# CS 446 / ECE 449 — Homework 4

*mukaiyu2*

Version 1.0

**Instructions.**

- Homework is due **Tuesday, April 5th, at noon CDT**; 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.

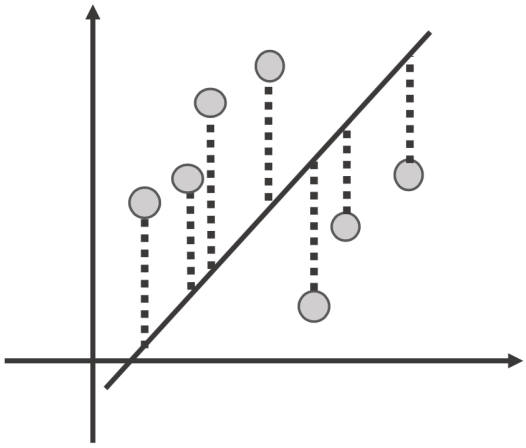
**Version History.**

1.0 Initial Version.

# 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.

(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, $\Sigma$ 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 $\Sigma$ in Eq. (1)? Also, explicitly write down the values of $\Sigma$ given the dataset.

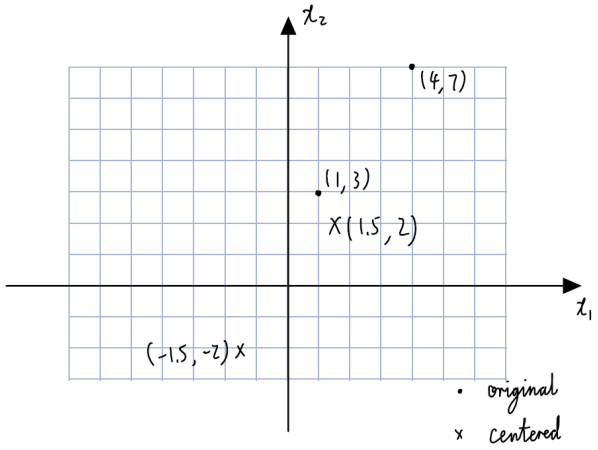
(d) What is the optimal $w$ and the optimal value of the program in Eq. (1) given

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

**Solution**

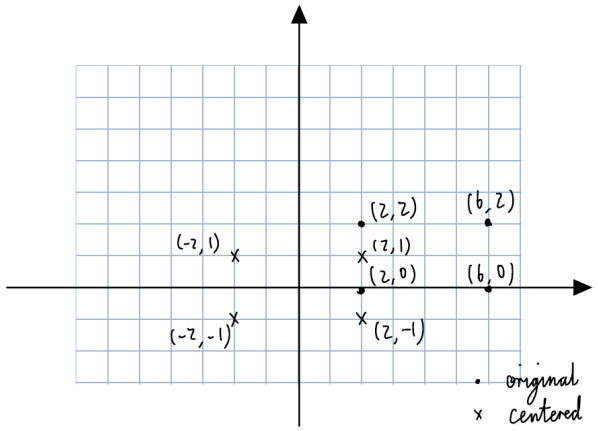
(a)
- False, PCA minimizes sum of squared orthogonal distance instead of vertical distance, which is minimized by linear regression.
- False, PCA seeks a projection that best represents the data in orthogonal axis, in which the variances are maximized.
- True.
- False. First of all, the goal of PCA is to find principal axis that are orthogonal to each other. Finding principal axis is equivalent to finding eigenvectors of $\sigma$, which is orthogonal to each other.

(b) The data points:




$$\bar{X} = \begin{bmatrix} -1.5 & 1.5 \\ -2 & 2 \end{bmatrix} \Rightarrow \Sigma = \frac{1}{|D|}\bar{X}\bar{X}^\top = \begin{bmatrix} 2.25 & 3 \\ 3 & 2.25 \end{bmatrix} \Rightarrow w_1 = (0.6, 0.8)$$

(c) The data points:




$\Sigma \in \mathbb{R}^{2\times 2}$

$$\begin{aligned}
\Sigma &= \frac{1}{|D|}\bar{X}\bar{X}^\top \\
&= \frac{1}{4}\begin{bmatrix} 2 & 2 & -2 & -2 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 2 & -1 \\ -2 & 1 \\ -2 & -1 \end{bmatrix} \\
&= \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

(d) The optimal $w = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

The optimal value of the program is:

$$w^\intercal \Sigma w = 20$$

## 2. Gaussian Mixture Models

Consider a Gaussian mixture model with $K$ components ($k \in \{1, \ldots, K\}$), each having mean $\mu_k$, variance $\sigma_k^2$, and mixture weight $\pi_k$. 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 $\mu_k$, $\sigma_k$, $\pi_k$, $K$, $x_i$, and $\mathcal{D}$)?

(b) Assume $K = 1$, find the maximum likelihood estimate for the parameters ($\mu_1$, $\sigma_1^2$, $\pi_1$).

(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 $\mu_k$ and variance $\sigma_k^2$.

(e) How are k-Means and Gaussian Mixture Model related? (There are three conditions)
**Hint:** Think of variance, $\pi_k$, and hard/soft assignment.

(f) 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's rule.

(g) Consider the modified Gaussian Mixture Model objective:

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

Conclude that the objective for k-Means is the 0-temperature limit of Gaussian Mixture Model.
**Hint:** Let $F_k = (x - \mu_k)^2$ and apply the equation you proved in (f).

**Solution**

(a) The log likelihood (without negative) is given by:

$$\log \prod_{i \in \mathcal{D}} p(x^{(i)}|\pi, \mu, \sigma) = \sum_{i \in \mathcal{D}} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x^{(i)}|\mu_k, \sigma_k) = \sum_{i \in \mathcal{D}} \log \sum_{k=1}^{K} \pi_k \frac{1}{\sigma_k \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x^{(i)} - \mu_k}{\sigma_k}\right)^2}$$

(b) When K = 1, it's just the single Gaussian model, and $\pi_1$ must be 1, so the log likelihood becomes:

$$\sum_{i \in \mathcal{D}} \log \mathcal{N}(x^{(i)}|\mu_1, \sigma_1)$$

Maximizing the log likelihood is equivalent to minimizing negative log likelihood:

$$\begin{aligned}
\underset{\mu_1, \sigma_1}{\operatorname{argmax}} \sum_{i \in \mathcal{D}} \log \mathcal{N}(x^{(i)}|\mu_1, \sigma_1) &= \underset{\mu_1, \sigma_1}{\operatorname{argmin}} - \sum_{i \in \mathcal{D}} \log \mathcal{N}(x^{(i)}|\mu_1, \sigma_1) \\
&= \underset{\mu_1, \sigma_1}{\operatorname{argmin}} \left( \sum_{i \in \mathcal{D}} \frac{1}{2\sigma_1^2} (x^{(i)} - \mu_1)^2 \right) + \frac{N}{2} \log(2\pi\sigma_1^2)
\end{aligned}$$

First, the program is convex to $\mu_1$ and $\sigma_1$. Then we can take gradient and set to 0 to try to get the close form solution:

$$\begin{cases} \frac{\partial}{\partial \mu_1} = \frac{1}{\sigma^2} \sum_{i \in \mathcal{D}} (x^{(i)} - \mu_1) = 0 \\ \frac{\partial}{\partial \sigma_1} = -\frac{1}{\sigma^3} \sum_{i \in \mathcal{D}} (x^{(i)} - \mu_1)^2 + \frac{N}{\sigma} = 0 \end{cases} \Rightarrow \begin{cases} \mu_1 = \frac{1}{N} \sum_{i \in \mathcal{D}} x^{(i)} \\ \sigma_1^2 = \frac{1}{N} \sum_{i \in \mathcal{D}} (x^{(i)} - \mu_1)^2 \end{cases}$$

So the solution is:

$$\begin{cases} \pi_1 = 1 \\ \mu_1 = \frac{1}{N} \sum_{i \in \mathcal{D}} x^{(i)} \\ \sigma_1^2 = \frac{1}{N} \sum_{i \in \mathcal{D}} (x^{(i)} - \mu_1)^2 \end{cases}$$

(c)

$$p(z_{i,k} = 1) = \pi_k, \ k \in \{1, ..., K\}, z_{i,k} \in \{0, 1\}, \sum_{k=1}^{K} \pi_{i,k} = 1$$

Thus:

$$p(z_{i,1}, z_{i,2}, ..., z_{i,K}) = \prod_{k=1}^{K} \pi_K^{z_{i,k}}$$

Called marginal distribution of $z_{i,k}$ when $z_{i,k} = 1$.

(d) According to Bayes rule:

$$\begin{aligned}
p(z_{ik} = 1|x^{(i)}) &= \frac{p(x^{(i)}|z_{ik} = 1) p(z_{ik} = 1)}{\sum_{k'=1}^{K} p(x^{(i)}|z_{ik'} = 1) p(z_{ik'} = 1)} \\
&= \frac{\pi_k \mathcal{N}(x^{(i)}|\mu_k, \sigma_k)}{\sum_{k'=1}^{K} \pi'_k \mathcal{N}(x^{(i)}|\mu'_k, \sigma'_k)}
\end{aligned}$$

(e) i. Gaussian Mixture Model is the soft assignment version of k-Means. Using posterior which is not hard-coded to 1, we can generate data that can belong to different cluster center.

ii. When $\sigma_k$ approaches 0, responsibility (or posterior in special case) $r_{ik}$ will be 1 for a given data point $x^{(i)}$ and its closest cluster center $\mu_k$, analogous to k-mean that does hard assignment.

(f)

$$
\begin{aligned}
&\lim_{\epsilon \to 0} -\epsilon \log \sum_{k=1}^{K} e^{-\frac{F_k}{\epsilon}} \\
=&\lim_{\epsilon \to 0} \frac{\log \sum_{k=1}^{K} e^{-\frac{F_k}{\epsilon}}}{-\epsilon} \\
=&\lim_{\epsilon \to 0} \frac{\frac{\sum_{k=1}^{K} \frac{F_k}{\epsilon^2} e^{-\frac{F_k}{\epsilon}}}{\sum_{k=1}^{K} e^{-\frac{F_k}{\epsilon}}}}{\frac{1}{\epsilon^2}} \\
=&\lim_{\epsilon \to 0} \frac{\sum_{k=1}^{K} F_k e^{-\frac{F_k}{\epsilon}}}{\sum_{k=1}^{K} e^{-\frac{F_k}{\epsilon}}} \\
=&\lim_{\epsilon \to 0} \frac{\sum_{k=1}^{K} F_k e^{\frac{min - F_k}{\epsilon}}}{\sum_{k=1}^{K} e^{\frac{min - F_k}{\epsilon}}} \\
=&\lim_{\epsilon \to 0} \frac{min}{1} \\
=&min
\end{aligned}
$$

Where $min$ refers to $\min_k F_k$.

(g)

$$
\begin{aligned}
&\min_{\mu} \lim_{\epsilon \to 0} \sum_{x_i \in \mathcal{D}} -\epsilon \log \sum_{k=1}^{K} e^{-\frac{(x_i - \mu_k)^2}{\epsilon}} \\
=&\min_{\mu} \sum_{x_i \in \mathcal{D}} \min_k (x_i - \mu_k)^2
\end{aligned}
$$

In words, this means that the program will assign the closest cluster center for a given data point, and minimize the sum of squared distance between the cluster center and its assigned points.
This is what happens when we set $\epsilon$, i.e. standard deviation, close to 0 for all Gaussian models in Gaussian Mixture Model.

# 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 $x_d$ in the $D$-dimensional vector $x$ 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 $p \triangleq \{p^{(1)}, \cdots, p^{(K)}\}$. Let $\pi_k$ denote the probability that the $k^{\text{th}}$ set of Bernoulli parameters is chosen. Assume a distribution $\pi \triangleq \{\pi_1, \ldots, \pi_K\}$ over all possible Bernoulli parameters. Write an expression for $P(x^{(i)}|p, \pi)$, as a function of $\pi_k$ and $P(x^{(i)}|p^{(k)})$.

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

(b) **Expectation step**.

i. Let $z^{(i)} \triangleq [z_1^{(i)}, \ldots, z_K^{(i)}] \in \{0,1\}^K$ be an indicator vector, such that $z_k^{(i)} = 1$ if $x^{(i)}$ was drawn from a Bernoulli($p^{(k)}$), and 0 otherwise. Write down the expression of $P(z^{(i)}|\pi)$ as a function of $\pi_k$ 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. Let $Z = \{z^{(i)}\}_{i=1}^n$. 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. $p$, $\pi$ 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:

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

**Hint:** $\tilde{\pi}$ needs to be a distribution. Use Lagrange multiplier to include this constraint into your objective function and solve it.

**Solution**

(a) i.

$$\begin{cases} P(x_d = 1) = q_d \\ P(x_d = 0) = 1 - q_d \end{cases} \Rightarrow P(x_d) = q_d^{x_d}(1-q_d)^{1-x_d} \Rightarrow P(x|q) = \prod_{d=1}^{D} q_d^{x_d}(1-q_d)^{1-x_d}$$

ii.

$$P(x^{(i)}|p,\pi) = \sum_{k=1}^{K} \pi_k P(x^{(i)}|p^{(k)})$$

iii.

$$\begin{aligned} \log P(\mathcal{D}|\pi,p) &= \log \prod_{i\in\mathcal{D}} P(x^{(i)}|p,\pi) \\ &= \sum_{i\in\mathcal{D}} \log \sum_{k=1}^{K} \pi_k P(x^{(i)}|p^{(k)}) \end{aligned}$$

(b) i.

$$p(z_k^{(i)} = 1) = \pi_k \Rightarrow p(z^{(i)}|\pi) = \prod_{k=1}^{K} \pi_k^{z_k^{(i)}}$$

ii.

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

iii. $Z$ is completely independent of $p$, thus $P(z^{(i)}|\pi) = P(z^{(i)}|\pi,p)$, then we have:

$$\begin{aligned} P(Z,\mathcal{D}|\pi,p) &= \prod_{i=1}^{n} P(z^{(i)},x^{(i)}|\pi,p) \\ &= \prod_{i=1}^{n} P(x^{(i)}|z^{(i)},p,\pi)P(z^{(i)}|\pi,p) \\ &= \prod_{i=1}^{n} P(x^{(i)}|z^{(i)},p,\pi)P(z^{(i)}|\pi) \\ &= \prod_{i=1}^{n}\prod_{k=1}^{K} \pi_k^{z_k^{(i)}} P(x^{(i)}|p^{(k)})^{z_k^{(i)}} \end{aligned}$$

iv. Since $z_k^{(i)} \in \{0,1\}$, we have:

$$\begin{aligned} \eta(z_k^{(i)}) &= \mathbb{E}[z_k^{(i)}|x^{(i)},\pi,p] \\ &= P(z_k^{(i)} = 1|x^{(i)},\pi,p) \times 1 + P(z_k^{(i)} = 0|x^{(i)},\pi,p) \times 0 \\ &= P(z_k^{(i)} = 1|x^{(i)},\pi,p) \end{aligned}$$

Besides, from ii.:

$$P(x^{(i)}|z_k^{(i)} = 1) = P(x^{(i)}|p^{(k)})$$

So, by Bayes Rule, we have:

$$\begin{aligned}\eta(z_k^{(i)}) &= P(z_k^{(i)} = 1 | x^{(i)}, \pi, p) \\ &= \frac{P(x^{(i)}|z_k^{(i)} = 1)P(z_k^{(i)} = 1)}{\sum_{j=1}^{K} P(x^{(i)}|z_j^{(i)} = 1)P(z_j^{(i)} = 1)} \\ &= \frac{\pi_k \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1 - x_d^{(i)}}}{\sum_{j=1}^{K} \pi_j \prod_{d=1}^{D} (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^{(j)})^{1 - x_d^{(i)}}}\end{aligned}$$

v.

$$\begin{aligned}\log P(Z, \mathcal{D}|\tilde{\pi}, \tilde{p}) &= \log \prod_{i=1}^{n} \prod_{k=1}^{K} \tilde{\pi}_k^{z_k^{(i)}} P(x^{(i)}|\tilde{p}^{(k)})^{z_k^{(i)}} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_k^{(i)} (\log \tilde{\pi}_k + \log P(x^{(i)}|\tilde{p}^{(k)})) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_k^{(i)} (\log \tilde{\pi}_k + \log \prod_{d=1}^{D} (\tilde{p}_d^{(k)})^{x_d^{(i)}} (1 - \tilde{p}_d^{(k)})^{1 - x_d^{(i)}}) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_k^{(i)} \left( \log \tilde{\pi}_k + \sum_{d=1}^{D} (x_d^{(i)} \log \tilde{p}_d^{(k)} + (1 - x_d^{(i)}) \log(1 - \tilde{p}_d^{(k)})) \right)\end{aligned}$$

Thus the expectation is given by setting $z_k^{(i)}$ to its expectation, i.e. $\eta(z_k^{(i)})$, and we have:

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

(c) i.

$$\begin{aligned}&\frac{\partial \mathbb{E}[\log P(Z, \mathcal{D}|\tilde{p}, \tilde{\pi})|\mathcal{D}, p, \pi]}{\partial \tilde{p}_d^{(k)}} \\ =& \sum_{i=1}^{n} \eta(z_k^{(i)}) \left( \frac{x_d^{(i)}}{\tilde{p}_d^{(k)}} - \frac{1 - x_d^{(i)}}{1 - \tilde{p}_d^{(k)}} \right) \\ =& \sum_{i=1}^{n} \eta(z_k^{(i)}) \frac{x_d^{(i)} - \tilde{p}_d^{(k)}}{\tilde{p}_d^{(k)}(1 - \tilde{p}_d^{(k)})} \\ =& \frac{\sum_{i=1}^{n} \eta(z_k^{(i)}) x_d^{(i)} - \tilde{p}_d^{(k)} \sum_{i=1}^{n} \eta(z_k^{(i)})}{\tilde{p}_d^{(k)}(1 - \tilde{p}_d^{(k)})} = 0\end{aligned}$$

$$\Downarrow$$

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

$$\Downarrow$$

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

ii. Constraints on $\tilde{\pi}$:

$$\begin{cases} \tilde{\pi}_k \geqslant 0, k \in \{1, ..., K\} \\ \sum_{k=1}^{K} \tilde{\pi}_k = 1 \end{cases}$$

Incorporating one constraint, the Lagrangian is formed as:

$$\mathcal{L}(\tilde{\pi}, \lambda) = \sum_{i=1}^{n} \sum_{k=1}^{K} \eta(z_k^{(i)}) \left( \log \tilde{\pi}_k + \sum_{d=1}^{D} (x_d^{(i)} \log \tilde{p}_d^{(k)} + (1 - x_d^{(i)}) \log(1 - \tilde{p}_d^{(k)})) \right) + \lambda \left( \sum_{k=1}^{K} \tilde{\pi}_k - 1 \right)$$

So:

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

$$\Downarrow$$

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

$$\Downarrow$$

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

$$\Downarrow$$

$$\tilde{\pi}_k \propto N_k$$

Satisfying $\sum_{k=1}^{K} \tilde{\pi}_k = 1$, we have:

$$\tilde{\pi}_k = \frac{N_k}{\sum_{k'=1}^{K} N_{k'}}$$

The results satisfy $\tilde{\pi}_k \geqslant 0, k \in \{1, ..., K\}$ automatically since $\eta(z_k^{(i)}) \geqslant 0$.

# 4. K-Means 1

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

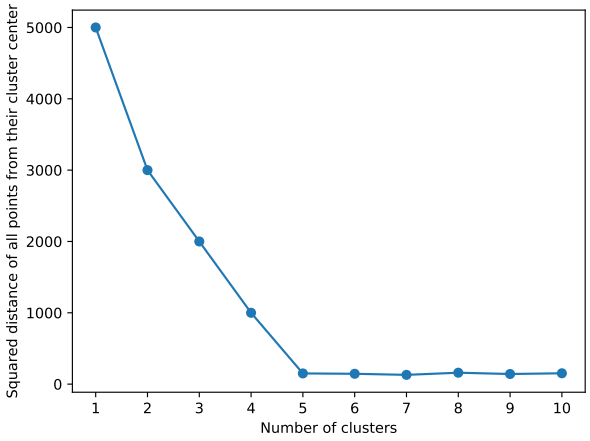
(b) Assume that you are trying to cluster data points $x_i$ for $i \in \{1, 2, \ldots, D\}$ into $K$ clusters each with center $\mu_k$ where $k \in \{1, 2, \ldots, K\}$. The objective function for doing this clustering involves minimization of the Euclidean distance between the points and the cluster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

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

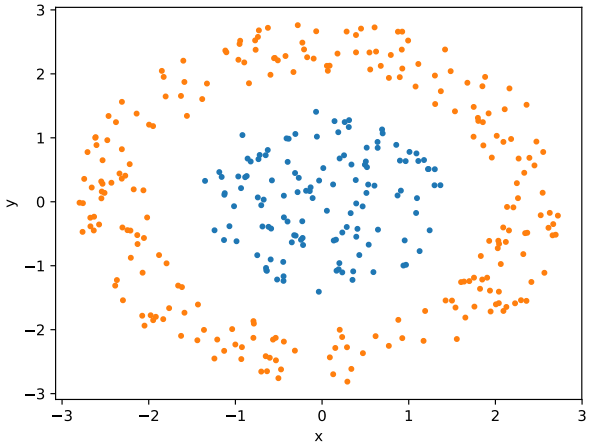
How do you ensure hard assignment of one data point to one and only one cluster at a given time?
**Hint:** By hard assignment we mean that you are 100 % sure that a point either belongs or doesn't belong to a cluster.

(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) Un-supervised learning because the data has no label(s).

(b) By enforcing the following constraints on $r_{ik}$: For $i \in \{1, ..., n\}$:

$$\begin{cases} r_{ik} \in \{0, 1\} \\ \sum_{k=1}^{K} r_{ik} = 1 \end{cases}$$

(c) It will be change into: For $i \in \{1, ..., n\}$:

$$\begin{cases} r_{ik} \geqslant 0 \\ \sum_{k=1}^{K} r_{ik} = 1 \end{cases}$$

(d) K = 5

(e) No.
Because the centers of coordinates of blue and orange data points will be close to each other, making reassignment of closest points on blue points for both clusters, although color intensity (if it's a input dimension) could distinguish the centers.

# 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 $\mu_k$ ($k \in \{1, \ldots, K\}$) via the $k$-Means algorithm. This algorithm optimizes the following cost function:

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

(a) What is the domain for $\mu_k$?

(b) Given fixed cluster centers $\mu_k$ $\forall k \in \{1, \ldots, K\}$, what is the optimal $r_{x,k}$ for the program in Eq. 2? Provide a reason?

(c) Given fixed $r_{x,k}$ $\forall x \in \mathcal{D}, k \in \{1, \ldots, K\}$, what are the optimal cluster centers $\mu_k$ $\forall k \in \{1, \ldots, K\}$ for the program in Eq. 2?

**Hint:** Reason by first computing the derivative w.r.t $\mu_k$.

(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)

$$r_{x,k} = \begin{cases} 1, \text{ if } k = \underset{k \in \{1,\dots,K\}}{\operatorname{argmin}} \|x - \mu_k\|^2 \\ 0, \text{ otherwise} \end{cases}$$

Because every $x$ gets assigned to its nearest cluster center, minimizing $\sum_{x \in \mathcal{D}} \|x - \mu_k\|^2$ for a give $k$.

(c)

$$\frac{\partial \left[ \sum_{x \in \mathcal{D}, k \in \{1,\dots,K\}} \frac{1}{2} r_{x,k} \|x - \mu_k\|_2^2 \right]}{\partial \mu_k} = \sum_{x \in \mathcal{D}} r_{x,k}(x - \mu_k) = 0$$

$$\Downarrow$$

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

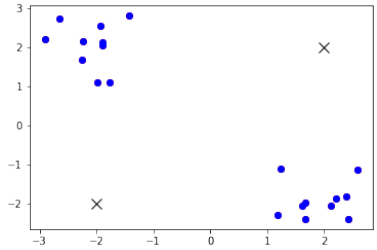
$\mu_k$ will be updated to the geometric center(mean) of its assigned data pionts.

**Algorithm 1** K-Means

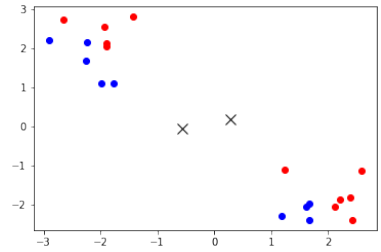
(d) **procedure** K-MEANS($num\_iterations, \mathcal{D}$) ▷ number of iterations and dataset
  initialize $\mu$ to random points
  **for** $i = 0; i < num\_iterations; ++i$ **do** ▷ iterate num_iteration times
    **for** $x \in \mathcal{D}$ **do** ▷ re-assign points to clusters
      $smallest\_distance \leftarrow +\inf$
      $last\_smallest\_index \leftarrow 1$
      **for** $k \in \{1, ..., K\}$ **do**
        **if** $\|x - \mu_k\|^2 < smallest\_distance$ **then**
          $smallest\_distance \leftarrow \|x - \mu_k\|^2$
          $r_{x,last\_smallest\_index} \leftarrow 0$
          $r_{x,k} \leftarrow 1$
          $last\_smallest\_index \leftarrow k$
        **else**
          $r_{x,k} \leftarrow 0$
        **end if**
      **end for**
    **end for**
    **for** $k \in \{1, ..., K\}$ **do** ▷ adjust cluster centers
      $current\_sum\_points \leftarrow \vec{0}$
      $current\_num\_points \leftarrow 0$
      **for** $x \in \mathcal{D}$ **do**
        $current\_sum\_points \leftarrow current\_sum\_points + r_{x,k} \times \vec{x}$
        $current\_num\_points \leftarrow current\_num\_points + r_{x,k}$
      **end for**
      $\mu_k \leftarrow \frac{current\_sum\_points}{current\_num\_points}$
    **end for**
  **end for**
  **return** $r, \mu$
**end procedure**

This algorithm is guaranteed to converge, because every step decreases the cost function, and the cost function is lower bounded by 0. It might not find the global optimum, since alternating approach is still different from gradient descent.

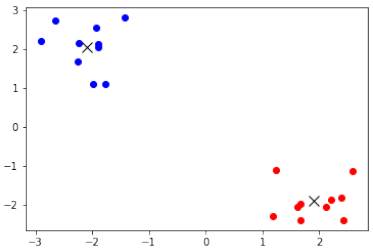
(e) The results are:



0th iteration, centers: (2, 2) (-2, -2)



1st iteration
centers: (0.2911, 0.1694) (-0.5545, -0.0518)
cost = 82.2666



>1st iteration
centers: (1.9163, -1.9143) (-2.0952, 2.0540)
cost = 4.5600