technische universität
dortmund

Ausarbeitung

K-Means-Clustering

Antonie Vietor

im Rahmen des Proseminars

Grundlagen des Data-Minings für strukturierte Daten

Dr. Nils M. Kriege

Wintersemester 2017/18

Fakultät für Informatik
Algorithm Engineering (Ls11)
Technische Universität Dortmund

Inhaltsverzeichnis

1 Einleitung

Die vorliegende Ausarbeitung basiert im Wesentlichen auf den beiden Werken „Mining of Massive Datasets“ (Leskovec et al., 2014) und „Foundations of Data Science“ (Blum et al., 2016). Daher werden diese Quellen für einen besseren Lesefluss im Folgenden nicht mehr einzeln aufgeführt.

Die Klassifikation von Daten kann durch unüberwachtes (unsupervised) Lernen oder durch überwachtes (supervised) Lernen erfolgen. Beim unüberwachtem Lernen besteht das Ziel darin, für eine Datenmenge Klassen mit möglichst ähnlichen Objekten zu identifizieren. Hierfür stehen verschiedene Methoden des Clusterings zur Verfügung. Beim überwachten Lernen sind die Klassen einer Datenmenge bereits bekannt und das Ziel besteht darin, Klassifikationsregeln aufzustellen, die es ermöglichen, diesen Klassen neue Daten zuordnen zu können (Voß und Buttler, 2004).

Bezüglich verschiedener Clusteringstrategien wird insbesondere zwischen hierarchischen und partitionierenden Verfahren unterschieden, daneben existieren auch Kombinationen aus beiden Verfahren sowie graphentheoretische Methoden und Optimierungsverfahren (Backhaus et al., 2008). Wie Abbildung 1 zeigt, gehört das K-Means-Clustering zu den partitionierenden, zentrumsbasierten Clusteringverfahren. Bei diesem Verfahren werden die Datenpunkte eines Datensatzes auf eine feste Anzahl von Clustern verteilt. Die Anzahl der Cluster k muss vor Ausführung des Algorithmus festgelegt werden. Jeder Datenpunkt wird genau einem Cluster in Abhängigkeit von seiner Distanz zu den verschiedenen Clusterzentren zugeordnet. Sämtliche Cluster sind somit disjunkt und jeder Punkt ist einem Cluster zugeordnet. Der Datensatz wurde also in k Partitionen aufgeteilt.

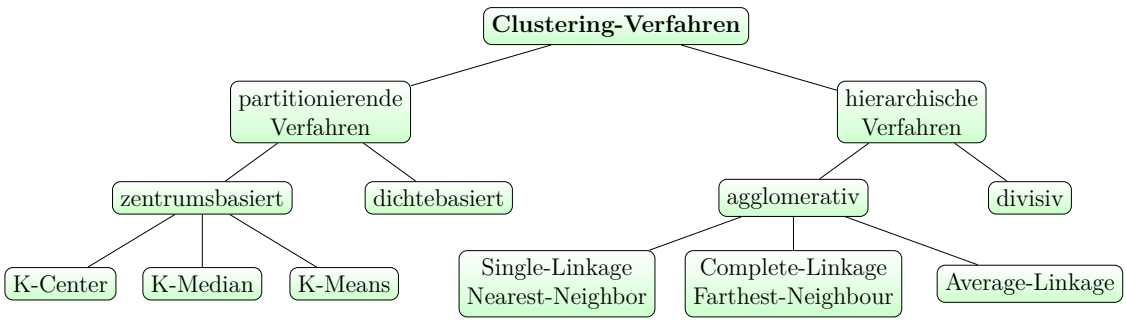


Abbildung 1: Clusteringverfahren (modifiziert nach Backhaus et al. (2008))

Neben dem K-Means-Clustering gehören zu den partitionierenden, zentrumsbasierten Verfahren auch das K-Center- und das K-Median-Verfahren. Beim K-Center-Verfahren wird die maximale Distanz zwischen den Datenpunkten und ihrem Clusterzentrum minimiert. Das K-Median-Verfahren minimiert für alle Cluster die Summe der Abstände zwischen den Datenpunkten und ihrem Clusterzentrum.

Ziel des K-Means-Clusterings ist es, für alle Cluster die Summe der quadratischen Abstände der Datenpunkte zu ihrem Clusterzentrum zu minimieren.

Im Bezug auf sogenannte Ausreißer im Datensatz weist K-Means durch die Summation der Distanzen eine bessere Robustheit als das K-Center-Verfahren auf, welches lediglich die Maxima der Distanzen berücksichtigt. Andererseits bewirkt die Quadrierung der Distanzen beim K-Means-Verfahren eine stärkere Gewichtung von Ausreißern gegenüber dem K-Median-Verfahren.

2 Berechnung und Bedeutung des K-Means

Voraussetzungen für die Anwendung des K-Means-Verfahrens sind, dass die Datenpunkte im d-dimensionalen Raum liegen und aus k sphärischen und gut von einander abgrenzbaren Datenpunktanhäufungen bestehen. Die Varianz der Datenpunkte dieser sphärischen Bereiche sollte hierbei idealerweise in jeder Dimension vom Zentrum zur Peripherie entsprechend der Gaußschen Normalverteilung abnehmen.

Sei k die vorgegebene Anzahl von Clustern, die aus dem Datensatz zu bilden sind und $C = \{c_1, ..., c_k\}$ die Menge der Vektoren der Clusterzentren. Sei zudem n die Anzahl der Datenpunkte im Datensatz $A = \{a_1, ..., a_n\}$. Dann lässt sich die Summe der quadratischen Abstände der Datenpunkte eines Clusters zu seinem Clusterzentrum c_j wie folgt errechnen (Jain, 2010):

$$J(c_j) = \sum_{a_i \in c_j} \|a_i - c_j\|^2 = \sum_{a_i \in c_j} d^2(a_i, c_j)$$

Hierbei können in Abhängigkeit von den Eigenschaften des Datensatzes als Abstand d auch verschiedene Distanzmaße verwendet werden (Morissette und Chartier, 2013), z.B. der euklidische Abstand im 2-dimensionalen Raum oder die Mahalanobis-Distanz im d-dimensionalen Raum bei Normalverteilung in allen Dimensionen (siehe Kapitel 5).

Die Summe der quadratischen Abstände $J(C)$ der Datenpunkte zu ihrem Clusterzentrum für alle k Cluster lässt sich somit als die Summe aller $J(c_j)$ darstellen:

$$J(C) = \sum_{j=1}^{k} J(c_j) = \sum_{j=1}^{k} \sum_{a_i \in c_j} \|a_i - c_j\|^2 = \sum_{j=1}^{k} \sum_{a_i \in c_j} d^2(a_i, c_j)$$

Beim K-Means-Verfahren soll $J(C)$ minimal werden, d.h. es müssen geeignete Clusterzentren c_j gewählt werden. Betrachtet man ein einzelnes Cluster, so werden die aufsummierten Distanzen zu allen im Cluster enthaltenen Punkten a_i genau dann minimal, wenn c_j der Mittelwert der betrachtetet Merkmale der Datenpunkte des Clusters ist. Da die Datenpunkte durch Vektoren a_i im d-Dimensionalen Raum repräsentiert werden und jedes Merkmal einer Dimension entspricht, ergibt sich der Mittelwert der Merkmale aus dem Mittelwert der einzelnen Koordinaten von a_i.

Seien die Koordinaten eines Clusterzentrums c also:

$$c = \frac{1}{n} \sum_{i=1}^{n} a_i = \bar{a}$$

Dann gilt für die Summe der quadratischen Abstände aller n Punkte eines Clusters zu einem Punkt x:

$$\begin{aligned}
\sum_{i=1}^{n} |a_i - x|^2 &= \sum_{i=1}^{n} |a_i - c + c - x|^2 \\
&= \sum_{i=1}^{n} |a_i - c|^2 + 2(c - x) \cdot \sum_{i=1}^{n} (a_i - c) + \sum_{i=1}^{n} |c - x|^2 \\
&= \sum_{i=1}^{n} |a_i - c|^2 + 2(c - x) \cdot \sum_{i=1}^{n} (a_i - c) + n|c - x|^2 \\
&= \sum_{i=1}^{n} |a_i - c|^2 + 2(c - x) \cdot 0 + n|c - x|^2 \\
&= \sum_{i=1}^{n} |a_i - c|^2 + n|c - x|^2
\end{aligned}$$

Denn es gilt:

$$\sum_{i=1}^{n} (a_i - c) = \sum_{i=1}^{n} (a_i - \bar{a}) = 0$$

Und $\sum_{i=1}^{n} |a_i - c|^2 + n|c - x|^2$ wird genau dann minimal, wenn gilt $x = c$. Also wird $J(C)$ minimal, wenn das Clusterzentrum c der Mittelwert der betrachteten Merkmale der Datenpunkte des Clusters ist.

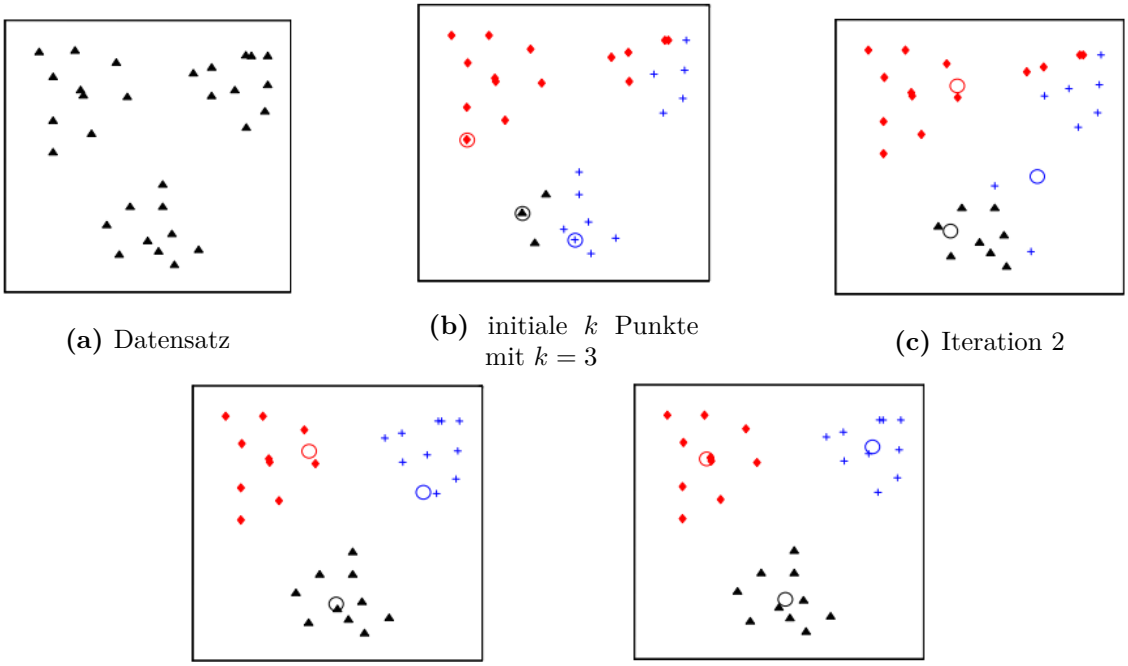
3 K-Means-Basisalgorithmus nach Lloyd

Der Begriff K-Means wurde durch MacQueen (1967) geprägt, allerdings gilt heute der Algorithmus von Lloyd (1982) als Basisalgorithmus für das K-Means-Clustering.

1: Wähle k Clusterzentren.
2: Ordne jeden Datenpunkt dem am nächsten gelegenen Clusterzentrum zu.
3: Bestimme das Clusterzentrum jedes Clusters erneut und ersetze die bisherigen Clusterzentren.
4: Wiederhole Schritt 2 und 3 bis sich die Positionen der Clusterzentren nicht mehr ändern.

Algorithm 3.1: Algorithmus nach Lloyd

Bei jeder Iteration wird also jeder Datenpunkt a_i erneut geprüft und demjenigen Cluster zugewiesen, dessen Clusterzentrum die geringste Distanz zu a_i hat. Während der Ausführung des Algorithmus können die Clusterzentren wandern und zwar in Abhängigkeit von den Punkten, die ihnen zugewiesen werden (siehe Abbildung 2). Da jedem Clusterzentrum allerdings nur Punkte in seiner Nähe zugewiesen werden, handelt es sich meist um keine großen Positionswechsel der Clusterzentren.



(a) Datensatz

(b) initiale k Punkte mit $k = 3$

(c) Iteration 2

(d) Iteration 3

(e) Clustering beendet

Abbildung 2: Darstellung des K-Means-Clusterings an einem zweidimensionalen Beispieldatensatz (nach Jain (2010))

Nachteile des Lloyd-Algorithmus ergeben sich vor allem aus der zufälligen Wahl der initialen Clusterzentren. Hierdurch können einerseits im Laufe der Iterationen „leere“ Cluster entstehen, für die eine Kalkulation ihres Clusterzentrums nicht mehr möglich ist. Folglich können einmal entstandene „leere“ Cluster bei den folgenden Iterationen nicht mehr berücksichtigt werden. Andererseits werden vor allem lokale Optima bestimmt, die jedoch nicht unbedingt auch ein globales Optimum im Hinblick auf die Verteilung der Cluster über dem gesamten Datensatz darstellen (Morissette und Chartier, 2013).

Die Ergebnisse des K-Means-Clusterings sind also maßgeblich von der initialen Wahl der k Clusterzentren in Schritt 1 des Algorithmus abhängig. Hierfür können k Punkte gewählt werden, die von einander möglichst weit entfernt liegen. Ebenso können aus einer Stichprobe der Daten zunächst hierarchisch k Cluster aufgebaut werden, aus denen dann jeweils ein Punkt als initiales Clusterzentrum zur Analyse des gesamten Datensatzes bestimmt wird.

Die optimale Clusteranzahl k lässt sich durch wiederholte Messungen ermitteln (Morissette und Chartier, 2013), insbesondere die Bestimmung des Clusterradius und -durchmessers sind hierfür geeignet. Der Clusterradius bezeichnet die maximale Distanz zwischen den Punkten des Clusters und seinem Clusterzentrum, der Clusterdurchmesser gibt die maximale Distanz zwischen zwei Punkten des Clusters an. Der durchschnittliche Radius bzw. Durchmesser eines Clusters verändert sich nur leicht beim Hinzufügen von weiteren Punkten, solange die Anzahl der zu erzeugenden Cluster der tatsächlichen Anzahl an Clustern im Datensatz entspricht oder höher liegt. Werden aber zwei Cluster zusammengefasst, die nicht zusammen passen, kommt es zu einem deutlichen Anstieg bei der Größe des Radius bzw. Durchmessers des Clusters. Eine effektive Vorgehensweise ist hierbei, für k Zweierpotenzen zu verwenden. Wurde dadurch ein passendes Intervall eingegrenzt, kann dieses Intervall ggf. durch weitere binäre Aufteilungen (binäre Suche) weiter angepasst werden, um so einen optimalen Wert für k zu ermitteln.

4 Ward-Algorithmus

Dieser Algorithmus startet mit einem separaten Cluster für jeden Datenpunkt. Schrittweise werden die kleineren Cluster zu größeren zusammengefasst, bis schließlich genau k Cluster gebildet wurden. Bei der Fusion der Cluster werden im Sinne eines Greedy-Algorithmus diejenigen Cluster zusammengefasst, welche die Varianz in einer Gruppe am wenigsten erhöhen (Backhaus et al., 2008). Problematisch bei diesem Clustering-Verfahren ist allerdings, dass der Algorithmus dazu tendiert, gleich große Cluster zu bilden (Voß und Buttler, 2004), was jedoch bei realen Datensätzen eine eher ungewöhnliche Verteilung ist.

5 BFR-Algorithmus

Der Algorithmus nach Fayyad et al. (1998) (BFR-Algorithmus) wird für Clusterdaten im mehrdimensionalen euklidischen Raum verwendet und eignet sich insbesondere für große Datenmengen, sogenannte Big Data. Voraussetzung für die Verwendung des Algorithmus ist allerdings, dass die Datenpunkte der Cluster in jeder Dimension eine Normalverteilung um die Clusterzentren herum besitzen. Dadurch wird eine runde/ellipsoide Clusterform um die Raumachsen herum impliziert. Der Mittelwert und die Standardabweichung dürfen sich in den verschiedenen Dimensionen unterscheiden, aber die Dimensionen müssen voneinander unabhängig sein. Abweichende Rotationen bezüglich der Raumachsen in den verschiedenen Dimensionen sind allerdings nicht zulässig.

5.1 Initialisierung

Die initiale Auswahl einer optimalen Anzahl von k Punkten kann mit der bereits beim K-Means-Basisalgorithmus vorgestellten Methode (siehe Kapitel 3) erfolgen. Anschließend werden die Daten auf n verschiedene Stichproben zufällig aufgeteilt.

Dies bietet den Vorteil, dass insbesondere bei Big Data nicht der gesamte Datensatz gleichzeitig in den Speicher eingelesen werden muss. Die Daten können in Fragmenten geladen werden, die relevanten Statistiken werden ermittelt und nur diese werden für die spätere Analyse des gesamten Datensatzes im Speicher belassen. Für diese relevanten Statistiken werden die folgenden drei Datenkategorien angelegt (siehe Abbildung 3):

- **DS (Discard Set):** enthält die eigentlichen Cluster mit denjenigen Punkten, die nahe genug an einem der k Clusterzentren liegen, so dass sie einem Cluster sicher zugeordnet werden können (siehe Kapitel 5.3).
- **CS (Compressed Set):** enthält Gruppen von Datenpunkten, die zwar nahe bei einander liegen, aber keinem der zuvor festgelegten Clusterzentren zugeordnet werden konnten, da sich diese Punkte nicht nahe genug an einem der Clusterzentren befinden.
- **RS (Retained Set):** in dieser Kategorie werden isolierte Datenpunkte gesammelt, sie liegen weder nahe genug an den Clusterzentren des CS noch an denen des DS.

5.2 Datenanalyse

Für jedes Cluster des DS bzw. CS werden folgende Werte berechnet (siehe Abbildung 3):

- N als Anzahl der Punkte des Clusters.
- Der Vektor SUM, dessen i-tes Element jeweils der Summe der Koordinaten der i-ten Dimension d von allen im Cluster enthaltenen Punkten entspricht. Dieser Vektor besitzt also die Länge d.
- Der Vektor $SUMSQ$, dessen i-tes Element jeweils die Summe der Koordinatenquadrate der i-ten Dimension von allen im Cluster enthaltenen Punkten darstellt. Dieser Vektor besitzt ebenfalls die Länge d.

Bei d-dimensionalen Daten wird also jedes Cluster des DS und des CS durch $2d+1$ Werte repräsentiert. Ziel ist es ein Set der Daten zu erzeugen, in dem ihre Anzahl, ihr Clusterzentrum und ihre Standardabweichung für jede Dimension gespeichert ist. Die o.g. $2d + 1$ Werte ermöglichen diese Metadaten:

- **Anzahl der Datenpunkte:** ist durch N bereits gespeichert.
- **Clusterzentrum:** aufgrund der Normalverteilung in jeder Dimension, entsprechen die Koordinaten des Clusterzentrums den Mittelwerten der Datenpunktkoordinaten in jeder Dimension. Somit können die Koordinaten des Clusterzentrums durch die Elemente des SUM-Vektors für jede i-te Dimension dargestellt werden durch:

$$\frac{SUM_i}{N}$$

– **Standardabweichung:** durch die SUM- und $SUMSQ$-Vektoren lässt sich die Varianz v in der i-ten Dimension wie folgt ermitteln:

$$v = \frac{SUMSQ_i}{N} - \left(\frac{SUM_i}{N}\right)^2$$

Aufgrund der Normalverteilung der Daten kann hieraus auch die Standardabweichung s bestimmt werden, denn es gilt $s = \sqrt{v}$.

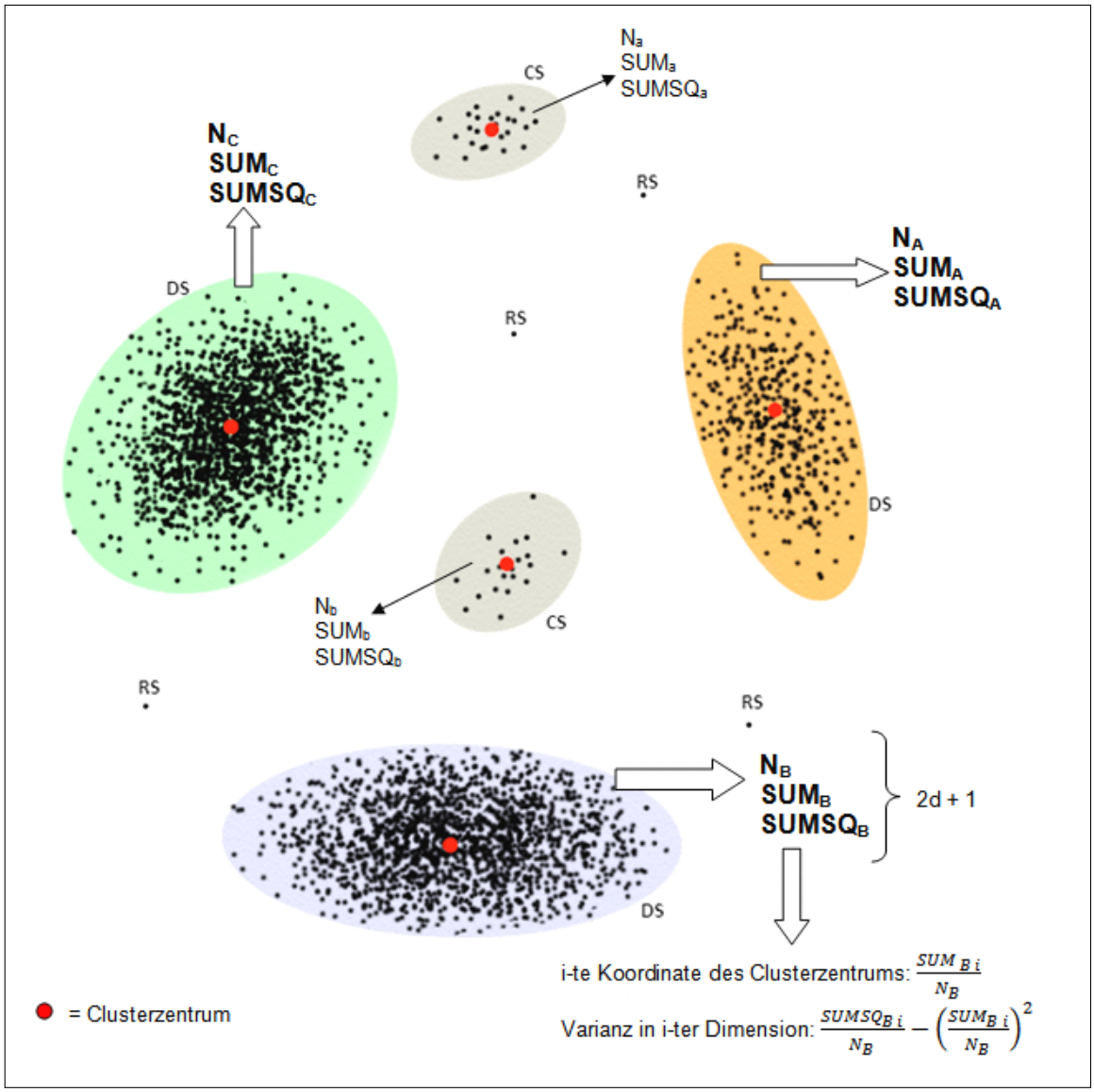


Abbildung 3: Darstellung der Datenkategorien des BFR-Algorithmus und der aus ihnen ermittelten Werte.

Aktualisierung der Metadaten:

Die Punkte werden einzeln den Sets zugeordnet. Sobald ein Datenpunkt dem DS oder CS zugeordnet wurde, werden N, SUM und $SUMSQ$ für das entsprechende Cluster angepasst und der Punkt wird verworfen. Dadurch ist eine sehr einfache Aktualisierung des Clusterzentrums und der Standardabweichung möglich:

- Die Anzahl der Daten wird um 1 erhöht: $N + 1$.
- Die Koordinaten des Punktes werden zum SUM-Vektor hinzuaddiert.
- Die Quadrate der Koordinaten des Punktes werden zum $SUMSQ$-Vektor hinzuaddiert.

Würden hier statt des SUM-Vektors direkt die Koordianten des Clusterzentrums verwendet werden, so würde die Berechnung der Standardabweichung beim Hinzufügen eines neuen Punktes deutlich komplexer werden.

Zuordnung der Punkte zu den Datensets:

1. Datenpunkte, die nahe genug an einem Clusterzentrum liegen werden dem DS bzw. CS zugeordnet.
2. Datenpunkte, die nicht nahe genug an einem der Clusterzentren liegen, werden mit den bislang vorhandenen Punkten im RS (aller vorherigen Stichproben) geclustert und falls möglich zu Compressed Sets zusammengefasst. Diejenigen Datenpunkte, die zu einem CS zusammengefasst wurden, werden aus dem RS entfernt.
3. Die so neu entstandenen CS werden mit den bereits vorhandenen CS abgeglichen und ggf. zu einem größeren CS zusammengefasst, falls sie nahe beieinander liegen. N, SUM und $SUMSQ$ werden beim zusammenführen der CS entsprechend aufsummiert.

Durch die schrittweise Verteilung der Datenpunkte auf DS, CS und RS bei jeder Stichprobe, erfolgt eine permanente Anpassung bzw. Korrektur der k Cluster, für die durch N, SUM und $SUMSQ$ die Koordinaten ihrer Clusterzenteren sowie ihre Standardabweichung in jeder Dimension gespeichert werden.

Nach Analyse aller Stichproben können alle verbliebenen CS und alle verbliebenen Datenpunkte des RS dem am nächsten liegenden Clusterzentrum zugewiesen werden. Ebenso besteht die Möglichkeit, diese Punkte als Ausreißer im Bezug auf das RS und Minicluster im Bezug auf das CS zu behandeln.

5.3 Distanzmaß

Um zu entscheiden, ob ein Datenpunkt nahe genug an einem der Clusterzentren des DS bzw. CS liegt, wird beim BFR-Algorithmus als Distanzmaß die Mahalanobis-Distanz verwendet. Die Mahalanobis-Distanz berücksichtigt im d-dimensionalen Raum

die Normalverteilung der Datenpunkte um das Clusterzentrum und damit die ellipsoide Form der Cluster. Bei Verwendung der euklidischen Distanz würde hingegen ein kreisförmiger Bereich um das Clusterzentrum berücksichtigt werden. Dadurch könnte ein Punkt durchaus außerhalb eines ellipsoiden Clusters liegen und würde dem Cluster dennoch zugeordnet werden.

Die Mahalanobis-Distanz bestimmt somit auch indirekt die Wahrscheinlichkeit eines Datenpunktes, einem bestimmten Cluster anzugehören und zwar in Abhängigkeit von der Normalverteilung um das Clusterzentrum. Mit den im DS bzw. CS gespeicherten Werten kann die Mahalanobis-Distanz für jeden Punkt wie folgt leicht berechnet werden.

Seien die Koordinaten eines Clusterzentrums gegeben durch:

$$c = (c_1, c_2, ..., c_d) = \left(\frac{SUM_1}{N}, \frac{SUM_2}{N}, ..., \frac{SUM_d}{N}\right)$$

Und seien die Standardabweichungen des Clusterzentrums in jeder Dimension gegeben durch:

$$\begin{aligned}\sigma &= (\sigma_1, ..., \sigma_d) \\ &= \left(\left(\frac{SUMSQ_1}{N} - \left(\frac{SUM_1}{N}\right)^2\right), ..., \left(\frac{SUMSQ_d}{N} - \left(\frac{SUM_d}{N}\right)^2\right)\right)\end{aligned}$$

Für einen Punkt a mit den Koordinaten $a = (a_1, a_2, ..., a_d)$ errechnet sich die normalisierte (euklidische) Distanz z in der Dimension i durch:

$$z_i = \frac{a_i - c_i}{\sigma_i}$$

Die Mahalanobis-Distanz MD ergibt sich dann aus der Summe der quadrierten, normalisierten Distanzen jeder Dimension zu diesem Clusterzentrum:

$$MD = \sqrt{\sum_{i=1}^{d} z_i^2}$$

Durch Festlegung einer bestimmten Grenze (1-, 2-, 3-fache Standardabweichung) kann entschieden werden, ob ein Punkt einem bestimmten Cluster zugeordnet wird. Diese Grenze kann bestimmt werden, indem ein Punkt angenommen wird, der in jeder Dimension genau die einfache Standardabweichung vom Clusterzentrum entfernt liegt. Dann gilt:

$$z_i = \frac{\sigma_i}{\sigma_i} = 1$$

Und somit:

$$MD = \sqrt{\sum_{i=1}^{d} 1^2} = \sqrt{d}$$

Eine Grenze für die 2- bzw. 3-fache Standardabweichung wären somit $2\sqrt{d}$ bzw. $3\sqrt{d}$. Im Grunde beschreibt die Mahalanobis-Distanz also die Korrelation zwischen einem Datenpunkt und dem Clusterzentrum, d.h. die Mahalanobis-Distanz nimmt zu, wenn die Korrelation abnimmt (Backhaus et al., 2008).

Um zu bestimmen, ob zwei CS nahe genug beieinander liegen, um zusammengefasst zu werden, wird die Varianz beider CS-Subcluster kombiniert. Hierzu werden von beiden CS-Clustern N, SUM und $SUMSQ$ aufsummiert und aus diesen Werten die Varianz des neuen kombinierten Clusters bestimmt. Wenn die so ermittelte Varianz unterhalb einer festgelegten Grenze liegt, werden beide CS zusammengeführt.

5.4 Kritische Betrachtung des BFR-Algorithmus

Im Gegensatz zum K-Means-Basisalgorithmus arbeitet der BFR-Algorithmus nicht iterativ, sondern es werden multiple Lösungen für multiple Startpunkte generiert und im Laufe der Datenanalyse ggf. fusioniert (Bradley et al., 1998). Der Datensatz muss also lediglich ein Mal durchlaufen werden.

Da für die analysierten Daten lediglich die Parameter N, SUM und $SUMSQ$ gespeichert werden, ermöglicht der BFR-Algorithmus die Analyse großer Datensätze unter sparsamem Einsatz des verfügbaren Speicherplatzes. Allerdings ist es durch die Verwendung dieser wenigen Parameter nicht mehr möglich einzelne Datenpunkte aus den gewonnenen Statistiken abzurufen. Ebenso können Hintergrundrauschen und Ausreißer das Ergebnis des BFR-Algorithmus deutlich beeinflussen (Kumar). Insbesondere kann sich bei großen, stark streuenden Datensätzen eine große Datenmenge im RS-Set ansammeln. Bei der Analyse von Big-Data-Datensätzen kann dies zu Problemen bezüglich des Speicherplatzbedarfes führen.

Auch das strenge Kriterium der Normalverteilung um das Clusterzentrum und die damit verbundene ellipsoide Form der Cluster kann sich als problematisch erweisen. Wie Kumar ebenfalls kritisch anmerkt, ist dieses Kriterium bei realen Messdaten häufig nicht gegeben. Auch kann das Kriterium der ellipsoiden Clusterform durch überlappende bzw. fusionierte Cluster gestört werden.

Insgesamt stellt der BFR-Algorithmus also eine gute Analysemethode für multivariante Big-Data-Datensätze dar, die das Kriterium der Normalverteilung um das Clusterzentrum herum erfüllen und eine möglichst geringe Streuung zwischen den Clustern aufweisen.

6 Ausblick

Neben den vorgestellten K-Means-Algorithmen finden sich in der Literatur zahlreiche Abwandlungen und Weiterentwicklungen der K-Means-Clusteringmethode (Bagirov und Mardaneh, 2006; Wu, 2008; Aletti und Micheletti, 2017). Sie finden insbesondere im Bereich der Molekulargenetik und Proteinanalyse Anwendung.

Ein in den vergangenen Jahren stark wachsender Zweig der Molekulargenetik ist die Genexpressionsanalyse, insbesondere im Bereich der Tumorforschung. Hierbei wird zeitgleich die Aktivität von in der Regel mehreren tausend Genen gemessen. Um Veränderungen bezüglich einer Erkrankung oder der Effizienz einer Therapie feststellen zu können, werden dazu die Genaktivitäten im kranken bzw. behandelten Gewebe im Vergleich zum gesunden gleichartigen Gewebe des selben Individuums betrachtet. Auf diese Weise lässt sich eine verstärkte Aktivierung oder Hemmung bestimmter Gene im krankhaft veränderten oder behandelten Gewebe detektieren.

Mithilfe des K-Means-Clusterings und auch anderer Clusteringverfahren können nun Gruppen von Genen identifiziert werden, welche ähnliche Aktivitätsmuster zeigen (siehe Abbildung 4 und 5). Hierbei können Gene, die sich im gleichen Cluster befinden, die also ein ähnliches Aktivitätsmuster aufweisen, auf die Aktivierung oder Hemmung bestimmter Stoffwechselwege hindeuten. Die so identifizierten Stoffwechselwege können Ansatzpunkte für neue Therapien der untersuchten Erkrankung darstellen.

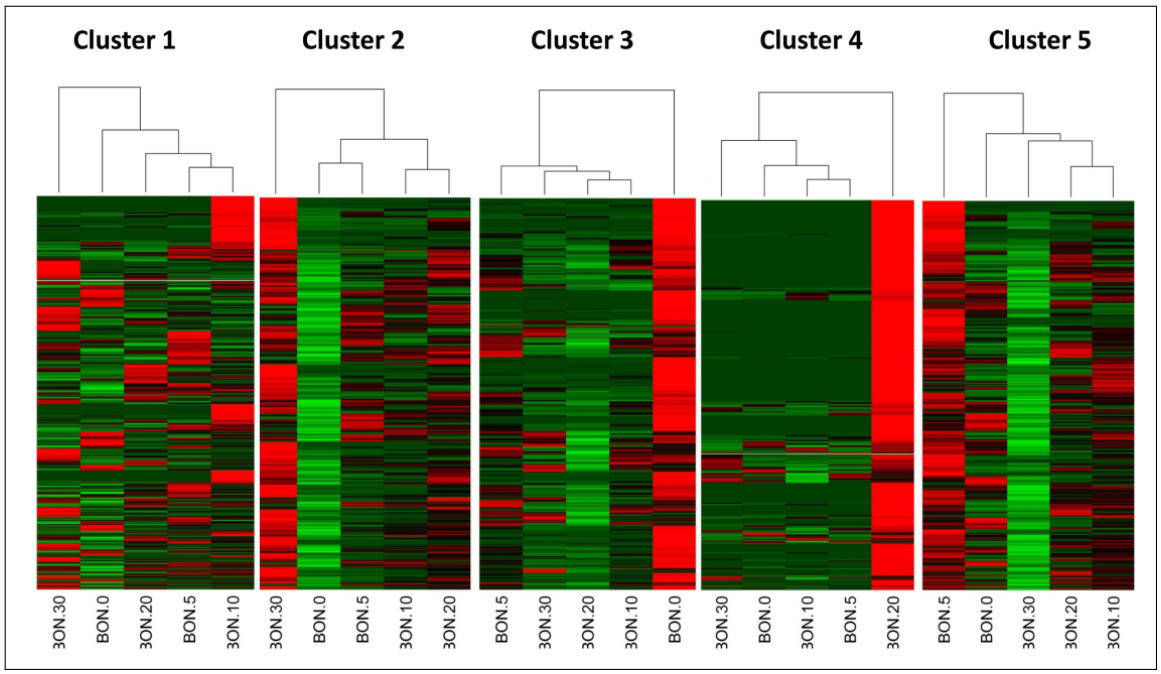


Abbildung 4: Expressionsprofile verschiedener mittels K-Means-Clustering gruppierter Gene aus verschiedenen Larvenstadien des Atlantik-Thunfischs (rot: verstärkte Genaktivität, grün: reduzierte Genaktivität) (Sarropoulou et al., 2014).

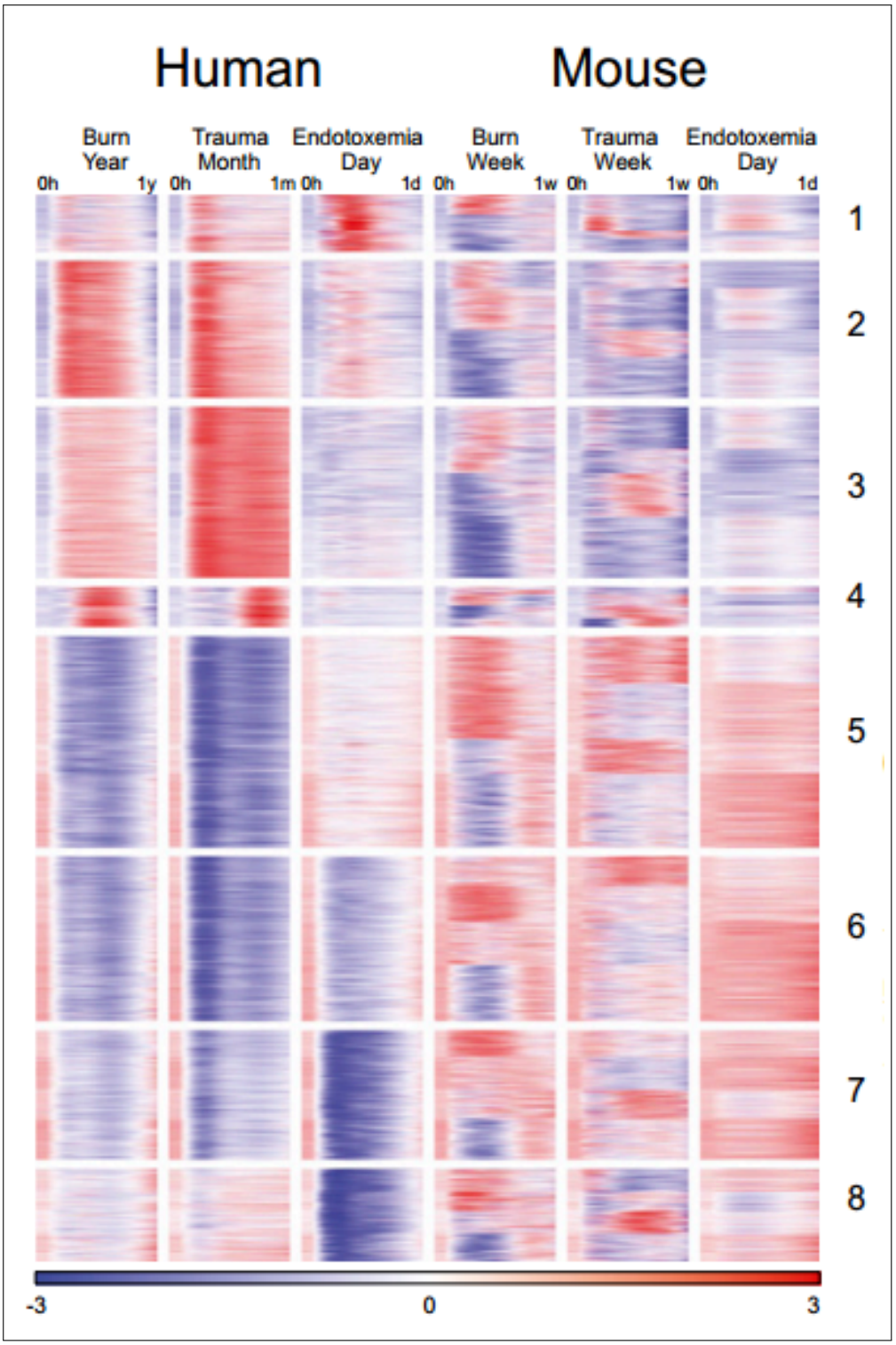


Abbildung 5: K-Means-Clustering von 4918 entzündungsassoziierten Genen bei Mensch und Maus (Seok et al., 2013). Die Messungen erfolgten zu verschiedenen Zeitpunkten im Anschluss an bestimmte entzündungsinduzierende Ereignisse. Rote Bereiche entsprechen verstärkt exprimierten Genen, blaue Bereiche zeigen Gene mit reduzierter Genaktivität an.

Literatur

G. Aletti und A. Micheletti. A clustering algorithm for multivariate data streams with correlated components. *Journal of Big Data*, 4(1):48, Dec 2017. ISSN 2196-1115. doi:10.1186/s40537-017-0109-0. URL `https://doi.org/10.1186/s40537-017-0109-0`.

K. Backhaus, B. Erichson, W. Plinke, und R. Weiber. *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung (Springer-Lehrbuch) (German Edition).* Springer, 2008. ISBN 9783540850441.

A. M. Bagirov und K. Mardaneh. Modified global k-means algorithm for clustering in gene expression data sets. In *Proceedings of the 2006 Workshop on Intelligent Systems for Bioinformatics - Volume 73*, WISB '06, Seiten 23–28, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc. ISBN 1-920-68254-6. URL `http://dl.acm.org/citation.cfm?id=1274172.1274176`.

A. Blum, J. Hopcroft, und R. Kannan. *Foundations of data science.* 2016. URL `http://infolab.stanford.edu/~ullman/mmds/book.pdf`.

P. S. Bradley, U. M. Fayyad, C. Reina, et al. Scaling clustering algorithms to large databases. In *Proceedings of the 4th International Conference on Knowledge Discovery & Data Mining (KDD98)*, Seiten 9–15, 1998. URL `https://www.aaai.org/Papers/KDD/1998/KDD98-002.pdf`.

U. Fayyad, C. Reina, und P. Bradley. Initialization of iterative refinement clustering algorithms. In *Proc. of KDD-1998*, Seiten 194–198. AAAI Press, August 1998. URL `https://www.aaai.org/Papers/KDD/1998/KDD98-032.pdf`.

A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31 (8):651–666, 2010. URL `https://www.sciencedirect.com/science/article/pii/S0167865509002323`.

G. Kumar. Reaction paper on bfr clustering algorithm. URL `https://de.scribd.com/document/181338472/Anatomy-of-BFR-clustering-algorithm`.

J. Leskovec, A. Rajaraman, und J. D. Ullman. *Mining Massive Datasets.* 2014. URL `http://infolab.stanford.edu/~ullman/mmds/book.pdf`.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, März 1982. URL `http://ieeexplore.ieee.org/document/1056489/`.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Seiten 281–297, Berkeley, Calif., 1967. University of California Press. URL `https://projecteuclid.org/euclid.bsmsp/1200512992`.

L. Morissette und S. Chartier. The k-means clustering technique: General considerations and implementation in mathematica. *Tutorials in Quantitative Methods for Psychology*, 9(1):15–24, 2013. doi:10.20982/tqmp.09.1.p015. URL `http://www.tqmp.org/RegularArticles/vol09-1/p015/p015.pdf`.

E. Sarropoulou, H. K Moghadam, N. Papandroulakis, F. de la Gándara, A. Ortega, und P. Makridis. The atlantic bonito (sarda sarda, bloch 1793) transcriptome and detection of differential expression during larvae development. *PLOS ONE*, 9:e87744, 02 2014. URL `http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0087744`.

J. Seok, H. S. Warren, A. G. Cuenca, M. N. Mindrinos, H. V. Baker, W. Xu, D. R. Richards, G. P. McDonald-Smith, H. Gao, L. Hennessy, C. C. Finnerty, C. M. López, S. Honari, E. E. Moore, J. P. Minei, J. Cuschieri, P. E. Bankey, J. L. Johnson, J. Sperry, A. B. Nathens, T. R. Billiar, M. A. West, M. G. Jeschke, M. B. Klein, R. L. Gamelli, N. S. Gibran, B. H. Brownstein, C. Miller-Graziano, S. E. Calvano, P. H. Mason, J. P. Cobb, L. G. Rahme, S. F. Lowry, R. V. Maier, L. L. Moldawer, D. N. Herndon, R. W. Davis, W. Xiao, R. G. Tompkins, the Inflammation, und L. S. C. R. P. Host Response to Injury. Genomic responses in mouse models poorly mimic human inflammatory diseases. *Proceedings of the National Academy of Sciences*, 110(9):3507–3512, 2013. doi:10.1073/pnas.1222878110. URL `http://www.pnas.org/content/110/9/3507.abstract`.

W. Voß und G. Buttler. *Taschenbuch der Statistik: mit 126 Tabellen*. Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2004. ISBN 9783446226050.

F.-x. Wu. Genetic weighted k-means algorithm for clustering large-scale gene expression data. *BMC bioinformatics*, 9(6):S12, 2008. URL `https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-S6-S12`.