Einführung in die Künstliche Intelligenz

Übungszettel 7

Prof. Dr. Claudia Schon

C.Schon@hochschule-trier.de

Fachbereich Informatik
Hochschule Trier

1 K-Means Clustering[1]

Gegeben ist die Punktmenge $X = \{x_0, x_1, x_2, x_3, x_4, x_5\}$ mit den Koordinaten:

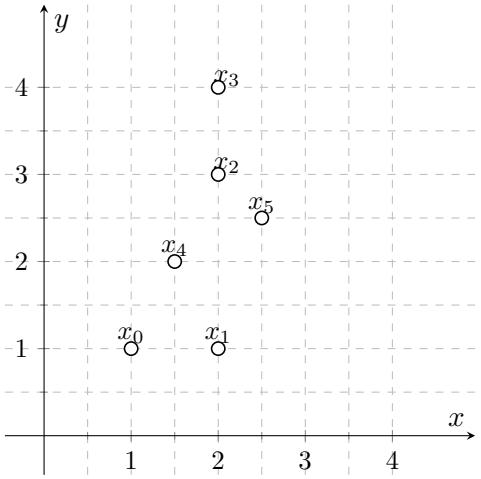
$$\begin{aligned} x_0 &= (1,1) & x_3 &= (2,4) \\ x_1 &= (2,1) & x_4 &= (1.5,2) \\ x_2 &= (2,3) & x_5 &= (2.5,2.5) \end{aligned}$$

Führen Sie den K-Means Algorithmus mit $k = 2$ durch. Wir gehen dabei davon aus, dass die Clusterzentren initial zufällig wie folgt zugewiesen wurden: $C = \{c_0, c_1\}$ mit $c_0 = x_2$ und $c_1 = x_3$. Füllen Sie die folgenden Tabellen aus:

Iteration 1

x_i	$\|x_i - c_0\|^2$	$\|x_i - c_1\|^2$	z_i
x_0			
x_1			
x_2			
x_3			
x_4			
x_5			

Zeichnen Sie im Koordinatensystem die Clusterzentren und die Zugehörigkeit zu den Clustern farblich ein.



Berechnen Sie nun die neuen Clusterzentren:

$c_0 =$ _____

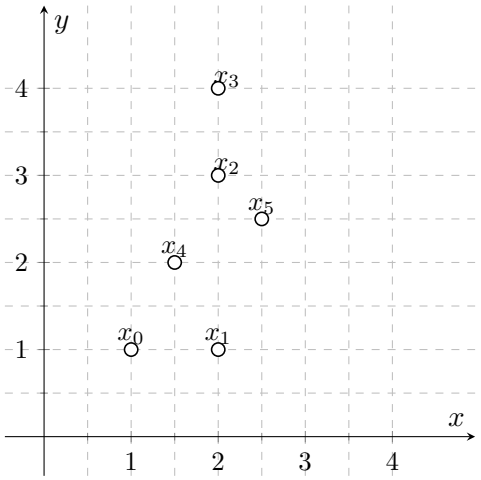
$c_1 =$ _____

[1] Die Aufgabe wurde mit Unterstützung von ChatGPT erstellt.

Iteration 2

x_i	$\|x_i - c_0\|^2$	$\|x_i - c_1\|^2$	z_i
x_0			
x_1			
x_2			
x_3			
x_4			
x_5			

Zeichnen Sie im Koordinatensystem die Clusterzentren und die Zugehörigkeit zu den Clustern farblich ein.



Berechnen Sie nun die neuen Clusterzentren:

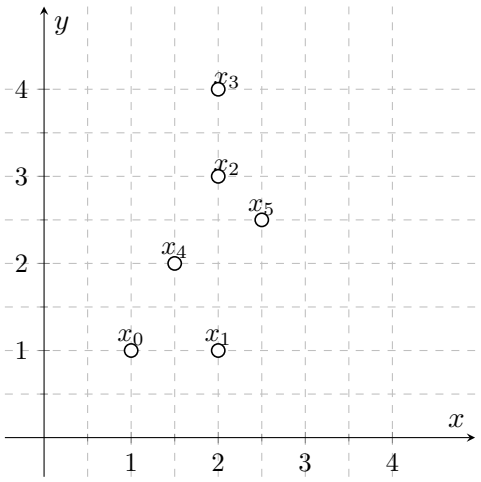
$c_0 =$ _____

$c_1 =$ _____

Iteration 3

x_i	$\|x_i - c_0\|^2$	$\|x_i - c_1\|^2$	z_i
x_0			
x_1			
x_2			
x_3			
x_4			
x_5			

Zeichnen Sie im Koordinatensystem die Clusterzentren und die Zugehörigkeit zu den Clustern farblich ein.



Berechnen Sie nun die neuen Clusterzentren:

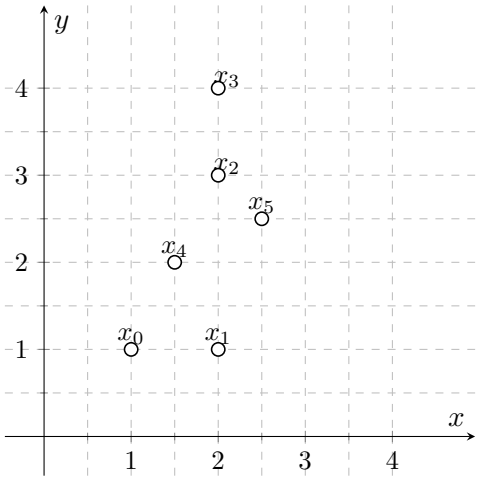
$c_0 =$ _____

$c_1 =$ _____

Iteration 4

x_i	$\|x_i - c_0\|^2$	$\|x_i - c_1\|^2$	z_i
x_0			
x_1			
x_2			
x_3			
x_4			
x_5			

Zeichnen Sie im Koordinatensystem die Clusterzentren und die Zugehörigkeit zu den Clustern farblich ein.



Berechnen Sie nun die neuen Clusterzentren:

$c_0 =$ _____

$c_1 =$ _____

Berechnen Sie den *Sum of Squares Error (SSE)* für diese Clustereinteilung.

2 K-Means in Python

Wir wollen nun den in der Vorlesung vorgestellten K-Means Algorithmus in Python implementieren. Sie finden den Pseudocode des Algorithmus in den Vorlesungsfolien.

Außerdem finden Sie auf Stud.IP ein Rahmenprogramm (im zip `uebungszettel7.zip` liegt das Programm `clustering_rahmen.py` sowie die erforderlichen Requirements) auf das Sie aufbauen können. Natürlich können Sie auch ohne das Rahmenprogramm arbeiten.

1. Implementieren Sie den K-Means Algorithmus und testen ihn mit dem Beispiel aus der vorherigen Aufgabe.
2. Erweitern Sie Ihre Implementierung so, dass am Ende des Algorithmus der SSE für die gefundene Einteilung in Cluster berechnet wird.

3 Einfluss der Initialisierung auf das Ergebnis von K-Means[2]

Erweitern Sie den Punktdatensatz aus Aufgabe 1 um den zusätzlichen Punkt

$$x_6 = (4.5,\ 4.5)$$

Führen Sie den K-Means Algorithmus mit $k = 2$ für folgende zwei Initialisierungen der Clusterzentren durch:

- **Variante A:** $c_0 = x_1$, $c_1 = x_6$
- **Variante B:** $c_0 = x_2$, $c_1 = x_5$

Nutzen Sie dafür Ihre Implementierung aus der vorherigen Aufgabe.

1. Führen Sie jeweils das Clustering durch, bis der Algorithmus konvergiert. Welche Clusterzentren wurden bestimmt?
2. Vergleichen Sie die finale Zuordnung der Punkte zu den Clustern bei den beiden Varianten. Visualisieren Sie die Clusterzugehörigkeiten der beiden Varianten in einem Koordinatensystem.
3. Berechnen Sie für das Ergebnis jeder Variante den *Sum of Squared Errors (SSE)*.
4. Bewerten Sie die Lösungen: Welche wirkt aus Sicht des SSE günstiger? Welche Clusteraufteilung scheint sinnvoller?
5. Ändern Sie Ihre Implementierung so ab, dass alle möglichen Kombinationen von Initialisierung der Clusterzentren ausprobiert werden und bestimmen Sie jeweils eine finale Clustereinteilung. Dokumentieren Sie:
   - Die gewählten Startzentren,
   - die finale Zuordnung der Punkte zu den Clustern,
   - die berechneten Clusterzentren nach Konvergenz und
   - den zugehörigen SSE.

   Wie viele verschiedene Ergebnisse finden Sie?

[2] Die Aufgabe wurde mit Unterstützung von ChatGPT erstellt.

4 Bestimmung einer geeigneten Clusteranzahl mit der Elbow-Methode[3]

In dieser Aufgabe sollen Sie den K-Means Algorithmus auf eine zweidimensionale Punktmenge anwenden, die aus einer CSV-Datei geladen wird. Ziel ist es, mit Hilfe der *Elbow-Methode* eine sinnvolle Anzahl von Clustern zu bestimmen.

Sie finden auf Stud.IP das Archiv `uebungszettel7.zip` in dem die Datei `kmeans_blobs.csv` liegt. Darin sind Punkt in zwei Dimensionen. Sie enthält keine Label-Informationen, sondern nur die Spalten `x` und `y`. Erstellen Sie nun eine Variante Ihrer Implementierung, indem Sie Ihre bestehende Implementierung wie folgt erweitern:

1. Lesen Sie die Datei ein und speichern Sie die Punkte als Liste von `numpy.array`-Objekten:

```
1: import numpy as np
2: import pandas as pd
3:
4: df = pd.read_csv("kmeans_blobs.csv")
5: X = [np.array([row.x, row.y]) for row in df.itertuples(index=False)]
```

2. Führen Sie K-Means für $k = 2$ bis 7 durch. Protokollieren Sie dabei den *Sum of Squared Errors (SSE)* für jedes k.
3. Erstellen Sie ein Diagramm, das SSE in Abhängigkeit von k zeigt.
4. Bestimmen Sie mit Hilfe der Elbow-Methode eine sinnvolle Anzahl an Clustern. Begründen Sie Ihre Wahl.
5. Visualisieren Sie das finale Clustering. Schreiben Sie eine Funktion, die die Punktmenge entsprechend der Clusterzuweisung farbig darstellt. Sie können dafür den auf der Rückseite abgebildeten folgenden Beispielcode verwenden.

[3] Die Aufgabe wurde mit Unterstützung von ChatGPT erstellt.

```
import matplotlib.pyplot as plt

def plot_clusters(X, z):
    X = np.array(X)
    z = np.array(z)
    plt.figure(figsize=(6, 6))
    for cluster_id in np.unique(z):
        cluster_points = X[z == cluster_id]
        plt.scatter(
            cluster_points[:, 0],
            cluster_points[:, 1],
            s=40,
            label=f"Cluster {cluster_id}"
        )
    plt.xlabel("x")
    plt.ylabel("y")
    plt.title("Clustervisualisierung")
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```