
$$\cdot \qquad \cdot \tag{20}$$
$$\tag{21}$$

(e) Converges after 2 updates

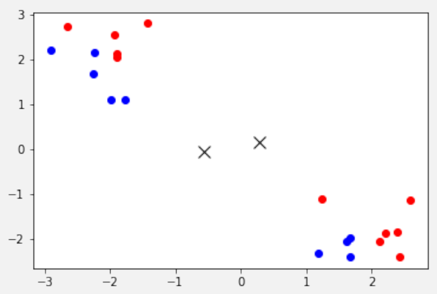
Converges to a cost function value of 4.56

Cluster centers:

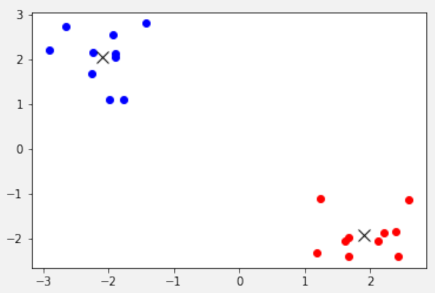
$$\mu_1 = \begin{bmatrix} 1.9 \\ -1.9 \end{bmatrix} \quad \text{and} \quad \mu_2 = \begin{bmatrix} -2.1 \\ 2.1 \end{bmatrix}$$



(a) Step 0.



(b) Step 1.



(c) Step 2.

Figure 1: Visualization.

```
def k_means(X=None, init_c=None, n_iters=50):
    """K-Means.

    Argument:
        X: 2D data points, shape [2, N].
        init_c: initial centroids, shape [2, 2]. Each column is a centroid.

    Return:
        c: shape [2, 2]. Each column is a centroid.
    """

    if X is None:
        X, init_c = hw4_utils.load_data()

    # [2, 2, 1]. 1st 2 is for #c; 2nd 2 is for (x, y).
    c = init_c.transpose(0, 1).unsqueeze(-1)

    for k in range(n_iters):
        # [2, N]
        dist = 0.5 * torch.sum(torch.pow(X - c, 2), 1)

        val, assign = dist.min(0)

        print("Cost: %f" % torch.sum(val))

        c1 = torch.mean(X[:, assign == 0], 1).unsqueeze(-1)
        c[0, :, :] = c1

        c2 = torch.mean(X[:, assign == 1], 1).unsqueeze(-1)
        c[1, :, :] = c2

        hw4_utils.vis_cluster(c1, X[:, assign == 0], c2, X[:, assign == 1])

    return c[..., 0].transpose(0, 1)
```