Adaptive Systeme
Aufgabe III

Prof. Dr. Nikolaus Wulff

06. Dezember 2019

1 K-Means Implementierung

Implementieren Sie den in der Vorlesung behandelten Algorithmus zur Vektorquantisierung, der auch unter dem Namen K-Means bekannt ist, für Vektoren eines n-dimensionalen Raums $\mathbb{R}^n$.

Bei vorgegebener Größe k des Codebooks wird mit einem initialen Codebook $\mathcal{C}^0$ mit zufällig generierten Vektoren $\vec{w}_j^{(0)}$ $j = 1, \ldots, k$ gestartet. In einem iterativen Prozess werden neue Codebücher $\mathcal{C}^{(\nu+1)}$ durch Klassifikation der Daten anhand des aktuellen Codebuch $\mathcal{C}^{(\nu)}$ und erneuter Schwerpunktsberechnung bestimmt. Das finale optimale Codebook $\mathcal{C} = \{\vec{w}_1, \cdots, \vec{w}_k\}$ ist gefunden, sobald sich die Schwerpunkte $\vec{w}_j^{(\nu)}$ nicht mehr verändern (oder als Spezialfall falls es zu oszillierendem Verhalten kommt!).

Test der Implementierung

Um Ihre Implementierung zu testen nehmen Sie als Daten $m = 90$ Zufallsvektoren $\vec{x}_j \in \mathbb{R}^2$ $j = 1, \ldots, m$, die Sie nach folgender Vorschrift erzeugen:

1. Ein Drittel der Vektoren werden zufällig in einer 1×1 großen Box mit Zentrum $\vec{z}_B = (1, 1)$ generiert.

2. Ein Drittel der Vektoren werden zufällig in einem Bereich mit dem Zentrum $\vec{z}_G = (-1, 1)$ nach einer $2D$-Gaussverteilung mit Varianz $\vec{\sigma}_G = (0.5, 0.25)$ erzeugt.

3. Das letzte Drittel der Vektoren wird zufällig in einem kreisförmigen Bereich mit dem Zentrum $\vec{z}_C = (0.5, -0.5)$ mit Radius $r_C = 0.5$ erzeugt.

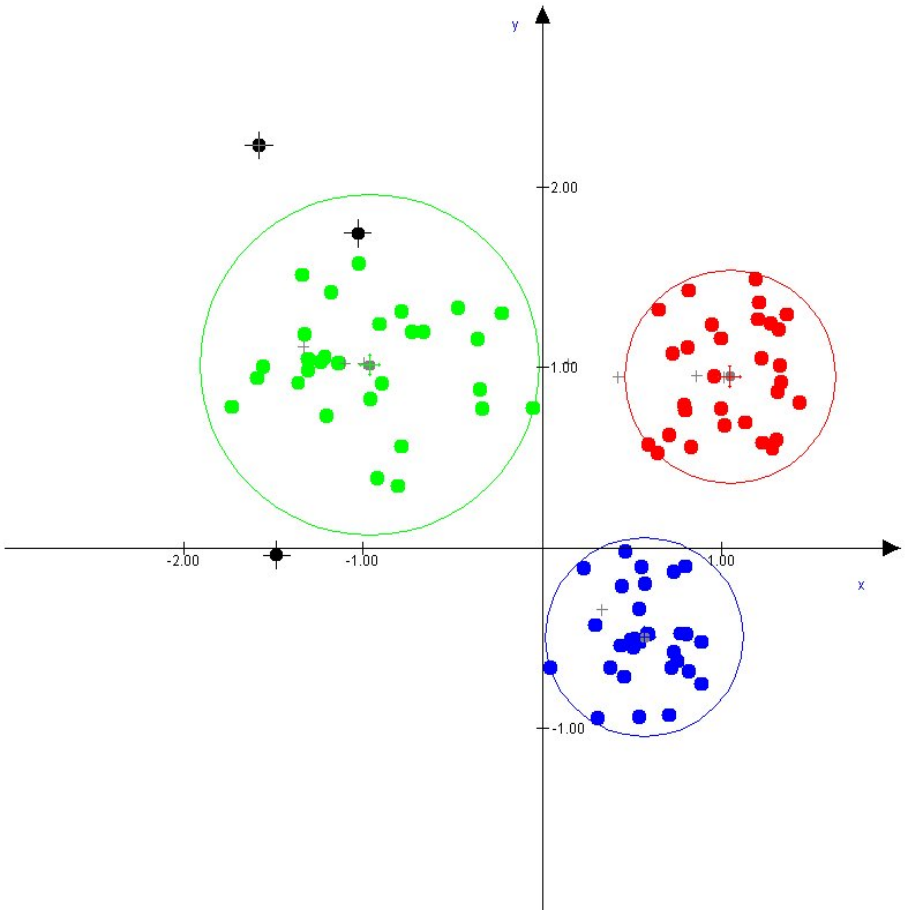

Abbildung 1: Visualisierung der Vektoren der drei Klassen B, G und C, der drei initialen und der gefundenen Codebuchvektoren mit ihrem jeweiligen *Einzugsbereichen* in der L_2-Norm.

Hinweis

Da die Daten- und Codebuchvektoren in einem beschränkten Bereich $U \subset \mathbb{R}^2$ liegen, lassen sich diese gut und anschaulich visualisieren, wie in Abbildung (1). Dies hängt natürlich von der verwendeten Programmiersprache und -umgebung und deren Möglichkeiten ab.

Die Vektoren zu den Klassen Kreis (C) und Box(B) sind sauber separiert, lediglich die Vektoren der asymmetrische Gaussverteilung (G) streuen weiter und können zufällig falsch klassifiziert werden, wobei durch die Wahl von $\vec{\sigma}_G$ immer noch eine gute Separation möglich sein sollte. Versuchen Sie für eigene Experimente auch diesen Parameter zu verändern, um die Anzahl an falschen Klassifikationen in Abhängigkeit davon ermitteln zu können.

Um Ihre Implementierung gegebenfalls zu debuggen, macht es Sinn die Anzahl an Datensätze auf z.B. $m = 6$ oder 9 zu reduzieren.

Aufgabe

1. Geben Sie die zeitliche Entwickung des Codebuchs $\mathcal{C}^{(\nu)}$, d.h. der drei Vektoren $\vec{w}_1^{(\nu)}, \vec{w}_2^{(\nu)}$ und $\vec{w}_3^{(\nu)}$ nach jeder Iteration ν aus.

2. Geben Sie die Zuordnung der Vektoren $\vec{x_j}$ zu den Klassen G, B und C nach jeder Iteration an und bestimmen Sie die Anzahl an Falschklassifizierungen.

3. Gegen welchen Grenzwert konvergieren die Codebuchvektoren? Stimmt dies mit Ihrer Erwartung überein?