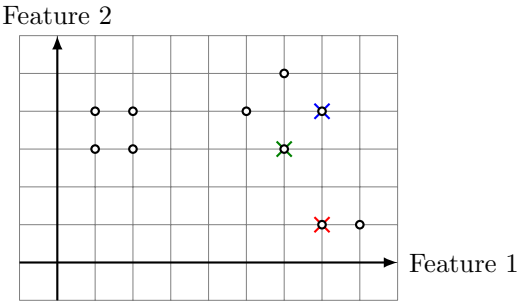
k-means Clustering

Clustering wird angewendet, wenn vermutet wird, dass sich die Instanzen auf natürliche Weise in Gruppen zerlegen lassen. Diese Cluster sollen Mechanismen sichtbar machen, die dafür sorgen, dass bestimmte Instanzen näher beieinander liegen als andere.

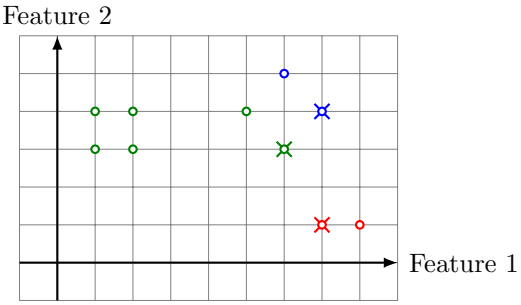
Der unten an einem Beispiel erklärte Algorithmus (Lloyd, 1982) zerlegt die Instanzen im $\mathbb{R}^2$ in k nicht überlappende Cluster.

Initialisierung



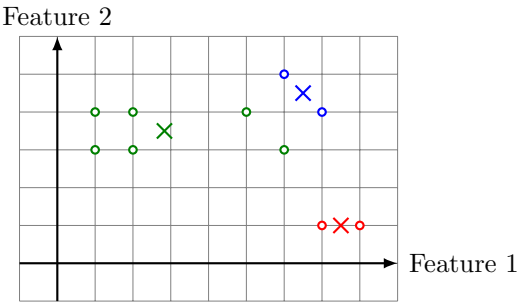
Zuerst muss eine Wahl für die Anzahl k der Cluster getroffen werden. Dieser Parameter gibt dem Verfahren seinen Namen. Als Clusterzentren (*Centroide*) werden $k = 3$ zufällige Datenpunkte gewählt.

1.1 Zuweisen



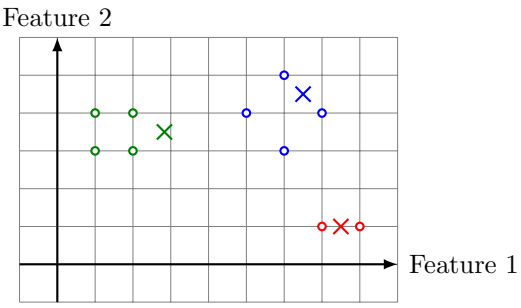
Jeder Punkt wird dem Cluster zugeordnet, zu dessen Centroid er den kürzesten Abstand hat.

1.2 Aktualisieren



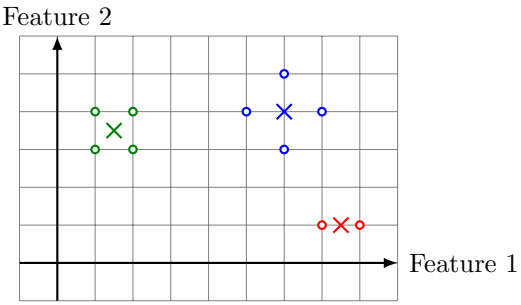
Der Schwerpunkt der zum gleichen Cluster gehörenden Punkte ergibt die neuen Centroide.

2.1 Zuweisen



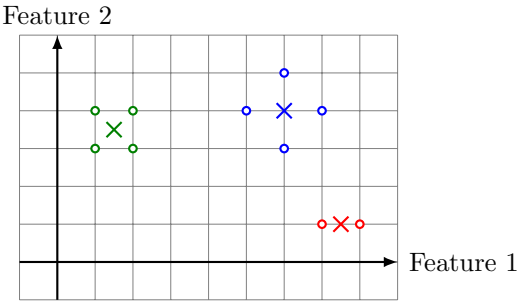
Jeder Punkt wird dem Cluster zugeordnet, zu dessen Centroid er den kürzesten Abstand hat.

2.2 Aktualisieren



Der Schwerpunkt der zum gleichen Cluster gehörenden Punkte ergibt die neuen Centroide.

Abbruch



Bleiben die Centroide stationär oder verändern sie sich nur wenig, endet das Verfahren.

Initialisierungsmethoden

In der Praxis sind unter anderem folgende Initialisierungsverfahren üblich:

- Wähle unter den Datenpunkten zufällig k Centroide aus.
- Ordne jeden Datenpunkt zufällig einem Cluster zu und fahre mit dem Aktualisierungsschritt fort.
- k-Means++ (Arthur und Vassilvitski, 2007)

Eine formalere Beschreibung

n Datenpunkte $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$

k Clusterzentren $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$

$$r_{ij} = \begin{cases} 1 & \text{falls Datenpunkt } \boldsymbol{x}_i \text{ zum Cluster } \boldsymbol{\mu}_j \text{ gehört} \\ 0 & \text{sonst} \end{cases}$$

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} ||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2 \tag{1}$$

Ziel: Bestimme die r_{ij} so, dass J (*distortion measure*) minimal wird

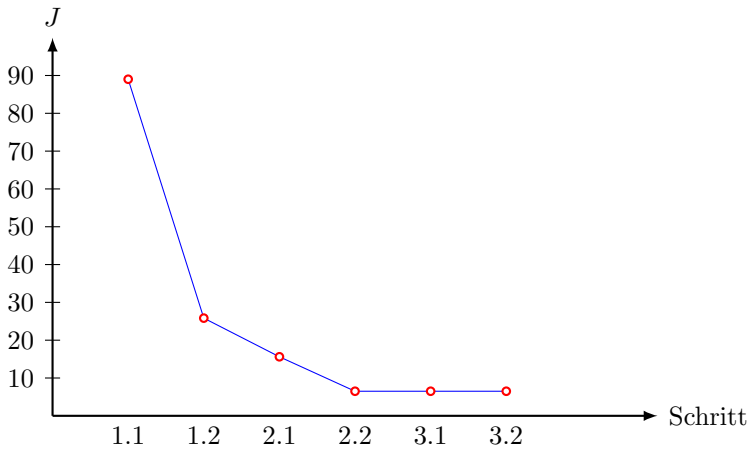
- Sind die Centroide $\boldsymbol{\mu}_j$ gegeben, genügt es, für jeden Datenpunkt $\boldsymbol{x}_i$ die innere Summe in (1) zu minimieren. Dies erreicht man wie folgt:

$$r_{ij} = \begin{cases} 1 & \text{für } \arg\min_j ||\boldsymbol{x}_i - \boldsymbol{\mu}_j||^2 \\ 0 & \text{sonst} \end{cases}$$

- Hält man dann die r_{ij} fest, so lässt sich (1) weiter minimieren, indem man jeweils nach $\boldsymbol{\mu}_j$ differenziert und nach $\boldsymbol{\mu}_j$ auflöst

$$-2 \sum_{i=1}^{n} r_{ij}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) = 0 \quad \Leftrightarrow \quad \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} r_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{n} r_{ij}}$$

Distortion measure



Bemerkungen

- Die Aufgabe ist NP-schwer. (k^n mögliche Clusterings)
- Der k-Means-Algorithmus konvergiert gegen ein *lokales Minimum* der Summe der quadrierten Abstände der Punkte von den zugehörigen Centroide. Daher ist es sinnvoll, das Verfahren mehrmals mit verschiedenen Anfangswerten durchzuführen und am Ende das beste Ergebnis zu wählen.

Anwendungen

- Marktsegmentierung
- Vektorquantisierung (Bildsegmentierung, Kompression)

Quellen

Lloyd, Stuart P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**(2), 129–137.

Arthur, D., Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proc. 18th ACM-SIAM symp. on Discrete algorithms*, pp. 1027–1035.