Institut für Informationsverarbeitung (TNT)
Appelstraße 9a, 30167 Hannover
www.tnt.uni-hannover.de

Prof. Dr.-Ing. Bodo Rosenhahn
Felix Winkler, M.Sc.

WiSe 2022/23 **23.11.2022**

Praktische Übung: Matlab für die medizinische und industrielle Bildinterpretation

Versuch 5: Clustering mit k-Means und Mean-Shift

Aufgabe 1 - k-Means Clustering

a) Schreiben Sie eine Funktion `L = kmeans(X, w, k)`, die Datenpunkte **beliebiger** Dimension in `k` verschiedene Cluster aufteilt. Sie können sich bei der Umsetzung an den folgenden Pseudo-Code orientieren:

Input: Datenpunkte $x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$
Input: Gewichte $w_i \in \mathbb{R}^+,\ i = 1, \ldots, n$
Input: Anzahl Clusterzentren k
Output: Label $l_i \in \{1, \ldots, k\},\ i = 1, \ldots, n$ für jeden Datenpunkt

1: Initialisiere zufällige Clusterzentren $c_j \leftarrow random\{x_i\},\ j = 1, \ldots, k$
2: **repeat**
3: $\quad c_j^{old} \leftarrow c_j,\ j = 1, \ldots, k$
4: $\quad$ Bestimme dichtestes Clusterzentrum für alle Datenpunkte x_i:

$$l_i \leftarrow \underset{j=1,\ldots,k}{\operatorname{argmin}} \|x_i - c_j\|_2,\ i = 1, \ldots, n$$

5: $\quad$ Aktualisiere Clusterzentren (Schwerpunkt der zugeordneten Datenpunkte):

$$c_j \leftarrow \frac{\sum_{i \in P_j} w_i x_i}{\sum_{i \in P_j} w_i} \quad \text{mit} \quad P_j = \{z \mid l_z = j\},\ j = 1, \ldots, k$$

6: **until** $c_j = c_j^{old},\ j = 1, \ldots, k$

Der k-Means Algorithmus durchläuft demnach die folgenden Schritte:

1. Bestimme `k` zufällige Datenpunkte als initiale Clusterzentren.
2. Ordne jeden Datenpunkt dem nächsten Clusterzentrum zu.
3. Bestimme den Schwerpunkt der einzelnen Cluster als neue Clusterzentren, wobei die Punkte anhand ihrer Gewichte eingehen.
4. Wiederhole die Schritte 2.+3. solange sich die Clusterzentren ändern.

Hinweise zur Implementierung:

- In MATLAB ist keine `repeat-until`- bzw. `do-while`-Schleife vorhanden. Passen Sie den gegebenen Code dementsprechend an.
- Mit `randperm` können Sie zufällige Integer ohne Wiederholung erstellen.
- Zur Erinnerung: Die Funktion `min` liefert nicht nur das Minimum, sondern bei Bedarf auch die Position, kann also auch als argmin verwenden werden.

b) Nutzen Sie ihre Funktion, um die 4 verschiedenen Mengen `X1,X2,X3,X4` in der Datei `test_data_clustering.mat` geeignet zu clustern. Setzen Sie die Gewichte hier für alle Punkte auf den Wert 1. Achten Sie auf eine sinnvolle Auswahl der Clusteranzahl. Zur einfachen Darstellung der Ergebnisse können Sie die Funktion `plotClusterResults(X, L)` verwenden.

c) Bei Daten, die sich gut darstellen lassen, kann die Clusteranzahl visuell festgelegt werden. Schwieriger ist dies bei höherdimensionalen Daten wie z.B. `X5`. Eine Möglichkeit zur Auswahl der Clusteranzahl ist die *Elbow Method*. Hierzu wird die Summe der quadrierten Abstände innerhalb der Cluster (*within-cluster sum of squares*)

$$wcss = \sum_{i=1}^{n} \|x_i - c_{l_i}\|^2$$

bestimmt und in Abhängigkeit der Clusteranzahl `k` dargestellt. Eine sinnvolle Clusteranzahl zeichnet sich üblicherweise durch einen "Ellenbogenknick" aus.

Schreiben Sie eine Funktion `elbowMethod(X, kMin, kMax)`, die diese Methode umsetzt. Eine optimale Clusteranzahl muss nicht automatisch bestimmt werden, eine Darstellung von *wcss* in Abhängigkeit von `k` ist ausreichend. Bestätigen Sie die Clusteranzahl, die Sie in Aufgabenteil b) verwendet haben und schätzen Sie anschließend die Anzahl der Cluster in der Menge `X5`.

Hinweise: *wcss* kann direkt berechnet werden, während der k-Means Algorithmus läuft.

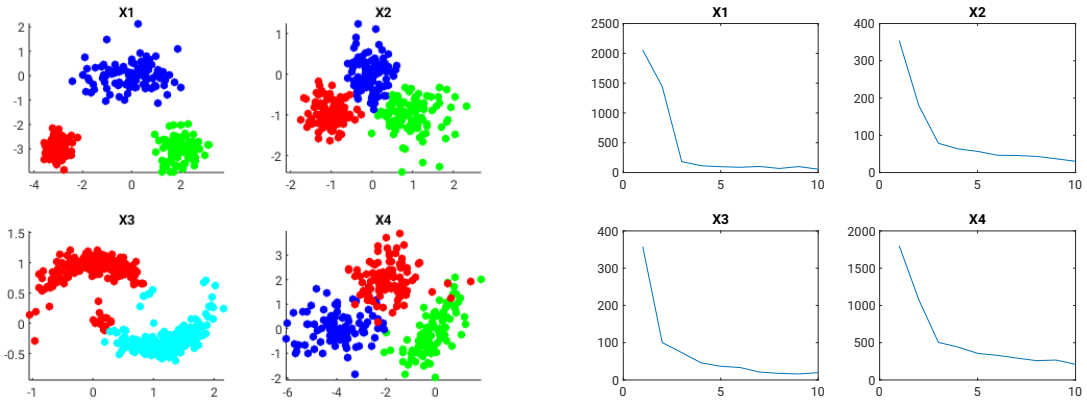

Abbildung 1: Beispielhafte Ergebnisse des k-Means Clusterings und der *Elbow Method*

Aufgabe 2 - Mean-Shift Clustering

Machen Sie sich mit der gegebenen Funktion `meanshift.m` vertraut und nutzen Sie diese um die gegebenen Daten `X1-X4` ebenfalls zu clustern. Verifizieren Sie zudem die Anzahl der von der *Elbow Method* gefundenen Cluster in `X5`. Nutzen Sie die Methode anschließend, um die Mehrfachdetektionen aus Versuch 4, Aufgabe 2 zu entfernen.
Hinweis: Sie können den Parameter `vis` auf 0 setzen, um die Darstellung zu unterdrücken und das Verfahren zu beschleunigen.

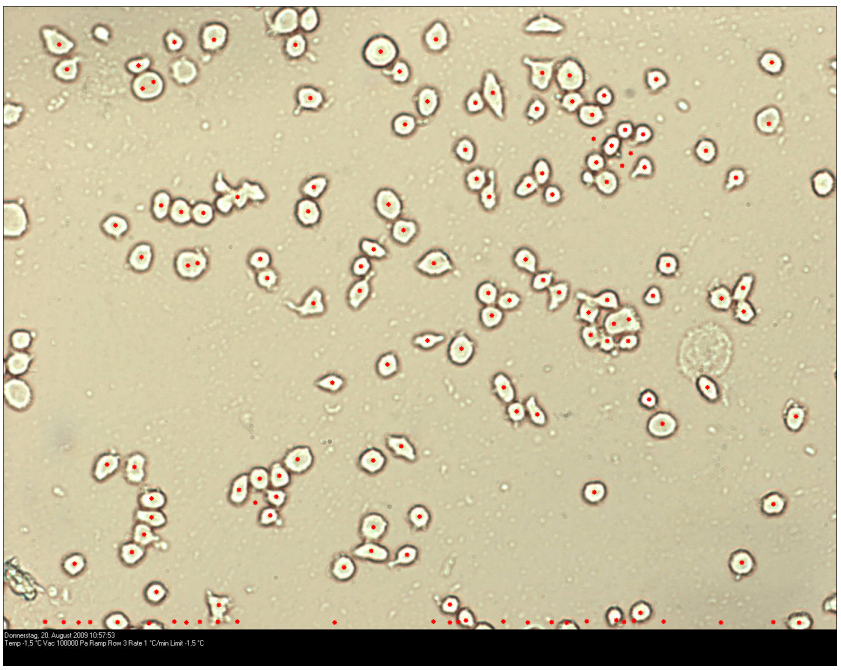

Abbildung 2: Mit Mean-Shift gefilterte Zelldetektionen aus Versuch 4, Aufgabe 2

Aufgabe 3 - Clustering zur Bildsegmentierung

Nutzen Sie k-Means, um ein Bild zu segmentieren. Prinzipiell lässt sich der Grauwert jedes Bildpunkts als eindimensionaler Datenpunkt ansehen. Das Verfahren kann jedoch deutlich beschleunigt werden, wenn die Grauwerte unter Berücksichtigung ihrer Häufigkeit geclustert werden.
Bestimmen Sie dazu das Histogramm des Graustufenbildes (`imhist`). Als Eingabe für den k-Means Algorithmus können Sie nun die Grauwerte (0-255) als Datenpunkte und die entsprechenden Gewichte aus dem Histogramm verwenden. Anschließend kann ein Labelbild erstellt werden, indem Sie - abhängig von seinem Grauwert - jeden Bildpunkt einem Cluster zuordnen.
Stellen Sie sowohl das Originalbild als auch das Segmentierungsergebnis dar, indem Sie den Grauwert jeden Bildpunktes auf den Mittelwert des entsprechenden Clusters (das Clusterzentrum) setzen.

Abbildung 3: Mit k-Means segmentiertes Bild bei Einbeziehung der Grauwerthäufigkeit

Kontrollfragen

a) Warum lässt sich die Menge `X3` nicht korrekt mit dem k-Means Algorithmus clustern?

b) Wieso liefert die *Elbow Method* nicht in jedem Fall ein eindeutiges Ergebnis?

c) Nennen Sie Unterschiede zwischen k-Means und Mean-Shift. Was sind die Vorteile der einzelnen Verfahren?