

Visual and Scientific Computing

K-Means und Nächste Nachbarn Klassifikatoren

WiSe 2019/20

Das Lernziel dieser Übungsaufgabe ist es, sich mit Verfahren des Supervised und Unsupervised Learning auseinanderzusetzen und diese auch zu implementieren.

Aufgabe 1: K-Means (6 Punkte)

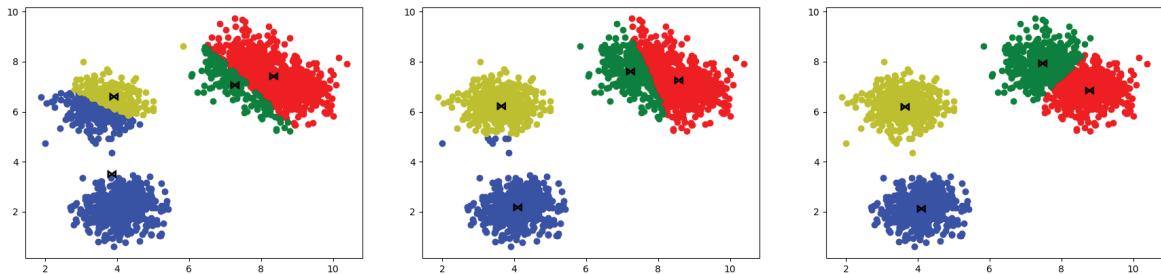


Abbildung 1: Ergebnisbild. Darstellung der Cluster und Cluster-Zentren in unterschiedlichen Iterationen.

Wie in der Vorlesung diskutiert ist K-Means ein häufig eingesetztes Verfahren zur Clusteranalyse in Datenmengen. Dieses sollen Sie in dieser Aufgabe implementieren. Abbildung 1 zeigt eine mögliche Clusterzuordnung nach mehreren Iterationen. Dieses Verfahren kann benutzt werden um repräsentative Punkte in einer Datenmenge zu berechnen. Das ist zum Beispiel nötig um die Anzahl der Farben im Bild zu quantisieren und GIF-Bilder zu erstellen. Sie sollen das Verfahren selber implementieren, sodass Sie selber diese Bilder erstellen können. Ein Beispiel ist in Abbildung 2 zu sehen. Implementieren Sie

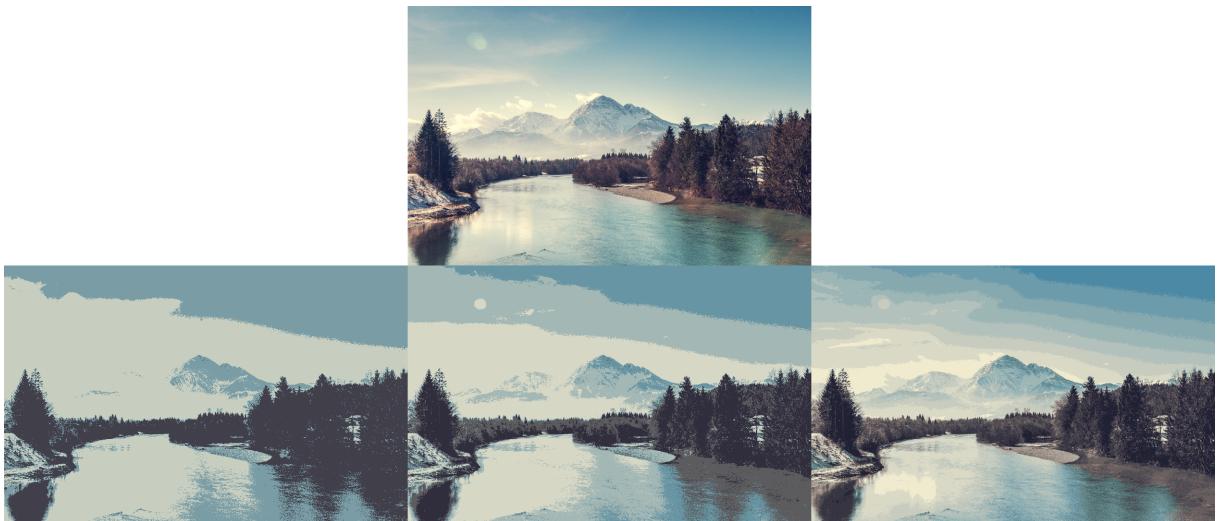


Abbildung 2: Oben das Original-Bild. Unten das Bild mit reduzierten Farben $k=3$, $k=5$ und $k=16$.

dazu die Funktion `k_means` in der Datei `k_means.py`. Dazu sind folgende Schritte notwendig:

- Implementieren Sie den `initialize`-Funktion. Sie berechnet den Initialisierungsschritt und erzeugt k zufällige Clusterzentren.
- Implementieren Sie die `assign_to_center`-Funktion. Sie ordnet alle Datenpunkte zu ihren nächsten Clusterzentren zu.
- Implementieren Sie die `update_mean`-Funktion. Sie aktualisiert die Clusterzentren mit dem Mittelwert aller zugeordneten Datenpunkten.

Der Algorithmus sollte nach maximal 20 Iterationen oder optional nach einer einstellbaren Minimaländerung des Fehlers abbrechen. Die Minimaländerung kann z.B. 1% Fehleränderung zur vorherigen Iteration betragen. Am Ende der Datei befindet sich Beispiel-Code, wie die Funktion `k_means` aufgerufen werden soll. Dort befindet sich auch Beispielcode für die Bilderstellung.

Achten Sie darauf, dass Ihre Funktion mit beliebigen Dimensionen funktionieren muss. Der Algorithmus soll zum Beispiel sowohl mit den 2 Dimensionalen Beispiel-Datenpunkten als auch mit den 3 Dimensionalen Pixel-Datenpunkten arbeiten können.

Aufgabe 2: K-Nächste Nachbarn Klassifikation (4 Punkte)



Abbildung 3: Bilder der Datenbank. Links die drei Test-Bilder. Rechts die Trainings-Bilder. Die labels sind `car`, `face` und `flower` von oben nach unten.

Die Klassifikationsmethode auf Basis der K-Nächsten Nachbarn wird häufig als Baseline für die Einschätzung einer Klassifikationsgenauigkeit eingesetzt. Wir wollen in dieser Aufgabe einen sehr einfache Bildklassifikator implementieren. Dieser besteht aus 20 Beispielbildern, die jeweils 256x256 Pixel groß sind und es gibt insgesamt 3 verschiedene Klassen: `car`, `face`, `flower` (wie in Abbildung 3 zu sehen). Dabei sind die Pixel des Bilder jeweils die Featurevektoren bzw. Datenpunkte, d.h. wir fassen die Bilder als eine Matrix auf, die wir pixelweise miteinander vergleichen können, um deren Ähnlichkeit zu bestimmen. Dies können wir einfach über die Distanzfunktion d zwischen zwei Bildern A und B implementieren. Zum Beispiel die L1-Distanz wie in Abbildung 4 zu sehen.

Implementieren Sie folgende Funktionalitäten/Aufgaben:

test image				training image				pixel-wise absolute value differences			
56	32	10	18	10	20	24	17	46	12	14	1
90	23	128	133	8	10	89	100	82	13	39	33
24	26	178	200	12	16	178	170	12	10	0	30
2	0	255	220	4	32	233	112	2	32	22	108

- = → 456

Abbildung 4: L1-Distanz zwischen einem Trainingsbild und einem Testbild.

- a) Laden Sie die Trainings-Bilder und geben Sie ihnen die passenden Labels ("car", "face", "flower").
- b) Implementieren Sie die Funktion *knn*.
- c) Testen Sie Ihre Funktion mit den drei Test-Bildern. Evaluieren Sie die Genauigkeit ihrer Klassifikators indem Sie k verändern. Beantworten Sie in einer separaten Textdatei wie die Klassifikation verbessert werden kann. Geben Sie mindestens 3 Verbesserungsvorschläge an (diese können sich auch bedingen).

Abgabe Die Bearbeitungszeit der Teilaufgabe ist für ca. zwei Woche ausgelegt. Die Abgabe soll via Moodle bis zu dem dort angegebenen Termin erfolgen. Verspätete Abgaben werden mit einem Abschlag von 3 Punkten je angefangener Woche Verspätung belegt. Geben Sie bitte jeweils nur eine einzige .zip-Datei mit den Quellen Ihrer Lösung ab.

Ich werde mich stichprobenartig Ihre Lösungen in der Übung demonstrieren und erläutern lassen. Die Qualität Ihrer Demonstration ist, neben dem abgegebenen Code, ausschlaggebend für die Bewertung! Ich behalte mir vor, einzelne hochgeladene Aufgaben zusätzlich anzuschauen.