

# ClusterLabor: Ein Werkzeug zur interaktiven Visualisierung und Analyse von Clusteralgorithmen

Andres, D., Joachim, S. und Hennecke, M.

DOI: 10.18420/ibis-02-02-09

## Zusammenfassung

In diesem Beitrag wird die Webanwendung ClusterLabor (verfügbar unter <https://www.ddi.informatik.uni-wuerzburg.de/cluster>) vorgestellt. ClusterLabor ermöglicht eine interaktive Visualisierung und Analyse von Clusteralgorithmen in zweidimensionalen Datensätzen. Damit können verschiedene Algorithmen hinsichtlich ihrer Ergebnisse in Abhängigkeit von der gewünschten Anzahl an Clustern verglichen werden. Anwender können aus verschiedenen Beispieldatensätzen wählen, eigene Datensätze hochladen oder Datensätze direkt durch manuelle Eingabe generieren. Zum Clustern stehen verschiedene Methoden zur Verfügung: der *k-Means-Algorithmus* mit Varianten wie Lloyd oder MacQueen, der *k-Medoids-Algorithmus* sowie *hierarchische Clusteranalyse* mit unterschiedlichen Distanzmaßen und Fusionierungsalgorithmen. Ein besonderer Fokus liegt dabei auf dem Unsupervised Learning, einem Bereich der Künstlichen Intelligenz (KI), bei dem Algorithmen Muster und Strukturen in unbeschrifteten Daten selbstständig erkennen. Zur Bestimmung der "optimalen" Clusterzahl  $k$  sind zudem Visualisierungen des Elbow Plots (Ellenbogen-diagramms), des Average Silhouette Plots (ASW-Kurve) sowie des Dendrogramms integriert.

## Einleitung

Politik und Wirtschaft schreiben Datenanalysetechniken ein großes Innovationspotential zu. Diesen Techniken erkennen in immer größer werdenden Datenbeständen Strukturen und Muster und gewinnen so neue Informationen. Gleichzeitig stehen vielseitige Bedenken im gesellschaftlichen Diskurs: Seien es die Gefahren für den Datenschutz oder die für Fachfremde oft intransparenten Abläufe und Algorithmen (BT 20/5149).

Zur Versachlichung der Diskussion besteht die Möglichkeit fachliche Kompetenzen zur Funktionsweise von und zum Umgang mit KI-Systemen bereits in der Schule zu vermitteln. Einige Lehrpläne haben die Thematik bereits aufgegriffen und integrieren z. B. bekannte Clusteralgorithmen aus dem Bereich der Datenanalyse in ihre Curricula. Clusteralgorithmen sind Methoden des maschinellen Lernens, die dazu

dienen, ähnliche Datenpunkte in Gruppen oder Clustern einzuteilen. Beispielsweise kann durch diese Algorithmen aus einem Datensatz der Bildschirmgröße (Breite und Länge) von Geräten, die auf eine Website zugreifen, eine Aufteilung in verschiedene Cluster ermittelt werden. Davon ausgehend können die entstandenen Cluster als Gruppen von Gerätetypen (z. B. "Handy", "Tablet" oder "Laptop") interpretiert werden. Ein Beispiel für einen solchen Lehrplan, ist der bayerische LehrplanPLUS (ISB o. J.), der in Jahrgangsstufe 13 die Implementierung des *k-Means-Algorithmus* und die Analyse der entstehenden Cluster in Abhängigkeit von der Clusterzahl  $k$  fordert. Mithilfe von ClusterLabor können einerseits viele dieser Anforderungen im Unterricht umgesetzt und andererseits die Thematik der Clusteralgorithmen didaktisch reduziert vertieft werden.

## Verwandte Arbeiten

Für die Visualisierung und die Implementierung des *k-Means-Algorithmus* gibt es bereits mehrere Werkzeuge. Neben diversen Möglichkeiten im Web, die einzelnen Schritte des *k-Means-Algorithmus* zu visualisieren, existiert z. B. mit Andres (2024) eine Möglichkeit, den *k-Means-Algorithmus* unplugged einzuführen und anschließend mit einer Vorlage in der Programmierumgebung BlueJ umzusetzen. Eine Möglichkeit, die entstandenen Cluster in Abhängigkeit von  $k$  zu analysieren, besteht jedoch nicht.

Eine Analyse der Qualität von verschiedenen Clusterungen bieten unterschiedliche bereits existierende Werkzeuge zum Data Mining:

Orange (Demšar 2013) ist ein Open-Source-Werkzeug für Datenvisualisierung, maschinelles Lernen und Data Mining. Neben vielen anderen Funktionen ist auch das Clustern von Daten mittels des *k-Means-Algorithmus* oder des hierarchischen Clusterings möglich. Gleichzeitig stehen verschiedene Möglichkeiten zur Visualisierung der entstandenen Cluster zur Verfügung. Orange stellt zudem viele weitere Funktionen im Bereich Data Mining bereit. Aufgrund des großen Funktionsumfangs ist es nicht didaktisch auf die Visualisierung des *k-Means-Algorithmus* oder verwandter Algorithmen reduziert. Schülerinnen und Schüler müssen erst in

das Programm eingewiesen und durch die einzelnen Arbeitsschritte und Möglichkeiten geführt werden. Eine mögliche Verwendung von Orange im Zusammenhang mit Clusteralgorithmen im Unterricht der neunten Jahrgangsstufe zeigt Pöhner (2023). In der vorgestellten Unterrichtssequenz werden mittels des Clusteranalysetools in Orange3 Filterblasen in sozialen Medien thematisiert.

Auch Bibliotheken aus R oder Python können verwendet werden, um verschiedene Clusterungen von unterschiedlichen Algorithmen und Methoden darzustellen und miteinander zu vergleichen. Diese Bibliotheken erfordern jedoch ein gewisses Maß an Einarbeitung und zusätzlich ein Grundverständnis der zugrundeliegenden Sprache, was im Schulbetrieb, sollte zuvor noch nicht mit R oder Python gearbeitet worden sein, einen erheblichen Aufwand darstellt.

Neben Abo- und kostenpflichtigen Versionen von professionellen Werkzeugen zur Datenanalyse gibt es auch kostenlose Werkzeuge wie Weka (Holmes 1994), welche mächtige Werkzeuge zur allgemeinen Datenanalyse sind. Diese gehen jedoch weit über den schulischen Aspekt des Clusters von Daten mittels einfacher Algorithmen hinaus und sind nicht für die didaktische Arbeit reduziert.

## Anforderungen an das Programm

Ziel von ClusterLabor ist es, Schülerinnen und Schülern ein Programm zur Verfügung zu stellen, mit dem sie ohne große Einarbeitung Clusteralgorithmen ausführen und sich die Ergebnisse visuell anzeigen lassen können. Insbesondere sollen die Schülerinnen und Schüler einfach mit der Anzahl  $k$  der zu bildenden Cluster, dem zentralen Parameter aller Cluster-Verfahren, experimentieren können. Bezogen auf den  $k$ -Means-Algorithmus entspricht dies z. B. den Anforderungen der Jahrgangsstufe 13 des bayerischen Lehrplans:

*"Die Schülerinnen und Schüler [...] analysieren für verschiedene Eingabedaten die Ergebnisse, die der  $k$ -Means-Algorithmus in Abhängigkeit von  $k$  liefert." (ISB o.J.)*

Ziel dieses Projektes war es, ein Programm zu entwickeln, mit dem dieser Lehrplanpunkt in der Unterrichtspraxis umgesetzt werden kann.

Daraus folgen direkt drei Anforderungen an ein solches Programm: Es müssen verschiedene Datensätze in das Programm geladen werden, das Programm muss in der Lage sein, diese Daten mithilfe eines Clusteralgorithmus zu clustern, und die entstandenen Cluster in Abhän-

gigkeit von  $k$  anzuzeigen. Es soll möglich sein zwischen verschiedenen Clusterzahlen  $k$  zu wechseln und so die Ergebnisse des Algorithmus zu vergleichen. Das Ziel dieses Vergleichs ist es, die Anzahl der Cluster  $k$  zu ermitteln, mit der die Aufteilung als "optimal" angesehen werden kann. Je nach Lehrplan kann es möglich sein, dass diese Analyse über die Betrachtung mit bloßem Auge hinausgehen soll und die Qualität einer Clusterung systematisch eingeschätzt und beurteilt werden muss. Dafür stehen in der Fachwissenschaft verschiedene Metriken und grafische Diagramme zur Verfügung, welche somit auch in ClusterLabor integriert werden sollen.

Der bayerische Lehrplan fordert beispielsweise nicht, dass der  $k$ -Means-Algorithmus über die Variation der Clusterzahl hinaus verändert werden soll. Dennoch ist eine Anpassung des  $k$ -Means-Algorithmus ein naheliegender Gedankengang. Schülerinnen und Schüler stoßen beim Experimentieren mit dem implementierten Algorithmus schnell auf die Einschränkung des  $k$ -Means-Algorithmus, konvexe und gleich große Cluster zu bevorzugen, was zu nicht sinnvollen Clusterergebnissen bei konkaven Datensätzen wie z. B. "zweiMonde" führen kann. Zum tieferen Verständnis des  $k$ -Means-Algorithmus sollen in dem Programm folglich Anpassungen der Metrik, der Art der Zentrumsbildung, der Wahl der Startzentren sowie der Zeitpunkt der Aktualisierung des Zentrums (Lloyd vs. MacQueen) möglich sein. Dadurch können die Schülerinnen und Schüler die Auswirkungen dieser Parameter auf den Algorithmus betrachten, ohne dass sie diese oder eine passende Visualisierung selbst programmieren müssen.

Ein differenziertes Bild, welches verschiedene Clusteralgorithmen, deren Stärken und Schwächen und damit folgend ihre unterschiedlichen Einsatzgebiete aufzeigt, kann im Unterricht nicht ohne weiteres entstehen. Deshalb ist eine weitere Forderung an das Programm, dass es verschiedene Clusteralgorithmen zur Verfügung stellt und die Ergebnisse unterschiedlicher Algorithmen miteinander vergleichbar macht. Somit kann den Schülerinnen und Schülern die Fülle an unterschiedlichen Methoden und Algorithmen in der Clusteranalyse aufgezeigt werden, ohne zuvor im Unterricht tiefgehende fachliche Grundlagen geschaffen zu haben. Der vom Lehrplan vorgesehene Clusteralgorithmus kann am Ende einer solchen Sequenz mit seinen Stärken und Schwächen als einer von vielen Algorithmen des Data Mining eingeordnet werden.

## Fachlicher Hintergrund

### Der $k$ -Means- und der $k$ -Medoids-Algorithmus

Beim  $k$ -Means-Algorithmus werden zunächst  $k$  Clusterzentren auf  $k$  zufällig gewählten Punkten des Datensatzes initialisiert. Danach werden alle Datenpunkte dem Zentrum zugeordnet, dem sie am nächsten sind. Anschließend erfolgt die Neuberechnung der Clusterzentren über Mittelwertbildung. Die so berechnete Zentrumposition muss insbesondere mit keinem echten Datenpunkt des Datensatzes übereinstimmen. Diese beiden Schritte werden so lange wiederholt, bis sich die Position der Clusterzentren nicht mehr ändert (vgl. Ertel 2021). Die linke Spalte von Abb. 1 zeigt den schematischen Ablauf des  $k$ -Means-Algorithmus. Der  $k$ -Means-Algorithmus hat eine gewisse Ähnlichkeit mit dem Konzept des Schwerpunkts in der Physik, da beide Methoden darauf abzielen, einen zentralen Punkt zu bestimmen, der die "Mitte" eines Systems repräsentiert: Der  $k$ -Means-Algorithmus berechnet das Clusterzentrum eines Clusters als den Mittelwert aller Punkte im Cluster, ähnlich wie der Schwerpunkt in der Physik. Der Schwerpunkt eines Körpers ist der Punkt, an dem die gesamte Masse des Körpers so betrachtet werden kann, als ob sie dort konzentriert wäre. Er ist der Durchschnitt der Positionen der Massenpunkte, gewichtet nach ihrer Masse.

Ein wesentlicher Aspekt des  $k$ -Means-Algorithmus ist die Wahl der Startzentren, auch Initialisierung genannt. Diese können erheblichen Einfluss auf die endgültige Clusterbildung haben. Verschiedene Startzentren können zu unterschiedlichen Clustern führen. Dies verdeutlicht Abb. 2, die zeigt wie unterschiedliche Ergebnisse des  $k$ -Means-Algorithmus in Folge unterschiedlicher Initialisierungen entstehen. Der Datensatz kann per Augenmaß leicht in drei Cluster unterteilt werden, welche aber nur in einem der drei Fälle ermittelt werden. Das Beispiel zeigt, wie stark die Wahl der Startzentren das Endergebnis des Algorithmus beeinflusst.

Der  $k$ -Means-Algorithmus reagiert zudem empfindlich auf Ausreißer, da der gebildete Mittelwert für die neuen Koordinaten eines Zentrums leicht durch Ausnahmen beeinflusst wird.

Der  $k$ -Medoids-Algorithmus ist eine Variante des  $k$ -Means-Algorithmus, der robuster gegenüber Rauschen und Ausreißern ist. Anstatt eines Mittelwertes wird ein sogenannter Medoid gebildet. Ein Medoid ist der am zentralsten gelegene Datenpunkt eines Clusters, d. h. der Datenpunkt mit der geringsten Summe der Abstände zu allen anderen Punkten des Clusters. Man kann sich den Medoid eines Clusters als den "repräsentativsten" Punkt im Cluster vorstellen, ähnlich dem Median in der Statistik, der den "zentralsten" Wert einer sortierten Liste darstellt. Der Medoid bleibt robust gegenüber Ausreißern, da er ein tatsächlicher Datenpunkt

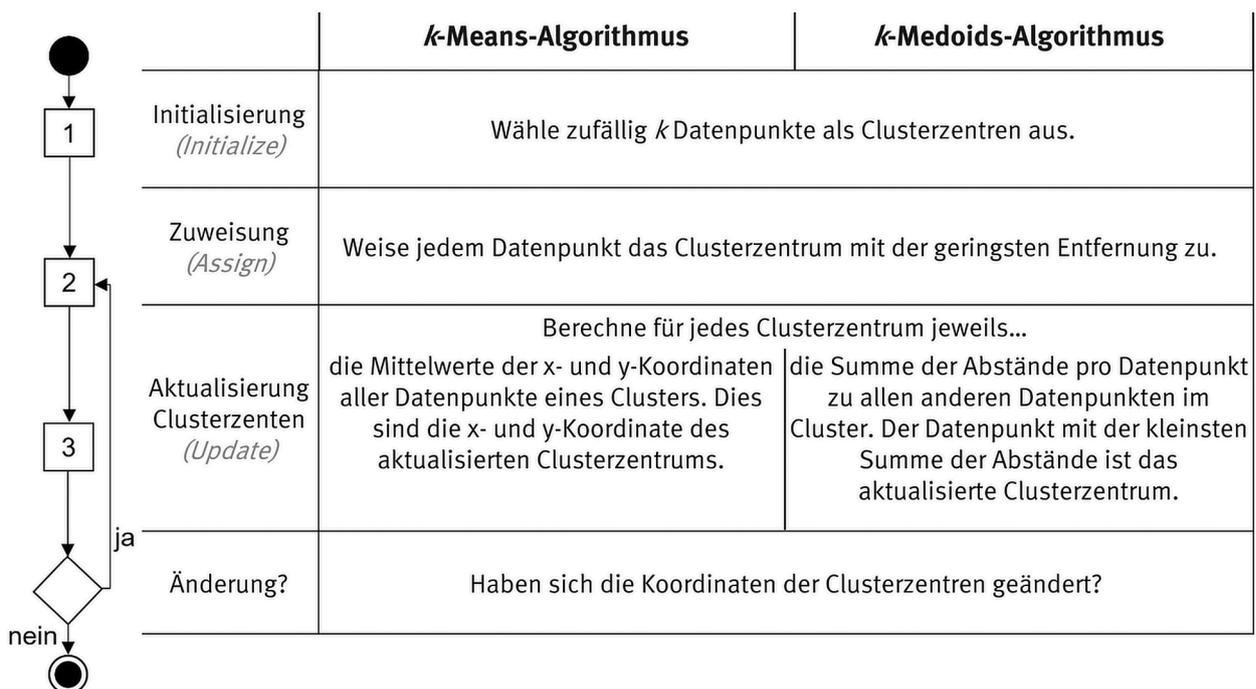


Abbildung 1: Schematischer Ablauf des  $k$ -Means- und des  $k$ -Medoids-Algorithmus

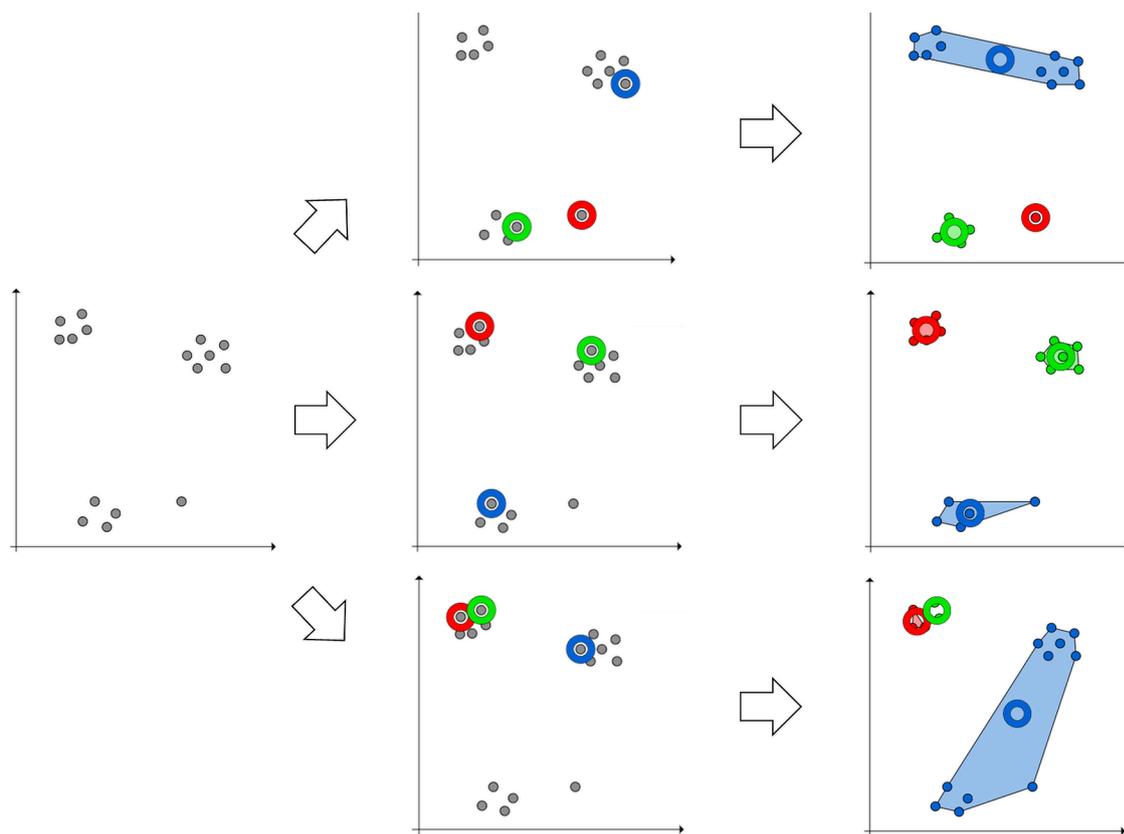


Abbildung 2: Einfluss der Wahl der Startzentren auf die durch den  $k$ -Means-Algorithmus entstehenden Cluster.  
 Erste Zeile: Der Ausreißer-Datenpunkt wurde als Startzentrum gewählt. Aufgrund der Verteilung der anderen Startzentren ist das rote Zentrum jedoch für keinen anderen Datenpunkt das Zentrum mit der geringsten Distanz und wird demnach nie aktualisiert.  
 Zweite Zeile: Günstige Verteilung der Startzentren führt zu dem erwarteten Cluster-Ergebnis.  
 Dritte Zeile: Die oberen Startzentren wurden zu nah aneinander gewählt und teilen sich somit das obere Cluster. Visualisierungen mit ClusterLabor

ist, der die kleinsten summierten Distanzen zu anderen Punkten im Cluster hat. Abb. 1 zeigt den schematischen Ablauf des  $k$ -Medoids-Algorithmus in Gegenüberstellung zum  $k$ -Means-Algorithmus.

Eine bekannte Umsetzung des  $k$ -Medoids-Algorithmus ist die Partitionierung um Medoide (Partitioning Around Medoids, PAM), welche auch für die Umsetzung des  $k$ -Medoids-Algorithmus in ClusterLab gewählt wurde (vgl. Jin 2010).

### Bestimmen der "optimalen" Clusterzahl $k$ mithilfe des $k$ -Means- oder $k$ -Medoids-Algorithmus

Bei realen Datensätzen ist die Anzahl  $k$  der Cluster oft nicht im Vorhinein bekannt. Oft gibt es in Abhängigkeit vom spezifischen Anwendungsfall und den Eigenschaften der Daten einen größeren Bereich von  $k$ -Werten, die sinnvoll sind. Deshalb stellt sich hier die Frage nach der "optimalen" Clusterzahl  $k$ . Die Wahl von  $k$  mag in einigen Fälle durch Darstellung der Punkte in einem Koordinatensystem mit dem

bloßen Auge möglich sein, bei höheren Dimensionen versagt diese Methode jedoch. Deshalb wurden verschiedene Heuristiken entwickelt, um ein möglichst gut passendes  $k$  bestimmen zu können.

Eine der ältesten und einfachsten Möglichkeiten wird oft in Verbindung mit dem  $k$ -Means-Algorithmus angewandt: Die *Ellbogenmethode* (Elbow Plot). Dafür wird der vorliegende Datensatz zunächst für verschiedene Clusterzahlen  $k$  geclustert. Das Ellbogendiagramm zeigt die WCSS-Werte (Within Cluster Sum of Square) der einzelnen Clusterungen auf der y-Achse und die dazugehörigen Werte von  $k$  auf der x-Achse. Ein heuristisch guter  $k$ -Wert ist der Punkt, an dem das Diagramm einen Ellenbogen, also eine Knickstelle, bildet. Dabei berechnet sich der WCSS-Wert eines Datensatzes  $X$  durch:

$$WCSS(X) = \sum_{x \in X} \|x - c(x)\|^2$$

wobei  $c(x)$  das Zentrum angibt, welchem der Datenpunkt  $x$  durch den  $k$ -Means-Algorithmus zugeordnet wurde (vgl. Schubert 2023). Die Ellbogenmethode ist in der Fachliteratur um-

stritten, da ihre Ergebnisse je nach Datensatz weder eindeutig noch korrekt sind (vgl. Schubert 2023). Aufgrund ihrer Einfachheit und guten Aussagekraft für kleine klar clusterbare zweidimensionale Datensätze ist sie dennoch in ClusterLabor integriert.

Eine andere Methode zur Clusteranalyse ist die *Average Silhouette Method*. Analog zur Ellenbogenmethode wird zunächst der Datensatz für verschiedene  $k$  geclustert. Für jedes Ergebnis wird nun der Durchschnitt der Silhouettenkoeffizienten aller Datenpunkte des Datensatzes berechnet. Der Silhouettenkoeffizient  $s(i)$  misst, wie gut ein Datenpunkt  $i$  zu seinem eigenen Cluster im Vergleich zu benachbarten Clustern passt. Es gilt:

$$s(i) = \begin{cases} 0, & \text{falls } a(i) = 0 \\ \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, & \text{sonst} \end{cases}$$

wobei  $a(i)$  der mittlere Abstand des Punktes  $i$  zu allen anderen Punkten desselben Clusters und  $b(i)$  der mittlere Abstand zu den Punkten des nächstgelegenen Clusters ist. Es gilt  $-1 \leq s(i) \leq 1$ , wobei negative Werte darauf hinweisen, dass der Punkt  $i$  dem falschen Cluster zugeteilt wurde, und positive Werte dafürsprechen, dass der Punkt  $i$  richtig eingeteilt wurde. Für den Vergleich verschiedener Clusterzahlen  $k$ , kann man die Durchschnittswerte aller Silhouettenkoeffizienten (Average Silhouette Width, ASW) auf der y-Achse und die zugehörige Clusterzahl  $k$  auf der x-Achse in einem Diagramm auftragen, ergibt sich ein Maximum an der Stelle des heuristisch zu wählenden  $k$  (vgl. Ertel 2021).

Aufgrund der Ähnlichkeit des  $k$ -Means- und  $k$ -Medoids-Algorithmus können die Konzepte der Ellenbogenmethode und Average Silhouette Method direkt auf den  $k$ -Medoids-Algorithmus übertragen werden.

## Das agglomerative hierarchische Clustering

Ein alternativer Algorithmus ist das agglomerative hierarchische Clustering. Beim agglomerativen hierarchischen Clustering fasst man zu Beginn jeden einzelnen Datenpunkt als ein eigenes Cluster auf. Danach werden die beiden Nachbarcluster mit der geringsten Distanz so lange zusammengeführt, bis alle Punkte in einem einzigen Cluster vereinigt sind (vgl. Ertel 2021). Um die Cluster mit minimalem Abstand zu bestimmen, gibt es bestimmte Metriken zum Messen der Distanz zwischen Clustern. Beim Single Linkage bestimmt sich die Distanz zwischen zwei Clustern A und B durch den minimalen Abstand zweier Punkte aus den Clustern, während beim Complete Linkage der maximale Abstand gewählt wird:

$$D_{\text{singleLinkage}}(A, B) = \max\{d(a, b) \mid a \in A, b \in B\}$$

$$D_{\text{completeLinkage}}(A, B) = \min\{d(a, b) \mid a \in A, b \in B\}$$

Beim Average Linkage wird der Durchschnitt aus den Abständen aller Elementpaare beider Cluster gebildet. Beim Centroid Linkage kehrt die Idee eines Clusterzentrums in Form des Schwerpunktes zurück. Die Entfernung zwischen zwei Clustern wird hierbei über die Distanz der beiden Clusterschwerpunkte definiert.

$$D_{\text{averageLinkage}}(A, B) = \frac{1}{|A| \cdot |B|} \sum_{a \in A, b \in B} d(a, b)$$

$$D_{\text{centroidLinkage}}(A, B) = d(\bar{a}, \bar{b})$$

mit  $\bar{a}$  und  $\bar{b}$  als Schwerpunkte der Cluster A und B (vgl. Miyamoto 2022). Im Vergleich zum  $k$ -Means-Algorithmus können mithilfe des agglomerativen hierarchischen Clusterings – besonders bei der Verwendung der Single Linkage Metrik – auch konkave Datensätze korrekt geclustert werden.

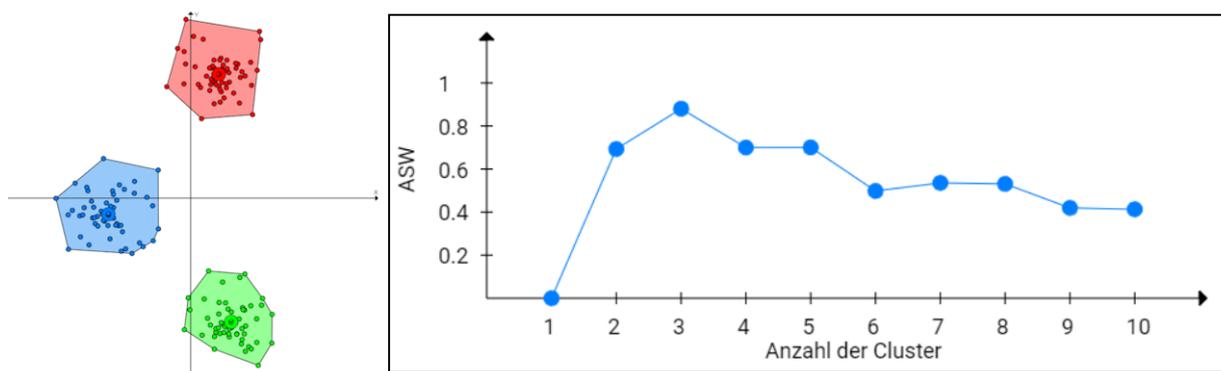


Abbildung 3: ASW Plot eines klar clusterbaren Datensatzes mit Maximum bei  $k = 3$ , Visualisierungen mit ClusterLabor

## Bestimmen der "optimalen" Clusterzahl $k$ mithilfe des agglomerativen hierarchischen Clusterings

Auch bei hierarchischem Clustering kann durch graphische Darstellungen das „optimale“  $k$  ermittelt werden. Der Verlauf des hierarchischen Clusterings kann schematisch als Baum dargestellt werden. Beginnend bei den einzelnen Datenpunkten als Startcluster (die Blätter) werden in jedem Schritt die jeweils nächsten Cluster (Teilbäume) miteinander fusioniert, bis zum Schluss ein einziges Cluster (die Wurzel) erreicht wird, welches alle Datenpunkte des Datensatzes enthält. Trägt man diesen Baum mit dem Abstand der fusionierten Cluster als die  $y$ -Koordinate der Knoten in ein Diagramm ein, erhält man das sogenannte *Dendrogramm*. Nach Konstruktion gibt die Höhe der Verbindungslinien an, wie ähnlich die verbundenen Cluster sind. Je niedriger die Höhe der Verbindungslinie, desto ähnlicher sind die in diesem Schritt fusionierten Cluster. Abb. 4 veranschaulicht einen kleinen geclusterten Datensatz mit dem zugehörigen Dendrogramm.

Ein Dendrogramm zu interpretieren kann je nach Datensatz schwierig sein. Oft sucht man für den "optimalen"  $k$ -Wert eine Stelle, an der es einen großen Sprung in der Höhe der Verbindungslinien gibt. Schneidet man an dieser Stelle waagrecht durch das Dendrogramm ergibt sich durch die Anzahl der gekreuzten senkrechten Verbindungslinien die vom Dendrogramm prognostizierte "optimale" Anzahl  $k$  an Clustern (vgl. Miyamoto 2022).

## Funktionsumfang

ClusterLabor führt den Nutzer in drei nummerierten Schritten – "Daten laden", "Daten normalisieren (optional)", "Algorithmus ausführen" – durch die Anwendung eines Clusteringalgorithmus. Zunächst müssen dazu Daten in das Programm geladen werden, welche daraufhin

normalisiert werden können. Zum Abschluss kann der gewünschte Algorithmus ausgewählt und auf die Daten angewandt werden. Hintergrundwissen zur genauen Funktionsweise der Algorithmen ist bei dem bloßen Clustern der Daten nicht erforderlich.

ClusterLabor ermöglicht verschiedene Möglichkeiten zum Laden eines Datensatzes. Es können Beispieldatensätze vom Server oder eigene Datensätze von der Festplatte geladen werden. Zudem bietet ClusterLabor an, einen klar clusterbaren Datensatz oder einen zufälligen Datensatz zu generieren. Schlussendlich ist es möglich mittels Mausklick einen Datensatz in einem interaktiven Editor in ClusterLabor selbst zu erstellen.

Ein geladener Datensatz kann in ebendiesem Editor auch nachträglich bearbeitet werden. So können einzelne Punkte hinzugefügt, verschoben oder gelöscht werden oder die Achsenbenennung angepasst werden. Diese Änderungen sind auch nach einem erfolgten Clustering noch möglich, wodurch das Verhalten der verschiedenen Algorithmen in Abhängigkeit von Ausreißern oder der Streuung der einzelnen Cluster analysiert werden kann.

Neben Datenpunkten ohne Sachzusammenhang (wie "zweiMonde"), welche zur Analyse der Stärken und Schwächen der einzelnen Clusteringalgorithmen dienen, stellt ClusterLabor auch zwei Datensätze mit Realitätsbezug zur Verfügung. Zum einen der bekannte Schwertlilien-Datensatz "iris", zum anderen der Datensatz "geraete", der die Bildschirmbreite und -höhe einiger (fiktiver) Geräte enthält, die auf eine Website zugegriffen haben. Es ergeben sich drei Cluster, die auf die verwendeten Geräte (Handy, Tablet, Laptop/Monitor) hinweisen.

Je nach Sachzusammenhang kann es nützlich oder auch notwendig sein, den Datensatz vor dem eigentlichen Clustern zu normalisieren. ClusterLabor bietet dies als Option an, es ist

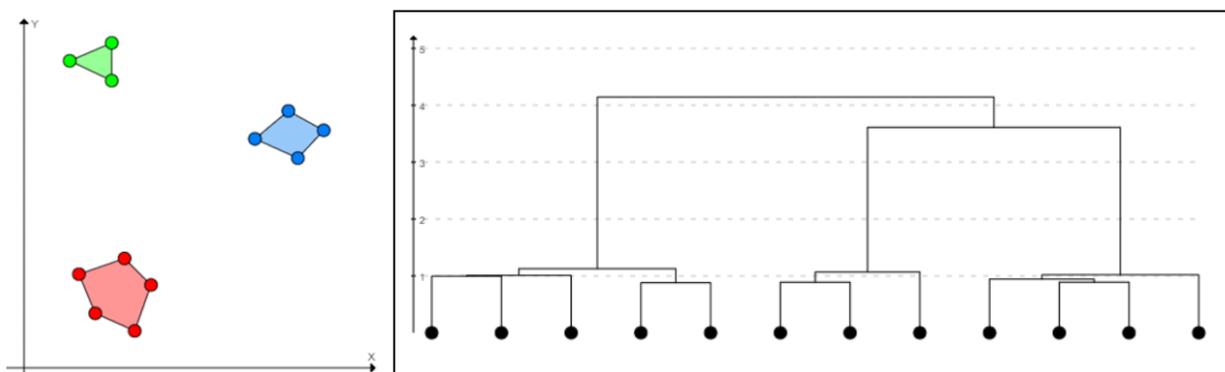


Abbildung 4: Dendrogramm eines kleinen Datensatzes, Visualisierungen mit ClusterLabor

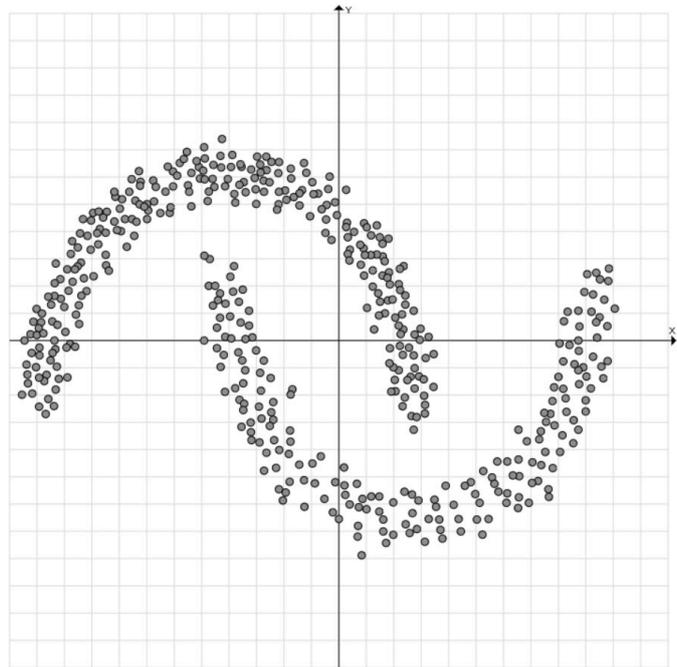
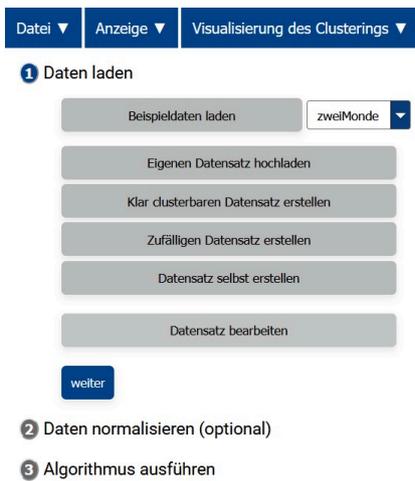


Abbildung 5: Die Ansicht in ClusterLabor, nachdem der Datensatz "zweiMonde" heruntergeladen wurde

aber auch möglich, den Datensatz direkt ohne Normalisierung zu clustern.

Zum Clustern stehen drei Algorithmen zur Auswahl: der  $k$ -Means-Algorithmus, der  $k$ -Medoids-Algorithmus und das agglomerative hierarchische Clustering. Der  $k$ -Means-Algorithmus ist mit den typischen Eigenschaften voreingestellt, sodass nur die Anzahl der Cluster eingegeben werden muss: Die Aktualisierung der Clusterzentren erfolgt durch Mittelwertbildung am Ende der Zuweisung (Lloyd), die Startzentren sind zufällig ausgewählte Punkte des Datensatzes und es wird das euklidische Abstandsmaß als Metrik verwendet. Jede dieser Parameter kann variiert werden. Es stehen verschiedene Möglichkeiten für Metrik, Startzentrenwahl, Aktualisierungsart und Aktualisierungszeitpunkt der Zentren zur Verfügung. Der  $k$ -Medoids-Algorithmus bietet weniger Möglichkeiten zur Variation. Hier kann lediglich die verwendete Metrik ausgewählt werden. Für das hierarchische Clustering kann ebenfalls die Metrik zum Messen der Distanz zwischen Datenpunkten variiert werden. Zusätzlich muss für den Algorithmus auch eine Distanz zwischen einzelnen Clustern ermittelt werden können. ClusterLabor bietet dafür die vier bekanntesten Linkage-Methoden – Single Linkage, Complete Linkage, Average Linkage und Centroid Linkage – an.

Nach dem Ausführen wird das Ergebnis des modifizierten Algorithmus in einem Koordinatensystem farblich dargestellt. Wurde der  $k$ -Means-

Algorithmus oder der  $k$ -Medoids-Algorithmus verwendet, werden die Clusterzentren als farbige Kreisringe dargestellt. Zusätzlich ist es möglich, die konvexe Hülle der Cluster oder das entsprechende Voronoi-Diagramm einzeichnen zu lassen.

Neben der Möglichkeit, einen Datensatz für eine feste Clusterzahl  $k$  mittels eines der Algorithmen clustern zu lassen, ermöglicht ClusterLabor auch eine Clustering über einen Bereich hinweg, z. B. von  $k=1$  bis  $k=5$ . Mittels eines Schiebereglers können die verschiedenen Clusterungen angezeigt und miteinander verglichen werden. Optional können bei einer Clustering über einen Bereich Graphen zur Clusteranalyse bereitgestellt werden. Bei dem  $k$ -Means- und dem  $k$ -Medoids-Algorithmus kann ein Elbow Plot oder ein ASW Plot angezeigt werden, bei hierarchischem Clustering ein Dendrogramm.

## Implementierung

ClusterLabor ist als Webanwendung auf der Basis von Scala.js erstellt worden. Als Webanwendung genügt zur Nutzung von ClusterLabor ein üblicher Browser (wie Mozilla Firefox, Google Chrome oder Microsoft Edge). Die GUI benötigt eine Mindestauflösung von 1024x768 Pixeln und ist für die Bedienung mit Maus und Tastatur optimiert. Weitere Bedingungen müssen nicht erfüllt sein. Die Anwendung selbst befindet sich unter <https://www.ddi.informatik.uni-wuerzburg.de/cluster>

Datei ▾ Anzeige ▾ Visualisierung des Clusterings ▾

- 1 Daten laden
- 2 Daten normalisieren (optional)
- 3 Algorithmus ausführen

k-Means-Algorithmus

Wähle einen Wert oder einen Bereich für die Anzahl  $k$  der Cluster:

Wert:

Von:  Bis:

Aktualisierung des Zentrums:

Am Ende der Zuweisung (Lloyd)

Schrittweise während der Zuweisung (MacQueen)

Wahl des Zentrums:

Zufälliger Datenpunkt

Zufälliger Punkt

Setze Zentren per Klick in das Koordinatensystem

Wahl der Metrik:

Euklidischer Abstand

Manhattan-Distanz

Wahl der Zentrumsbildung:

Mittelwert

Median

k-Medoids-Algorithmus

Hierarchisches Clustering

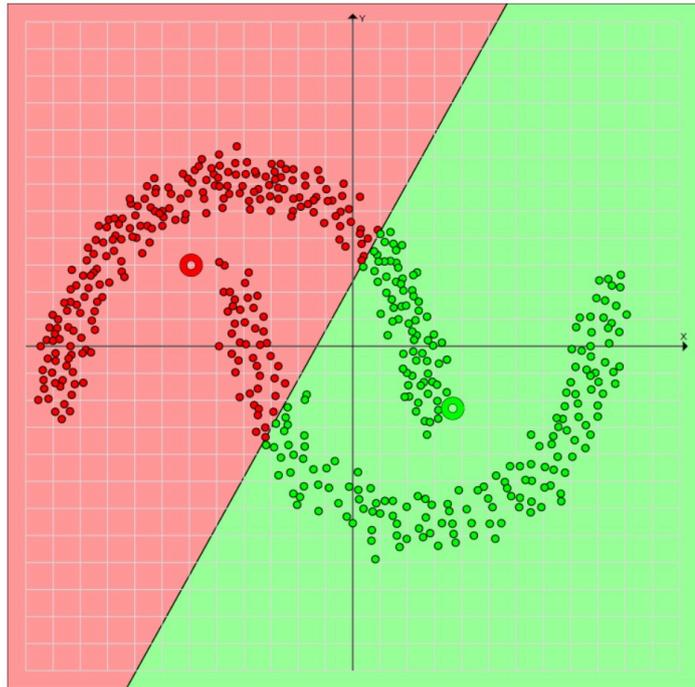


Abbildung 6: Entstandene Cluster bei Anwendung des  $k$ -Means-Algorithmus für  $k=2$  mit den von ClusterLabor voreinstellten Eigenschaften und eingezeichnetem Voronoi-Diagramm

## Ausblick

Das in diesem Beitrag vorgestellte Werkzeug ClusterLabor dient als didaktisches Werkzeug zum Vermitteln und Vergleichen von Clusteralgorithmen. Durch die interaktive Visualisierung und die verschiedenen Analysemöglichkeiten von Cluster-Ergebnissen ermöglicht ClusterLabor ein tieferes Verständnis der Funktionsweise verschiedener Algorithmen und der Auswirkungen unterschiedlicher Parameter auf die Ergebnisse dieser Algorithmen.

In naher Zukunft sollte die Effektivität und der Nutzen von ClusterLabor im Unterrichtsfeld empirisch evaluiert, Arbeitsaufträge formuliert und das Programm auf eine effiziente Anwendbarkeit hin optimiert werden. Zudem sind verschiedene Erweiterungen des Programms denkbar. Beispielsweise könnte es einen Modus geben, in dem das Clustering nicht automatisch komplett ausgeführt wird, sondern eine Animation der einzelnen Schritte bis zum Endergebnis erfolgt. So könnte z. B. die Funktionsweise des hierarchischen Clusterings oder der einzelnen Linkage-Methoden mit einem geführten Arbeitsauftrag von den Schülerinnen und Schülern selbst hergeleitet werden. Auch kleine Erweiterungen wie die Integration von anderen Clusteralgorithmen, die Verwendung zusätzlicher Metriken oder die Integration von weiteren Diagrammen und Methoden, um das "optimale"  $k$  zu bestimmen, sind denkbar und könnten im Unterricht produktiv eingesetzt werden.

## Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 30.05.2024.

Andres, D., Joachim, S., Hennecke, M.: Den  $k$ -Means-Algorithmus verstehen: Mit Stift & Papier und BlueJ. Informatische Bildung in Schulen Praxisbeiträge Jg. 2 (2024) (1), S.44-55. url: <https://doi.org/10.18420/ibis-02-01-06>

Demšar, J.; Curk, T.; Erjavec, A.; Gorup, Č.; Hočevar, T.; Milutinovič, M.; Možina, M.; Polajnar, M.; Toplak, M.; Starič, A.; Štajdohar, M.; Umek, L.; Žagar, L.; Žbontar, J.; Žitnik, M.; Zupan, B.: Orange: Data Mining Toolbox in Python. Journal of Machine Learning Research 14, S. 2349–2353, 2013, url: <http://jmlr.org/papers/v14/demsar13a.html>

Deutscher Bundestag: Drucksache 20/5149. Data-Mining – gesellschaftspolitische und rechtliche Herausforderungen, 2023, url: <https://dip.bundestag.de/vorgang/bericht-des-ausschusses-f%C3%BCr-bildung-forschung-und-technikfolgenabsch%C3%A4tzung-18-ausschuss/295126>

Ertel, Wolfgang (2021): Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung, Springer, Wiesbaden. [5. Auflage]

Holmes, G.; Donkin, A.; Witten, I. H.: Weka: A machine learning workbench. In: Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems. Australia, 1994, url: <https://ml>

[cms.waikato.ac.nz/publications/1994/Holmes-ANZIIS-WEKA.pdf](https://cms.waikato.ac.nz/publications/1994/Holmes-ANZIIS-WEKA.pdf)

Jin, X.; Han, J.: K-Medoids Clustering. In (Sammut, C.; Webb, G. I., Hrsg.): Encyclopedia of Machine Learning. Springer US, Boston, MA, S. 564–565, 2010, isbn: 978-0-387-30164-8, url: [https://doi.org/10.1007/978-0-387-30164-8\\_426](https://doi.org/10.1007/978-0-387-30164-8_426)

Miyamoto, S.: Theory of Agglomerative Hierarchical Clustering. Springer Singapore, 2022, isbn: 978-981-19-0420-2.

Pöhner, N. (2023). Filterblasen verstehen. Informatische Bildung in Schulen 1(1). url: <https://doi.org/10.18420/ibis-01-01-06>

Schubert, E.: Stop using the elbow criterion for k-means and how to choose the number of clusters instead. ACM SIGKDD Explorations Newsletter 25 (1), S. 36–42, 2023, issn: 1931-0153, doi: 10.1145/3606274.3606278, url: <http://dx.doi.org/10.1145/3606274.3606278>

Staatsministerium für Schulqualität und Bildungsforschung München: LehrplanPLUS Gymnasium Bayern, Informatik 13 und spät beginnende Informatik 13 (grundlegendes Anforderungsniveau), url: <https://www.lehrplanplus.bayern.de/fachlehrplan/gymnasium/13/informatik/grundlegend>

Staatsministerium für Schulqualität und Bildungsforschung München: LehrplanPLUS Gymnasium Bayern, Informatik 13 (erhöhtes Anforderungsniveau), url: <https://www.lehrplanplus.bayern.de/fachlehrplan/gymnasium/13/informatik/erhoeht>

## Lizenz



Dieser Artikel steht unter der Lizenz CC BY-NC-SA 4.0 zur Verfügung.

## Materialien

Weitere Materialien zum Themenfeld Künstliche Intelligenz finden Sie unter <https://go.uniwiue.de/ki>

ClusterLabor  
<https://www.ddi.informatik.uni-wuerzburg.de/cluster>

## Kontakt

Daniela Andres  
E-Mail: [daniela.andres@uni-wuerzburg.de](mailto:daniela.andres@uni-wuerzburg.de)

Dr. Silvia Joachim  
E-Mail: [silvia.joachim@uni-wuerzburg.de](mailto:silvia.joachim@uni-wuerzburg.de)

Prof. Dr. Martin Hennecke  
E-Mail: [martin.hennecke@uni-wuerzburg.de](mailto:martin.hennecke@uni-wuerzburg.de)

Didaktik der Informatik, Julius-Maximilians-Universität Würzburg, <https://go.uniwiue.de/ddi>