K-Means Algorithmus

Nadja Seeberg, Sinem Demiraslan

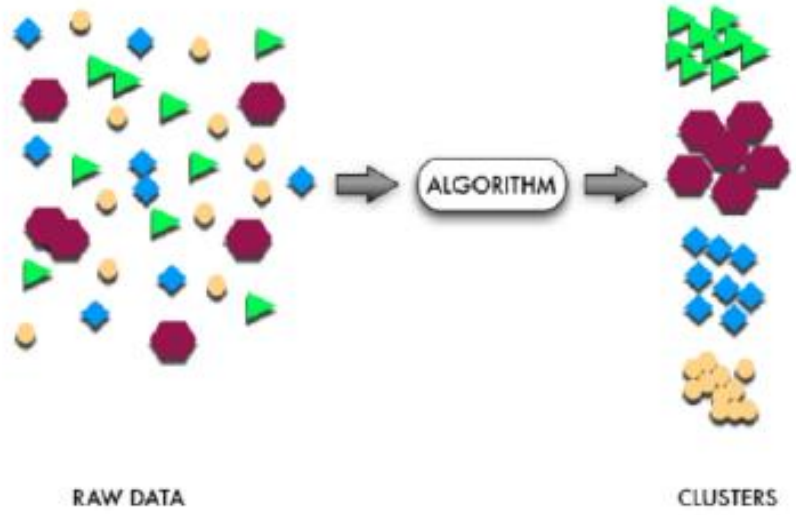
02. Februar 2021

Seminar: Vertiefung der Grundlagen der Computerlinguistik

Leonie Weißweiler

CIS, LMU München

Theoretischer Hintergrund



(Quelle: https://sp1819.github.io/Unsupervised_vs_Supervised_Learning.pdf)

Allgemeine Fakten

→ **Clustering** Algorithmus

→ Unüberwachtes Lernen

Motivation:

Gegebene Datenmenge $X = \{x_1, x_2, \ldots, x_n\}$ in eine Menge von k disjunkten Clustern $C_1, \ldots, C_k$ einteilen,

sodass das **Clustering-Kriterium** optimiert wird[1]

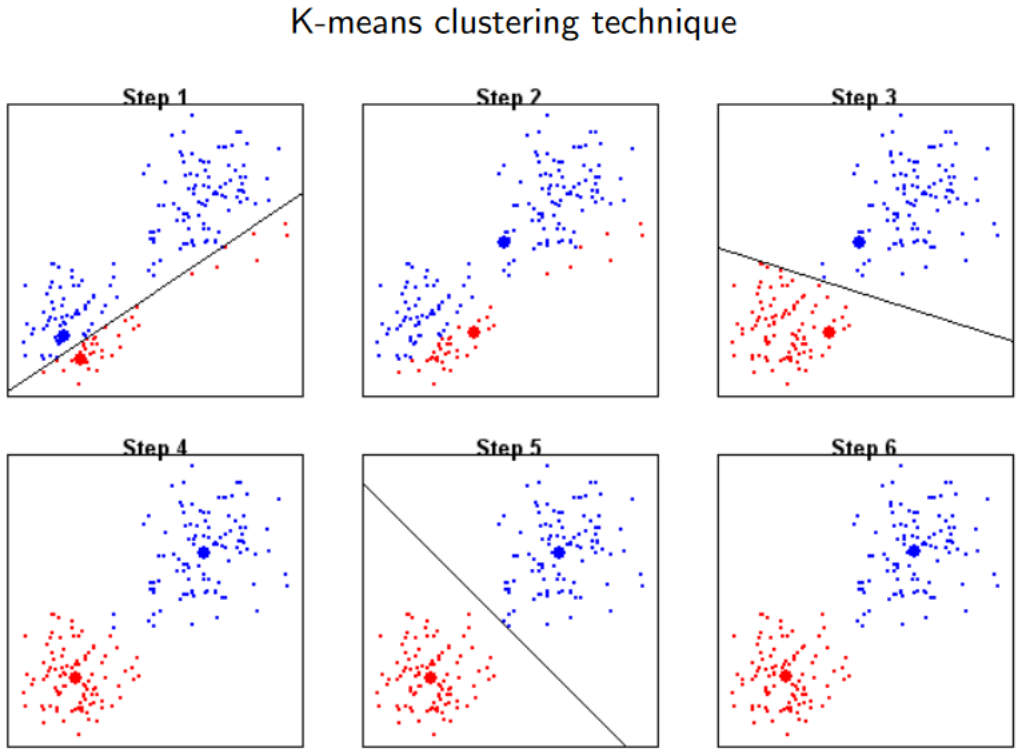
Natürlichsprachlich ausgedrückt:

Bestehende Datenpunkte zu einem ***Cluster* aus zueinander ähnlichen Daten** gruppieren → Suche nach Muster in den Daten

Neu dazukommende Datenpunkte einem *Cluster* **zuweisen** („ähnlich“ = minimale **euklidische Distanz**)

1 Vgl. Aristidis et al., 2003.

Vorgehensweise



Schritt 1

❶ Zufällig *k* Cluster *centroids* **initialisieren**

❷ Datenpunkte dem nächsten Cluster *centroid* **zuweisen**

→ **Euklidische Distanz**: $$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Schritt 2

Update Cluster *centroids* (Erneute Berechnung der *centroids*)

Schritt 3

Wiederhole ab ❷,

bis Konvergenz oder Anzahl max. Iterationen erreicht

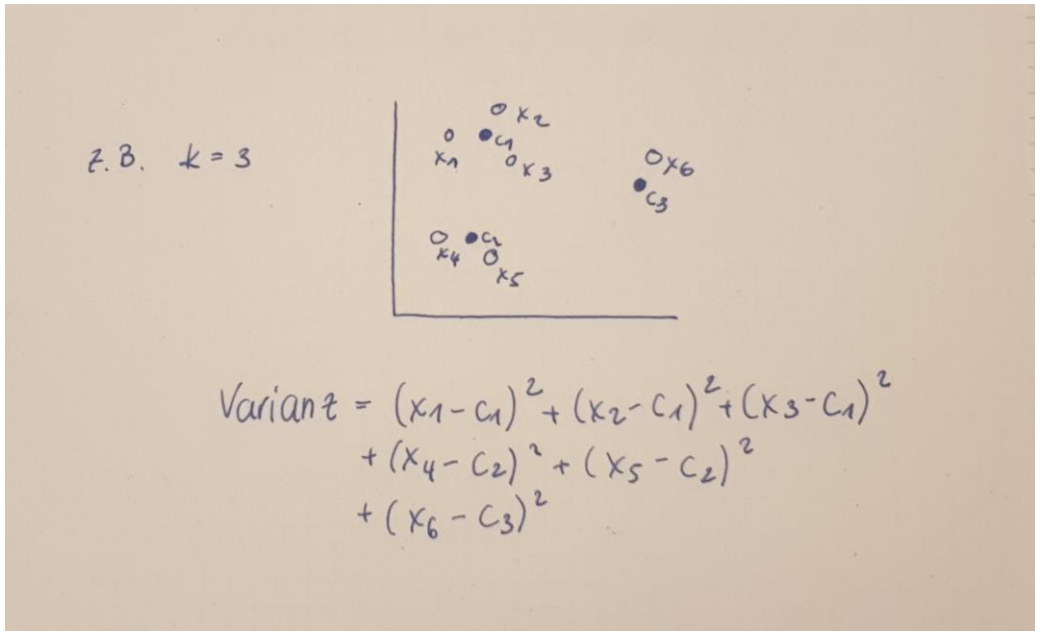
(Quelle: https://sp1819.github.io/Unsupervised_vs_Supervised_Learning.pdf)

Welches Kriterium wird optimiert?

→ Varianz innerhalb eines Clusters minimieren

$$J_{\mathrm{K}} = \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \mathbf{m}_k)^2$$

Beispiel:



Wobei

$\{x_1, \ldots, x_n\} = X$

$m_k = \sum_{i \in C_k} x_i / n_k$

Centroid von Cluster C_k, mit n_k Anzahl der Elemente in C_k

Formeln aus Ding und He, 2004.

Variationen des K-Means Clustering (Beispiele)

***K-Median Clustering*[1]**

→ *k* Cluster so finden, dass **Summe der Distanzen**

zum nächsten **Median** am kleinsten ist

(vgl. *K-Means*:

Summe der quadrierten Distanzen minimieren)

***K-Means++ Clustering*[2]**

Statt rein zufälliger Initialisierung:

centroids **gleichmäßig** verteilen

Für jeden *centroid* C_k gibt es genau ein $x \epsilon X$, sodass gilt $\boldsymbol{C_k = x}$

***Hierarchical Clustering*[3]**

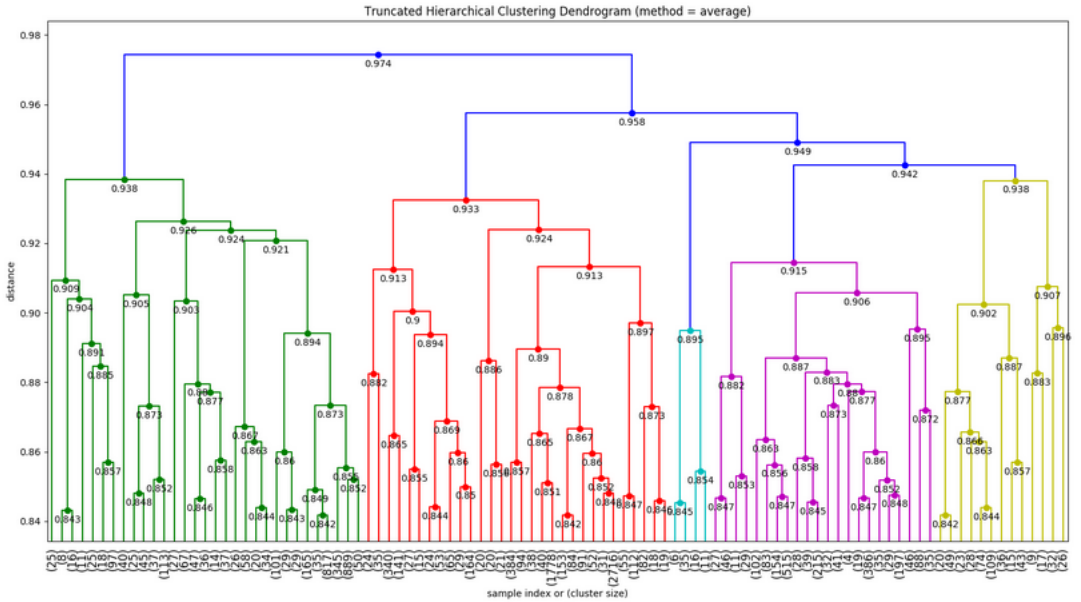
→ *Agglomerative* vs. *Divisive*

Ähnlichkeit zwischen Clustern berechnen

Ähnlichsten **Cluster zusammenführen**

Wiederhole, bis jedes Cluster behandelt wurde

→ **Dendrogramm** mit hierarchischen Strukturen



1 Vgl. Arora et al., 1998.
2 Arthur und Vassilvitskii, 2006.
3 Vgl. Abbas, 2008.

Anwendung von K-Means: Ziele und Beispiele

✳ **Ziel**: (Text-)Daten in Cluster aufteilen, um zueinander ähnliche Datenpunkte (Texte) zu gruppieren

- Erhalt erster Einblicke in Datenmenge
- Klassifizierung von Daten

- Beispiele für Textklassifizierungen:
  - Nationalhymnen[1]
    - Gruppierung nach Leitmotiv (Religion, Militär,…)
  - Forenbeiträge (z.B. 20 newsgroups dataset)
    - Klassifizierung von Themengebieten/ Rubriken (Space, Computer Graphics,…)

1: Idee: https://medium.com/@lucasdesa/text-clustering-with-k-means-a039d84a941b

Anwendung von K-Means: Preprocessing

- 20 newsgroups dataset
  - 3387 Texte
  - 4 Kategorien (“atheism”, “religion”, “computer graphics”, “space”)

```
vectorizer = TfidfVectorizer(max_df=0.5, stop_words="english", use_idf=True)
X = vectorizer.fit_transform(texts)
```

- Maximum Document Frequency
- Entfernung der Stoppwörter (sprachspezifisch)
- Wortvektoren
  - Bag-of-Words Repräsentation
  - TF-IDF Gewichtung

Anwendung von K-Means: Text Clustering mit Scikit-learn

```
km = KMeans(init="k-means++", n_clusters=4, n_init=8)
indices = km.fit_predict(X)
```

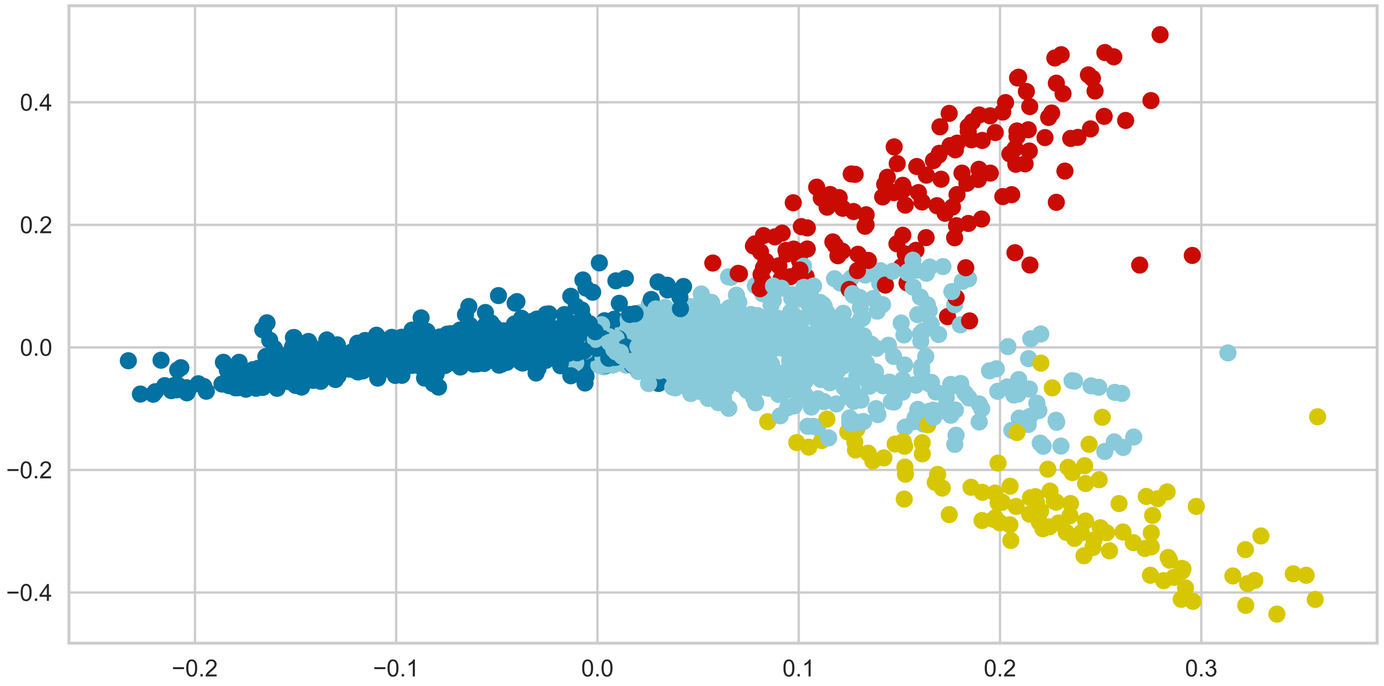


Abbildung 1: Cluster für Beiträge aus dem 20 newsgroups dataset

Anwendung von K-Means: *k* ermitteln

- Geeignete Werte für *k* ermitteln:
  - Ellenbogen-Methode
    - K-means für verschiedene Werte für *k* berechnen
    - Für jedes *k* den total within-cluster sum of square (WSS) berechnen
    - Ergebnis als Graph abbilden und Wert bei Knick wählen

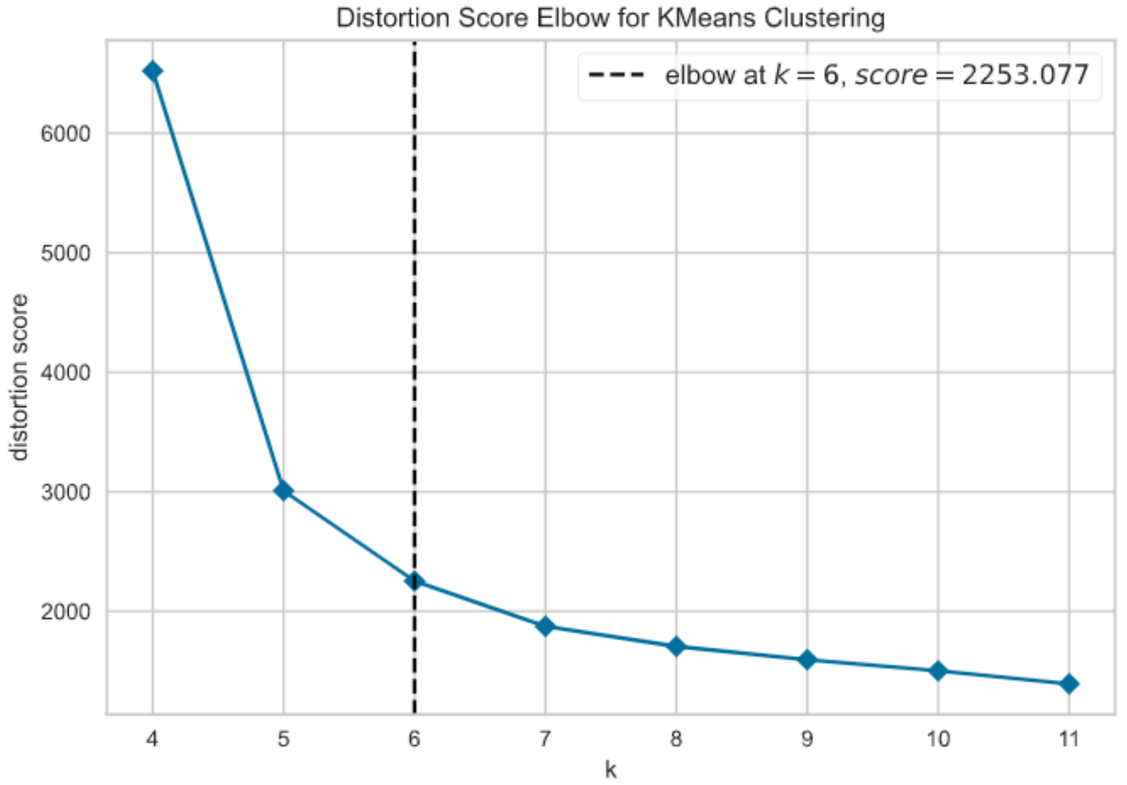


Abbildung 2: Beispielhafter Graph für die Auswahl eines geeigneten Wertes für *k*

Anwendung von K-Means: Vor- und Nachteile

\+ Benötigt kein Labelling

\+ Vielseitig aufgrund modifizierbarer Parameter

\+ Simplifizierte, übersichtliche Darstellung von Daten

! Texte aus überlappenden Themenbereichen ggf. nicht eindeutig trennbar

  ! Kategorie “alt.atheism” vs. “talk.religion.misc”

! Konvergenz stark abhängig von der Wahl der initialen Zentroide

! Großer Einfluss von Ausreißern in Datenpunkten

➡**Fazit**: Primäre Anwendung in **explorativer Datenanalyse**, ggf. nicht präzise genug für exakte Klassifizierungen

Referenzen I

1. Abbas, Osama Abu. (2008). "Comparisons between data clustering algorithms." International Arab Journal of Information Technology (IAJIT) 5.3

2. Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. 2003. "The global k-means clustering algorithm." *Pattern recognition* 36.2 451-461.

3. Arora, Sanjeev, Prabhakar Raghavan, and Satish Rao. 1998. "Approximation schemes for Euclidean k-medians and related problems." *Proceedings of the thirtieth annual ACM symposium on Theory of computing*.

4. Arthur, David, and Sergei Vassilvitskii. 2006. k-means++: The advantages of careful seeding. Stanford.

5. Ding, Chris, and Xiaofeng He. 2004. "K-means clustering via principal component analysis." *Proceedings of the twenty-first international conference on Machine learning*.

Referenzen II

6. Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert systems with applications*, *40*(1), 200-210.

7. Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal*, *1*(6), 90-95.

8. De Sá, L., (2019, 18. Dezember). *Text Clustering with K-Means. Clustering national anthems with unsupervised learning.* Medium. https://medium.com/@lucasdesa/text-clustering-with-k-means-a039d84a941b

9. Scikit-learn Dokumentation zu K-Means. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

10. Scikit-learn Dokumentation zu Tf-Idf Vectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html?highlight=tf%20idf#sklearn.feature_extraction.text.TfidfVectorizer

11. Yellowbrick Dokumentation zu Elbow-Method. https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

Danke für eure Aufmerksamkeit!