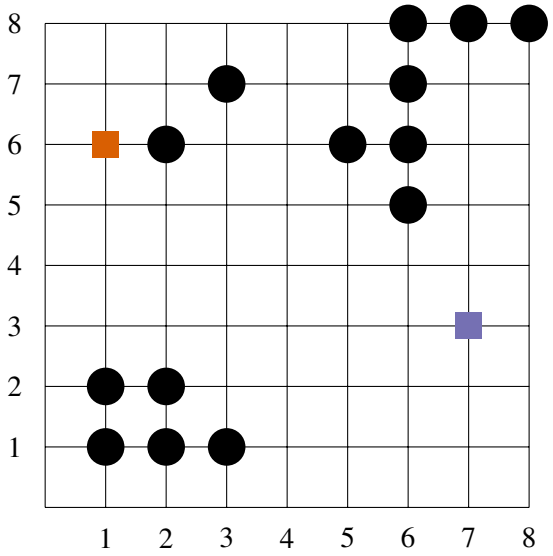
**Ludwig-Maximilians-Universität München**
**Institut für Informatik**
Prof. Dr. Thomas Seidl
Janina Sontheim, Maximilian Hünemörder

**Knowledge Discovery in Databases**
WS 2019/20

## Exercise 6: $k$-Means, $k$-Modes, $k$-Medoids (PAM), EM

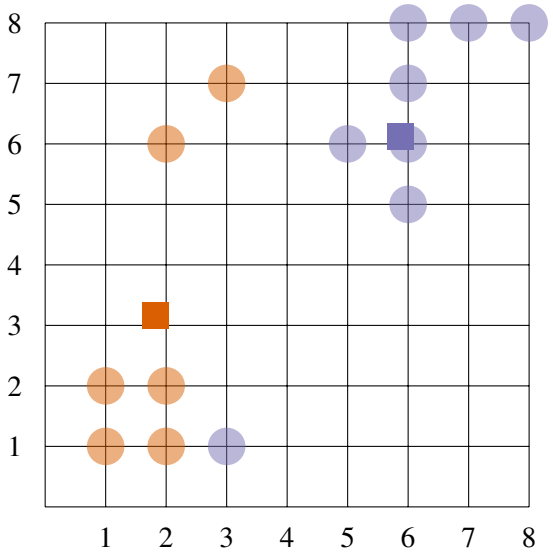
### Exercise 6-1 $k$-Means

Given the following data set with 14 objects in $\mathbb{R}^2$ (the black dots):

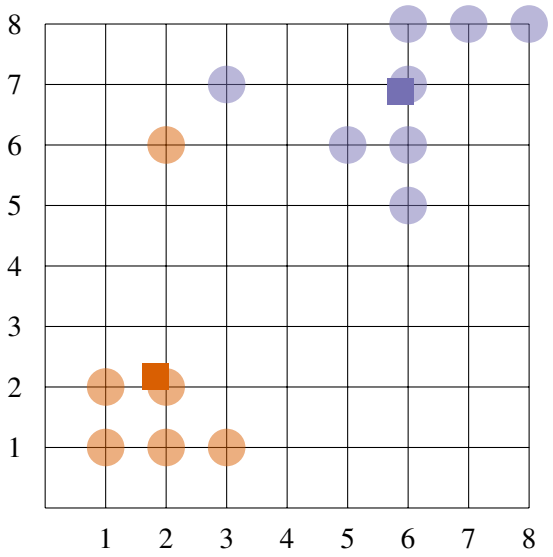



Compute a partitioning into $k = 2$ clusters using the $k$-means algorithm. As initial representatives use the red and violet square. Start with computing the initial assignment. Explain and draw the assignments as well as the updated centroids after each step.

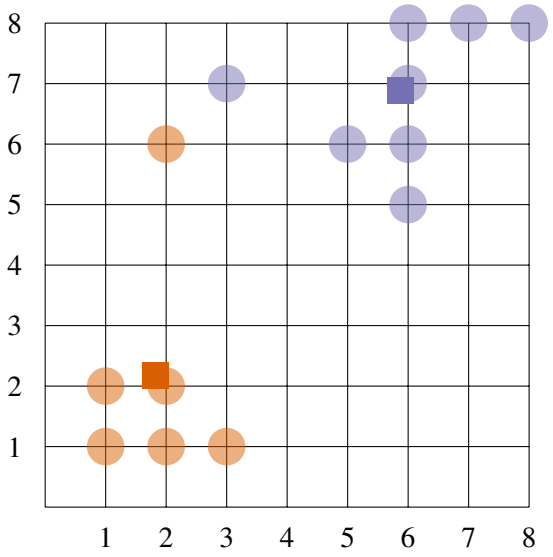
The initial assignment and the updated means are given by



Updating the assignments and subsequently the means yields:

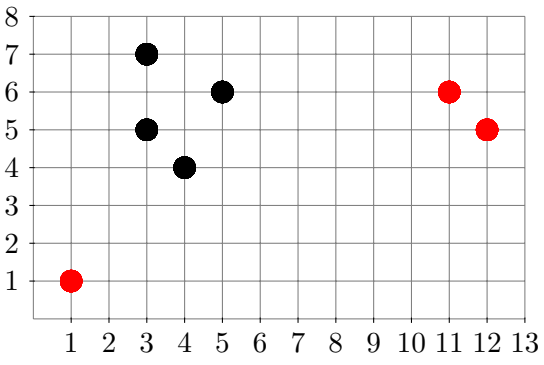



In the next iteration, the assignment does not change anymore. Hence, the means also stay the same and the final result is given by:




**Exercise 6-2** $k$**-Means**

Given the following data set with 7 objects in $\mathbb{R}^2$ represented by the black and red dots:

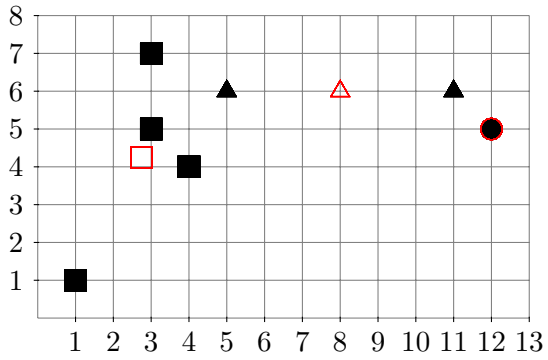


In the following, we would like to compute complete partitionings of the dataset into $k = 3$ clusters using the $k$-means algorithm.

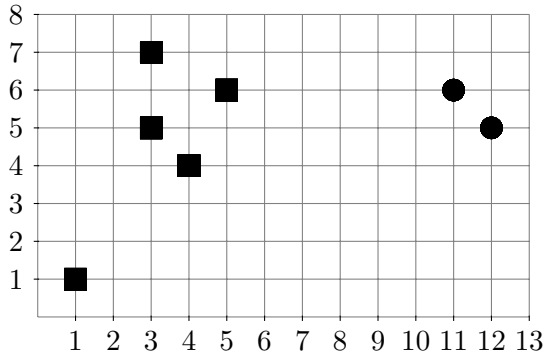
Let the initial cluster centers be given by the points marked in red. Carry out the $k$-Means algorithm as presented in the lecture. Which problem arises?

One of the clusters becomes empty!

First round of assignments, new centers:




Second round of assignments:




Cluster “triangle” is empty – what should be the new cluster center??

Possible workarounds for such a degenerate case would be to simply restart the algorithm with a different initialization, to remove the empty cluster from consideration and continue with $k-1$ clusters, or to introduce a new cluster center somewhere far away from the existing ones. Empty clusters usually occur as a consequence of bad initialization. Sensible initialization and running the algorithm for multiple iterations are in general important for the success of $k$-means, not only to prevent empty clusters.

**Exercise 6-3** $k$**-Mode**

Given the following Dataset of 15 Persons with their Jobs and Pets:

| Name | Job | Pet |
|---|---|---|
| James | Programmer | Cat |
| Hans | Manager | None |
| Marcel | Programmer | Snake |
| Sebastian | Cook | None |
| Max | Technician | Cat |
| Michael | Cook | Cat |
| Anna | Manager | Dog |
| Friederike | Manager | None |
| Sarah | Programmer | Snake |
| Florian | Advisor | None |
| Theresa | Programmer | Cat |
| Jonas | Manager | None |
| Julian | Programmer | Cat |
| Nadine | Programmer | Dog |
| Thomas | Manager | None |

Compute a partitioning using the k-Modes algorithm by Huang, Z. (Link). For initial modes choose a technician, who owns a snake and an advisor, who owns a dog.

From the paper we can find out that the k-mode algorithm is very similiar to the k-means algorithm, but using a trivial distance measure and a multivariate mode. Formally, given two objects X and Y described by m categorical features, the distance between them is calculated by:

$$d(X, Y) = \sum_{i=1}^{m} \delta(x_i, y_i)$$

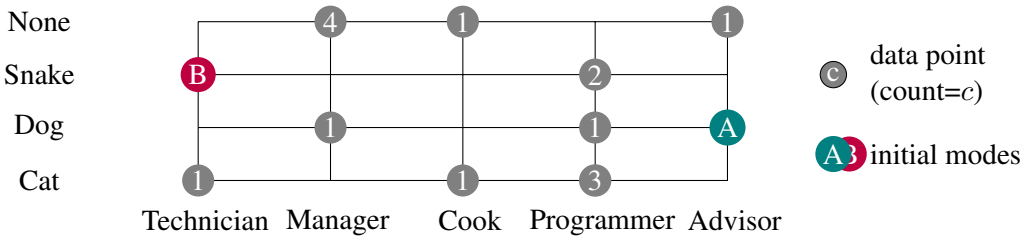
where $\delta(a, b)$ is:

$$\delta(a, b) = \begin{cases} 0 & \text{if } a = b \\ 1 & \text{if } a \neq b \end{cases}$$

Looking at this equation, we can ignore the NameFeature from now on, since it only leads to the distance of each person being incremented by one and thereby does not impact the clustering in any way.

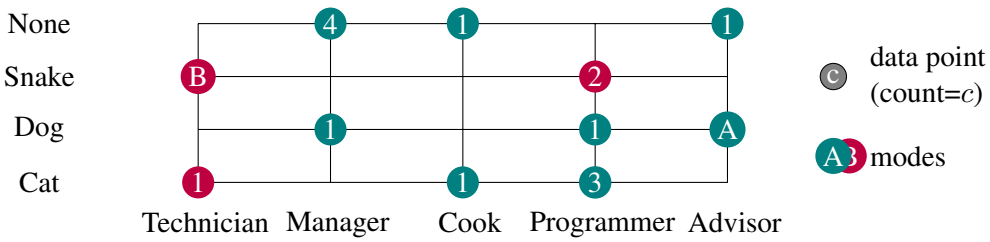
The mode of a set X is defined as the vector Q that minimizes the sum of distances from itself to each point in X.

At the beginning the dataset and the two modes look like this:

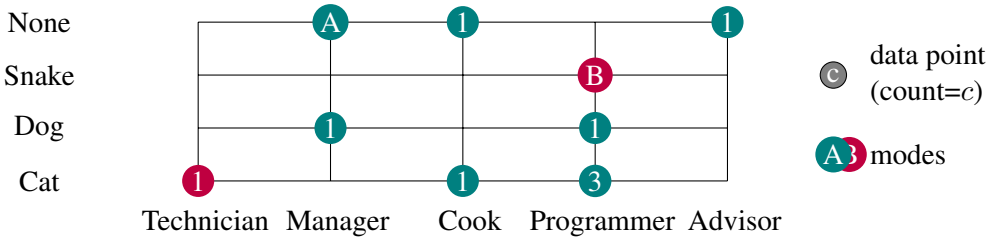


Now we have to calculate the distance of each object to the two modes and assign them to the mode with the smaller distance. In our case our two modes are A: [Advisor, Dog] and B: [Technician, Snake]. If both distances are the same we will assign the object to A.

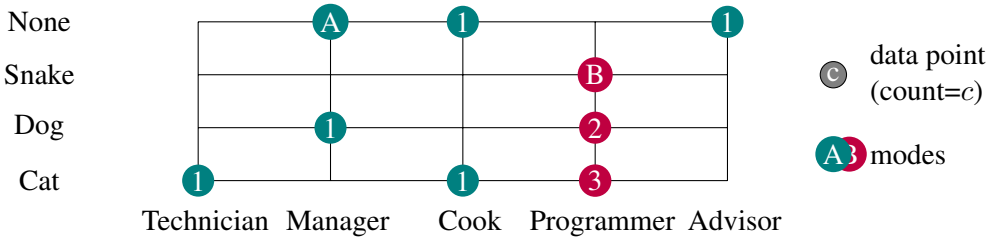
| Name | Job | Pet | d(X, A) | d(X, B) | Assignment |
|---|---|---|---|---|---|
| James | Programmer | Cat | 2 | 2 | A |
| Hans | Manager | None | 2 | 2 | A |
| Marcel | Programmer | Snake | 2 | 1 | B |
| Sebastian | Cook | None | 2 | 2 | A |
| Max | Technician | Cat | 2 | 1 | B |
| Michael | Cook | Cat | 2 | 2 | A |
| Anna | Manager | Dog | 1 | 2 | A |
| Friederike | Manager | None | 2 | 2 | A |
| Sarah | Programmer | Snake | 2 | 1 | B |
| Florian | Advisor | None | 1 | 2 | A |
| Theresa | Programmer | Cat | 2 | 2 | A |
| Jonas | Manager | None | 2 | 2 | A |
| Julian | Programmer | Cat | 2 | 2 | A |
| Nadine | Programmer | Dog | 2 | 2 | A |
| Thomas | Manager | None | 2 | 2 | A |



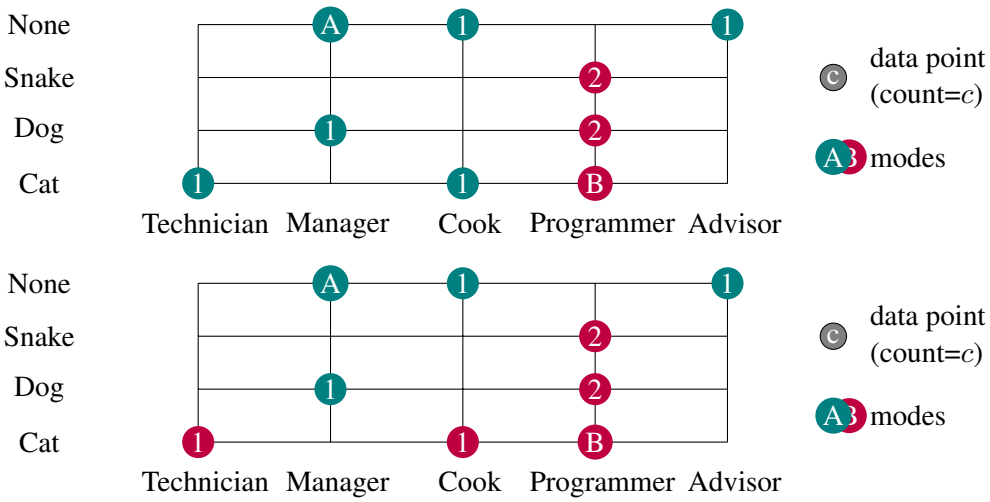

Now we recalculate the new modes for each cluster.




Now we have to calculate the distance of each object to the two modes and reassign them if the distance is smaller



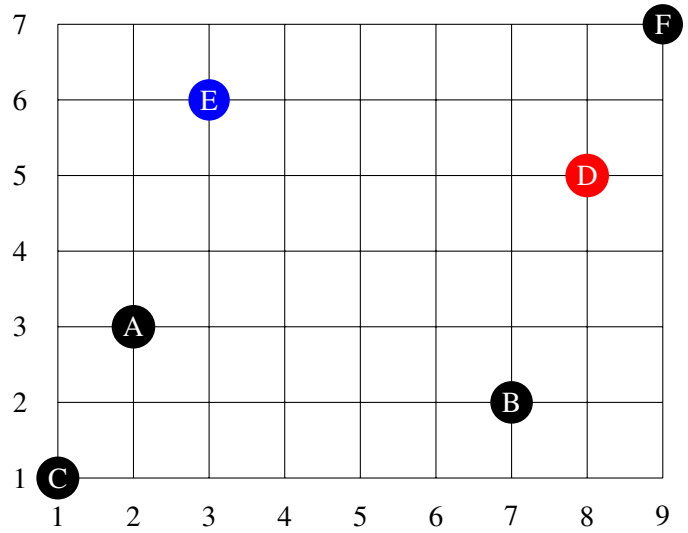

And we continue until nothing can change anymore.




**Exercise 6-4** **K-Medoid (PAM)**

Consider the following 2-dimensional data set:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| $x_1$ | 2 | 7 | 1 | 8 | 3 | 9 |
| $x_2$ | 3 | 2 | 1 | 5 | 6 | 7 |




(a) Perform the first loop of the PAM algorithm ($k = 2$) using the Manhattan distance. Select $D$ and $E$ (highlighted in the plot) as initial medoids and compute the resulting medoids and clusters.
**Hint**: When $C(m)$ denotes the cluster of medoid $m$, and $M$ denotes the set of medoids, then the total distance $TD$ may be computed as

$$TD = \sum_{m \in M} \sum_{o \in C(m)} d(m, o)$$

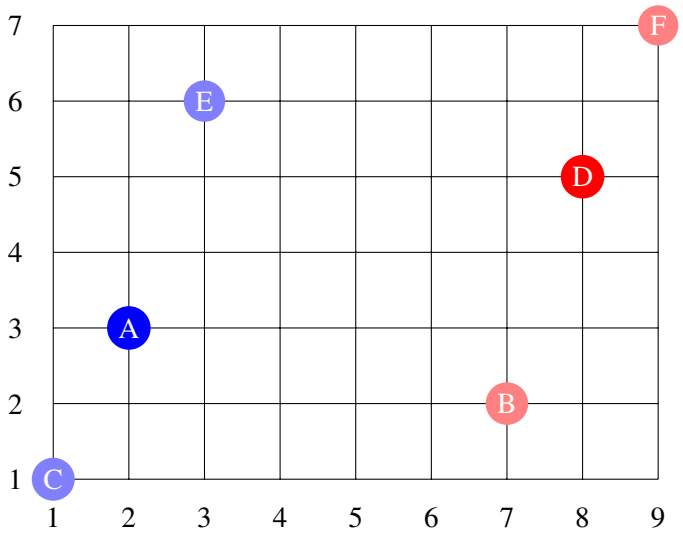
We have the following distance values (values which are clear by symmetry and reflexivity are left out):

| | B | C | D | E | F |
|---|---|---|---|---|---|
| A | 6 | 3 | 8 | 4 | 11 |
| B | | 7 | 4 | 8 | 7 |
| C | | | 11 | 7 | 14 |
| D | | | | 6 | 3 |
| E | | | | | 7 |

The following table shows assignments and $TD_{m \leftrightarrow n}$ value for each pair $(m, n) \in M \times N$ with $M = \{D, E\}$ and $N = \{A, B, C, F\}$.

| Medoids | | Assignment | | | | | | TD |
|---|---|---|---|---|---|---|---|---|
| $m_1$ | $m_2$ | A | B | C | D | E | F | |
| D | E | 1 | 0 | 1 | 0 | 1 | 0 | **18** |
| D | A | 1 | 0 | 1 | 0 | 1 | 0 | **14** |
| D | B | 1 | 1 | 1 | 0 | 0 | 0 | 22 |
| D | C | 1 | 0 | 1 | 0 | 0 | 0 | 16 |
| D | F | 0 | 0 | 0 | 0 | 0 | 1 | 29 |
| E | A | 1 | 1 | 1 | 0 | 0 | 0 | 22 |
| E | B | 0 | 1 | 0 | 1 | 0 | 0 | 22 |
| E | C | 1 | 1 | 1 | 0 | 0 | 0 | 23 |
| E | F | 0 | 1 | 0 | 1 | 0 | 1 | 21 |

The table shows that swapping $E$ and $A$ yields the largest improvement in terms of $TD$. The updated clustering after the first iteration is shown in the following figure.




(b) How can the clustering result $C_1 = \{A, B, C\}, C_2 = \{D, E, F\}$ be obtained with the PAM algorithm ($k = 2$) using the weighted Manhattan distance

$$d(x, y) = w_1 \cdot |x_1 - y_1| + w_2 \cdot |x_2 - y_2|?$$

Assume that B and E are the initial medoids and give values for the weights $w_1$ and $w_2$ for the first and second dimension respectively.

Consider $(w_1, w_2) = (0, 1)$, i.e. the distance is computed solely based on the second dimension. Then, we can use the reduced data set:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| $x_2$ | 3 | 2 | 1 | 5 | 6 | 7 |

Hence, we have the following distance values (values which are clear by symmetry and reflexivity are left out):

| | B | C | D | E | F |
|---|---|---|---|---|---|
| A | 1 | 2 | 2 | 3 | 4 |
| B | | 1 | 3 | 4 | 5 |
| C | | | 4 | 5 | 6 |
| D | | | | 1 | 2 |
| E | | | | | 1 |

The following tables shows assignments and $TD$ values for the initial setting as well as for all $m \leftrightarrow n$ for $(m, n) \in M \times N$ with $M = \{B, E\}$, and $N = \{A, C, D, F\}$:

| Medoids | | Assignment | | | | | | TD |
|---|---|---|---|---|---|---|---|---|
| $m_1$ | $m_2$ | A | B | C | D | E | F | |
| B | E | 0 | 0 | 0 | 1 | 1 | 1 | **4** |
| B | A | 1 | 0 | 0 | 1 | 1 | 1 | 10 |
| B | C | 0 | 0 | 1 | 0 | 0 | 0 | 13 |
| B | D | 0 | 0 | 0 | 1 | 1 | 1 | 5 |
| B | F | 0 | 0 | 0 | 1 | 1 | 1 | 5 |
| A | E | 1 | 1 | 1 | 0 | 0 | 0 | 5 |
| C | E | 1 | 1 | 1 | 0 | 0 | 0 | 5 |
| D | E | 1 | 1 | 1 | 1 | 0 | 0 | 10 |
| F | E | 0 | 0 | 0 | 0 | 0 | 1 | 13 |

Hence, these medoids are the final ones and the partitioning stays stable.

**Exercise 6-5 Assignments in EM-Algorithm**

Given a data set with 100 points consisting of three Gaussian clusters $A$, $B$ and $C$ and the point $p$.

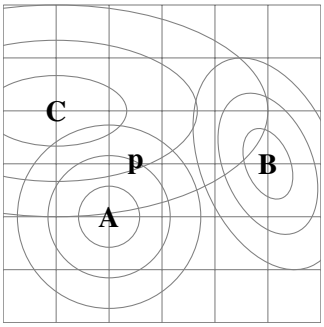
The cluster $A$ contains $30\%$ of all objects and is represented using the mean of all his points $\mu_A = (2, 2)$ and the covariance matrix $\Sigma_A = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$.

The cluster $B$ contains $20\%$ of all objects and is represented using the mean of all his points $\mu_B = (5, 3)$ and the covariance matrix $\Sigma_B = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$.

The cluster $C$ contains $50\%$ of all objects and is represented using the mean of all his points $\mu_C = (1, 4)$ and the covariance matrix $\Sigma_C = \begin{pmatrix} 16 & 0 \\ 0 & 4 \end{pmatrix}$.

The point $p$ is given by the coordinates $(2.5, 3.0)$.
Compute the three probabilities of $p$ belonging to the clusters $A$, $B$ and $C$.

The following sketch is not exact, and only gives a rough idea of the cluster locations:




We have

$$\gamma_i(p) := \pi_i \cdot \mathcal{N}(p \mid \mu_i, \Sigma_i) = \pi_i \cdot \frac{1}{\sqrt{(2\pi)^2 \det(\Sigma_i)}} \exp\left(-\frac{1}{2}(p-\mu_i)^T \Sigma_i^{-1}(p-\mu_i)\right) \tag{1}$$

Substituting $\pi_i, \mu_i, \Sigma_i$ by the given parameters for each cluster yields:

**Cluster A** For $A$ we have $\pi_A = \frac{3}{10}$, $\Sigma_A = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$, and $\mu_A = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$. First, we compute

$$\det(\Sigma_A) = \det \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} = 3^2 = 9.$$

Hence, we have

$$\Sigma_A^{-1} = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}^{-1} = \frac{1}{9}\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Furthermore,

$$p - \mu_A = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 2 \end{pmatrix},$$

and moreover

$$(p-\mu_A)^T \Sigma_A^{-1}(p-\mu_A) = \frac{1}{2 \cdot 3 \cdot 2}\begin{pmatrix} 1 \\ 2 \end{pmatrix}^T \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \frac{1^2 + 2^2}{12} = \frac{5}{12}.$$

Finally, we can use these values together with (1) to obtain

$$\begin{aligned} \gamma_A(p) &= \pi_A \cdot \frac{1}{\sqrt{(2\pi)^2 \det(\Sigma_A)}} \exp\left(-\frac{1}{2}(p-\mu_A)^T \Sigma_A^{-1}(p-\mu_A)\right) \\ &= \frac{1}{20\pi} \exp\left(-\frac{5}{24}\right) \\ &\approx 0.0129223682965846 \end{aligned}$$

**Cluster B** For $B$ we have $\pi_B = \frac{1}{5}$, $\Sigma_B = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$, and $\mu_B = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$. First, we compute

$$\det(\Sigma_B) = \det \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix} = 2 \cdot 4 - 1 \cdot 1 = 7.$$

Hence, we have

$$\Sigma_B^{-1} = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}^{-1} = \frac{1}{7}\begin{pmatrix} 4 & -1 \\ -1 & 2 \end{pmatrix}.$$

Furthermore,

$$p - \mu_B = \begin{pmatrix} 2.5 \\ 3 \end{pmatrix} - \begin{pmatrix} 5 \\ 3 \end{pmatrix} = -\frac{1}{2}\begin{pmatrix} 5 \\ 0 \end{pmatrix},$$

and moreover,

$$(p-\mu_B)^T \Sigma_B^{-1} (p-\mu_B) = \frac{1}{2\cdot 7\cdot 2} \begin{pmatrix}5\\0\end{pmatrix}^T \begin{pmatrix}4 & -1\\-1 & 2\end{pmatrix} \begin{pmatrix}5\\0\end{pmatrix} = \frac{1}{28} \begin{pmatrix}5\\0\end{pmatrix}^T \begin{pmatrix}20\\-5\end{pmatrix} = \frac{100}{28} = \frac{25}{7}.$$

Finally, we can use these values together with (1) to obtain

$$\begin{aligned}\gamma_B(p) &= \pi_B \cdot \frac{1}{\sqrt{(2\pi)^2 \det(\Sigma_B)}} \exp\left(-\frac{1}{2}(p-\mu_B)^T \Sigma_B^{-1} (p-\mu_B)\right)\\ &= \frac{1}{10\sqrt{7}\pi} \exp\left(-\frac{25}{14}\right)\\ &\approx 0.00201732210214117\end{aligned}$$

**Cluster C** For $C$ we have $\pi_C = \frac{1}{2}$, $\Sigma_C = \begin{pmatrix}16 & 0\\0 & 4\end{pmatrix}$, and $\mu_C = \begin{pmatrix}1\\4\end{pmatrix}$. First, we compute

$$\det(\Sigma_C) = \det\begin{pmatrix}16 & 0\\0 & 4\end{pmatrix} = 16\cdot 4 = 64.$$

Hence, we have

$$\Sigma_C^{-1} = \begin{pmatrix}16 & 0\\0 & 4\end{pmatrix}^{-1} = \frac{1}{64}\begin{pmatrix}4 & 0\\0 & 16\end{pmatrix} = \frac{1}{16}\begin{pmatrix}1 & 0\\0 & 4\end{pmatrix}.$$

Furthermore,

$$p-\mu_C = \begin{pmatrix}2.5\\3\end{pmatrix} - \begin{pmatrix}1\\4\end{pmatrix} = \frac{1}{2}\begin{pmatrix}3\\-2\end{pmatrix},$$

and moreover,

$$(p-\mu_C)^T \Sigma_C^{-1} (p-\mu_C) = \frac{1}{2\cdot 16\cdot 2} \begin{pmatrix}3\\-2\end{pmatrix}^T \begin{pmatrix}1 & 0\\0 & 4\end{pmatrix} \begin{pmatrix}3\\-2\end{pmatrix} = \frac{1}{64} \begin{pmatrix}3\\-2\end{pmatrix}^T \begin{pmatrix}3\\-8\end{pmatrix} = \frac{25}{64}.$$

Finally, we can use these values together with (1) to obtain

$$\begin{aligned}\gamma_C(p) &= \pi_C \cdot \frac{1}{\sqrt{(2\pi)^2 \det(\Sigma_C)}} \exp\left(-\frac{1}{2}(p-\mu_C)^T \Sigma_C^{-1} (p-\mu_C)\right)\\ &= \frac{1}{32\pi} \exp\left(-\frac{25}{128}\right)\\ &\approx 0.00818233032076434\end{aligned}$$

Given that the point was generated by the model, these probabilities are divided by

$$\gamma = \sum_{c\in\{A,B,C\}} \gamma_c \approx 0.0231220207194901,$$

yielding:

$$\begin{aligned}\gamma'_A &= \frac{\gamma_A}{\gamma} \approx \frac{0.0129223682965846}{0.0231220207194901} \approx 55.89\%\\ \gamma'_B &= \frac{\gamma_B}{\gamma} \approx \frac{0.00201732210214117}{0.0231220207194901} \approx 8.72\%\\ \gamma'_C &= \frac{\gamma_C}{\gamma} \approx \frac{0.00818233032076434}{0.0231220207194901} \approx 35.39\%\end{aligned}$$