Datamining – Ein kleiner Einblick

Autoren: Boris Kulig u. Bertram Schäfer

Inhaltsverzeichnis

1 Begriff, Funktion, Verfahren 1

2 Clusteranalyse 1

2.1 Proximitätsmaße 3
2.1.1 Nominal-Skala 3
2.1.2 Metrische Skalen 4
2.2 Daten Transformation / Standardisierung 5
2.3 Clusterverfahren / Fusionsalgorithmen 6
2.3.1 Hierarchisches Clustering 6
2.3.2 K-Means Clustering 9
2.4 Die Darstellung von Clustern / Ergebnisse der Clusteranalyse 11

3 Decision Trees / Classification Tree / Regression Trees 13

4 Neuronale Netze 14

5 Durchführung mit JMP an Hand von Beispielen 15

5.1 Clusteranalyse 15
5.1.1 Grundsätzliche Vorgehensweise 15
5.1.2 Beispiel für eine Hierarchisches Clusteranalyse 16
5.1.3 Beispiel für die K-Means-Clusteranalyse 19
5.2 Trees 22
5.2.1 Ein einfaches Beispiel 22

STAT*CON* B. Schäfer
Schulstr. 2
37213 Witzenhausen
Tel.: 05542-93300
Fax: 05542-933030
E-Mail: vertrieb@statcon.de

1 Begriff, Funktion, Verfahren

Der Begriff Datamining ist im allgemeinen recht ungenau definiert. In jedem Fall beinhaltet das Datamining die Suche nach *vorteilhaften Mustern in Daten* und die Anwendung von Verfahren zum Datamining. Die Entdeckung von *vorteilhaften Mustern* und entsprechenden Schlussfolgerungen können eine große Hilfe in Entscheidungs- bzw. Optimierungsprozessen sein. Die oftmals großen Datensätze (hohe Anzahl von Fällen und Variablen) und der Ansatz neue Zusammenhänge in vorhandenen Datensätzen zu finden macht es schwierig a priori Hypothesen zur Auswertung aufzustellen. Dataminingverfahren sollten also weitestgehend ohne a priori Hypothesen auskommen oder diese auf eine einfache Stufe stellen.

Typische Verfahren des Dataminings sind Clusteranalyse, Tree-Verfahren und neuronale Netzwerke. Für Clusteranalyse und Tree-Verfahren werden neben theoretischen Grundlagen auch praktische Beispiele besprochen.

2 Clusteranalyse

Mit dem Begriff Clusteranalyse wird ein Verfahren zur Bildung von Gruppen innerhalb einer Stichprobe bezeichnet. Sie kommt zum Einsatz, wenn bei einem großen Stichprobenumfang eine große Zahl von Daten (Variablen) erhoben wird und keine konkrete Hypothese zum Datenmodell vorhanden ist.

Die übliche Situation ist, dass die Daten nicht gleichmäßig im n-dimensionalen Raum verteilt sind. Sie bilden vielmehr Klumpungen, lokale Dichtezonen, bzw. Clusterungen. Ein abbildbares, 2-dimensionales Beispiel einer Clusterung zeigt folgende Grafik:

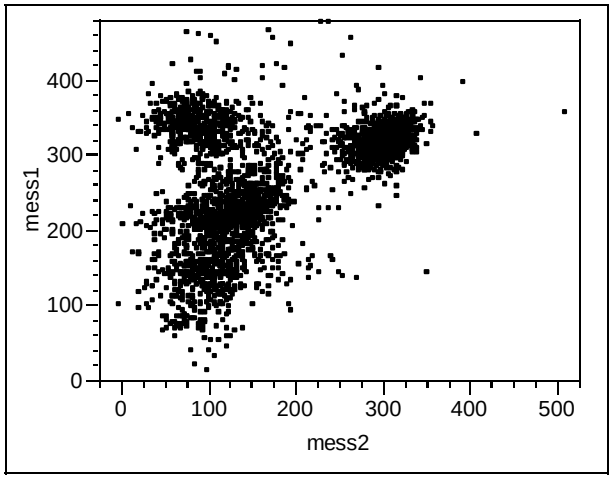

Abbildung 1: 2-dimensionales Beispiel einer Clusterung

Ziel der Clusteranalyse ist es nun die Daten zu Gruppen zusammenzufassen. Die Daten innerhalb einer Gruppe sollten eine möglichst verwandte Eigenschaftsstruktur aufweisen. Darüber hinaus sollten die Gruppen untereinander möglichst unähnlich sein. Folgende Abbildung zeigt eine mögliche Clusterung.

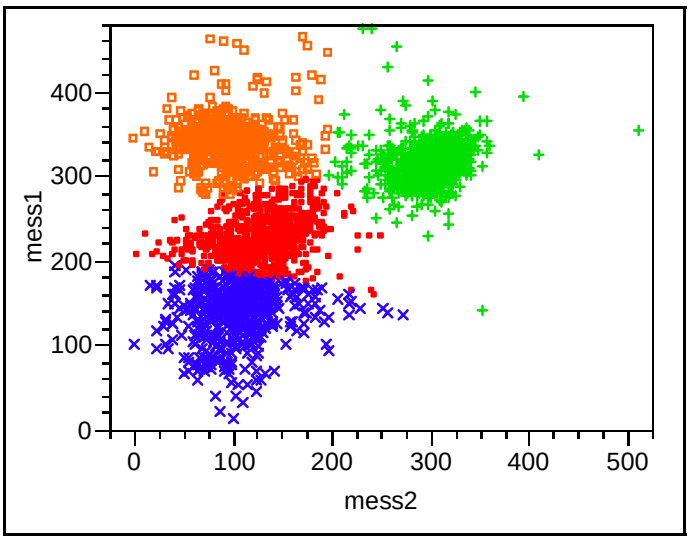

Abbildung 2: Lösung für ein 2-dimensionales Beispiel einer Clusterung
(Farben zeigen gefundene Cluster)

Ein weiteres wesentliches Merkmal der Clusteranalyse ist die gleichzeitige Heranziehung aller vorliegenden Merkmale.

Die Clusteranalyse benötigt eine rechteckige Rohdatenmatrix in folgender Form:

Tabelle 1: Beispiel einer Rohdatenmatrix

	Variable 1	Variable 2	...	Variable J
Fall 1 Fall 2 ... ... Fall K				

Die Weiterverarbeitung dieser Daten ist abhängig vom eingesetzten Verfahren. Zu unterscheiden sind hierarchische (vergl. 2.3.1) und partitionierende (vergl.2.3.2) Verfahren.

Bei der Anwendung eines hierarchischen Verfahrens erstellt das Statistikprogramm während des Auswertungsvorgangs eine quadratische Distanz- bzw. Ähnlichkeitsmatrix von der Dimension K x K. Diese ist in der Regel für den Anwender nicht sichtbar, wird aber sehr schnell groß.

Tabelle 2: Beispiel einer Distanzmatrix

	Fall 1	Fall 2	...	Fall K
Fall 1 Fall 2 ... Fall K				

Je nach Statistikprogramm können bei der Erstellung dieser Ähnlichkeitsmatrix verschiedene Proximitätsmaßes (vergl. 2.1) eingesetzt werden.

Die Größe dieser Ähnlichkeitsmatrix wird durch die Größe des verfügbaren Arbeitsspeichers im Computer begrenzt. Darüber hinaus vermindert sich die Arbeitsgeschwindigkeit bei sehr großen Datensätzen erheblich.

Mit Hilfe von Fusionsalgorithmen bzw. Clusterverfahren (vergl. 2.3) werden nun aus der Distanz- bzw. Ähnlichkeitsmatrix Gruppen gebildet. Auch an diese Stelle kann je nach eingesetztem Statistikprogramm zwischen verschiedenen Verfahren gewählt werden. im Abschnitt 2.3 werden einige dieser Verfahren beschrieben.

Partitionierenden Verfahren wie das in Abschnitt 0 beschriebene k-means-Verfahren gehen einen anderen Weg. Bei diese Verfahren wird ausgehend von einer vorzugebenden Anzahl von Clustern ein Optimierungsprozess gestartet. Einzelne Fälle werden zwischen den Clustern so lange verschoben, bis ein vorgegebenes Optimum erreicht wird.

2.1 Proximitätsmaße

Das Proximitätsmaß überprüft für jeweils zwei Fälle die Ausprägung der verschiedenen Merkmale und versucht durch einen Zahlenwert die Unterschiede bzw. Ähnlichkeiten zwischen diesen Fälle zu messen. Die berechnete Zahl symbolisiert die Ähnlichkeit der Fälle hinsichtlich der untersuchten Merkmale. Die Berechnung des Proximitätsmaßes kann durch unterschiedliche Verfahren vorgenommen werden. Nicht jede Statistiksoftware lässt dem Anwender die Wahl zuwischen verschiedenen Maßen. Trotzdem soll ein kurzer Überblick (vergl. Abbildung 3) über mögliche Verfahren gegeben werden. Einige Relevante werden unten näher beschrieben.

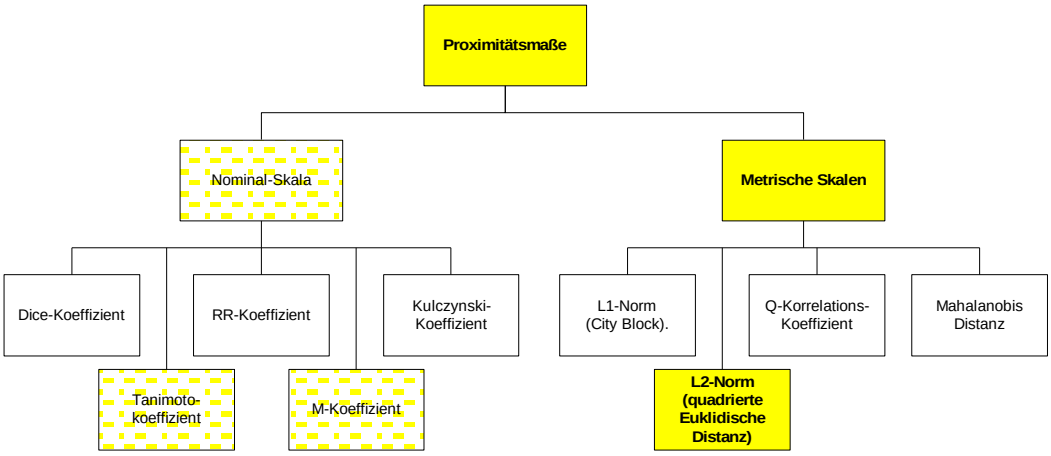

Abbildung 3: Übersicht über verschiedene Proxitätsmaße nach Skalenniveau

2.1.1 Nominal-Skala

Als Proximitätsmaße für Nominalskalen seien Tanimoto- und M-Koeffizient genannt. Beide Koeffizienten können nur für binäre Merkmale (0/1; ja/nein) angewendet werden und bilden die Ähnlichkeit der Fälle untereinander ab (0 = absolut unähnlich; 1 = absolut ähnlich). Mehrkategoriale Merkmale müssen erst eine *binäre Transformation* durchlaufen.

Tabelle 3: Beispiel für eine *binäre Transformation*

Var1	Stufe	Transformation in binäre Merkmale
0	1	1 0 0 0
1-5	2	0 1 0 0
6-10	3	0 0 1 0
mehr als 10	4	0 0 0 1

Aus der mehrkategorialen Variable Var1 mit 4 Stufen sind insgesamt 4 binäre Variablen entstanden.

Das Proximitätsmaß zwischen zwei Fällen für alle einzubeziehenden Variablen wird aus der Anzahl Übereinstimmungen dividiert durch die Anzahl aller Variablen gebildet.

Tabelle 4:

	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9	Var 10
Fall 1	1	1	1	1	1	0	0	1	0	0
Fall 2	1	1	1	1	1	0	1	0	0	0

0 = nicht vorhandener Besitz des Merkmals

1 = vorhandener Besitz des Merkmals

Tanimoto-Koeffizient: $5/10 = 0{,}5$

M-Koeffizient: $8/10 = 0{,}8$

Für den Tanimoto-Koeffizienten werden nur Übereinstimmungen des jeweiligen Fallpaares mit vorhandenem Besitz des Merkmals (im obigen Beispiel die 1) im Zähler gewertet. Beim M-Koeffizienten gilt dies ebenso bei nicht vorhandenem Besitz (im obigen Beispiel die 0). Die Anzahl der Übereinstimmung wird dann durch die Gesamtanzahl von Variablen geteilt.

Der Tanimoto-Koeffizient findet Anwendung, wenn *vorhandener Besitz* eines Merkmales in gegensätzlicher Aussage zum Zustand *nicht vorhandener Besitz* steht. Gegensätzliches gilt für den M-Koeffizienten.

2.1.2 Metrische Skalen

Als wichtigstes Proximitätsmaß für metrische Skalen ist die L2-Norm bzw. quadrierte Euklidische Distanz zu nennen. Wie der Name schon ausdrückt wird die Distanz bzw. Unähnlichkeit der Falle zueinander ausgedrückt. Sie wird berechnet, indem man die Beträge der quadrierten Differenzen für alle einzubeziehenden Variablen eines Fallpaares addiert. Das Ergebnis dieser Berechnung ergibt das Distanzmaß. Das nachfolgende Beispiel verdeutlicht die Berechnung der Euklidischen Distanz.

Gegeben ist eine Rohdatenmatrix aus vier Fällen mit drei Variablen. Aus dieser wird mit Hilfe der beschriebenen Rechnung die Distanzmatrix abgeleitet.

Tabelle 5: Rohdatenmatrix

	Var 1	Var 2	Var 3
Fall 1	1	2	1
Fall 2	2	3	3
Fall 3	3	2	1
Fall 4	4	4	7

$$d_{Fall1,Fall2} = (1-2)^2 + (2-3)^2 + (1-3)^2 = 1+1+4 = 6$$

Tabelle 6: Distanzmatrix

	Fall 1	Fall 2	Fall 3	Fall 4
Fall 1	0			
Fall 2	6	0		
Fall 3	4	6	0	
Fall 4	49	21	41	0

2.2 Daten Transformation / Standardisierung

Ein Problem der Euklidischen Distanz ist, dass der Einfluss von Variablen mit hohem Wertebereich zu ungunsten von Variablen mit geringem Wertebereich verstärkt ist. Nachfolgendes Beispiel verdeutlicht dieses.

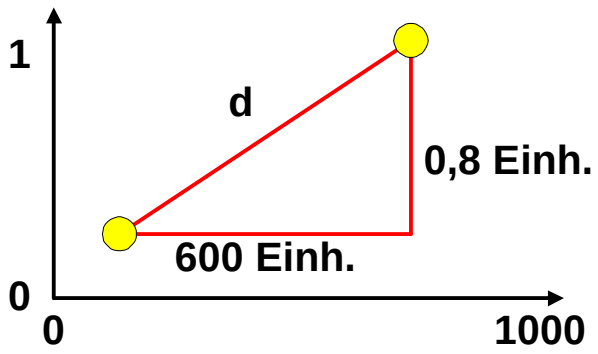

Abbildung 4: Darstellung der Euklidischen Distanz d bei unterschiedlicher Achsendimension

Die Euklidische Distanz zwischen den beiden Punkten errechnet sich wie folgt:

$$d_{(i,j)} = \sqrt{\left(600^2 + 0{,}8^2\right)} = \sqrt{\left(36000 + 0{,}64\right)} = \sqrt{600{,}0007}$$

Die Distanz ist offensichtlich fast ausschließlich durch die Variable der x-Achse dominiert. Eine Lösung für dieses Problem ist die Standardisierung der Variablen. Wenn also für beide Variablen eine Skala von 0 bis 1 angenommen wird, haben die beiden Distanzen einen Wert von 0,6 und 0,8. Beide Variablen sind somit recht gleichgewichtig. Die Euklidische Distanz errechnet sich dann wie folgt:

$$d_{(i,j)} = \sqrt{\left(0{,}6^2 + 0{,}8^2\right)} = \sqrt{\left(0{,}36 + 0{,}64\right)} = \sqrt{1}$$

Somit haben beide Variablen signifikant zum Wert der Distanz beigetragen.

2.3 Clusterverfahren / Fusionsalgorithmen

Nachdem die Ähnlichkeit bzw. Unähnlichkeit von Fällen bestimmt wurde ist die nächste Aufgabe diese Fälle zu gruppieren. Diese geschieht mit Fusionsalgorithmen bzw. Clusterverfahren. Je nach Algorithmus kann das Ergebnis der Clusterung bei gleichem Datensatz unterschiedlich ausfallen. Darum ist es wichtig grundsätzliche Kenntnis über die Verschiedenen Verfahren und ihre Vorzüglichkeit zu haben. Die unten gezeigte Abbildung gibt einen systematischen Überblick über die Verfahren. Einige wichtige werden unten ausführlicher beschrieben. Im Abschnitt 5.1 sind Beispiele für die Anwendung und Ergebnisse der Verfahren zu finden. Zu bemerken ist, dass die Auswahl der Verfahren je nach Statistikprogramm durchaus eingeschränkt ist.

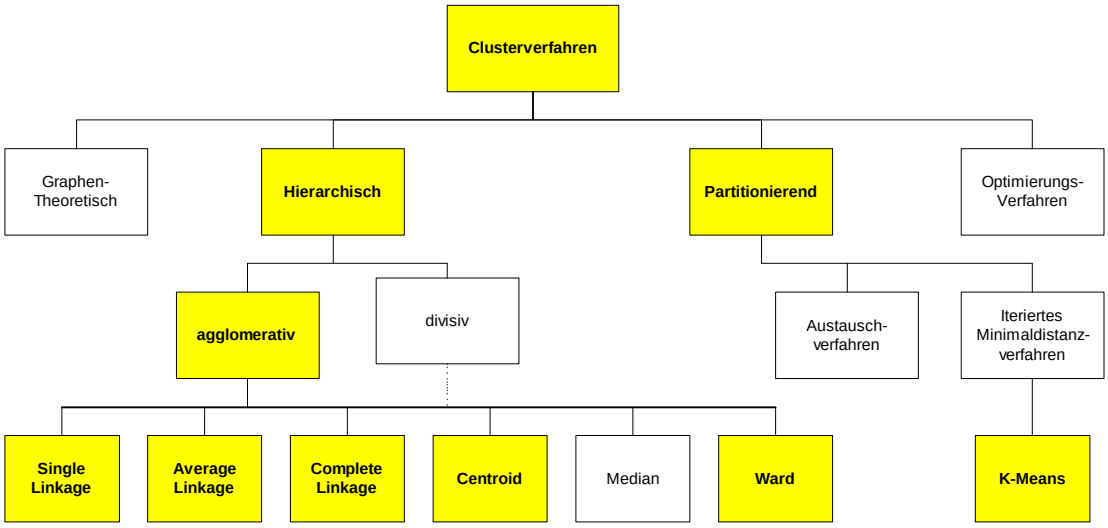

Abbildung 5: Übersicht über verschiedene Clusterverfahren nach Herangehensweise

2.3.1 Hierarchisches Clustering

Unterschieden wird zwischen einer agglomerativen und einer divisiven Herangehensweise.

	Zahl der Cluster am Anfang	Zahl der Cluster am Ende
agglomerativ	n	k
divisiv	1	K

Somit lässt sich der Ablauf der ersten Verfahrensart durch die Zusammenfassung von Gruppen und der Ablauf der zweiten Verfahrensart durch die Aufteilung einer Gesamtheit in Gruppen charakterisieren. Das divisive Verfahren wird wegen seiner geringen Bedeutsamkeit nicht weiter beachtet. Die unten beschriebenen Cluster-Algorithmen könnten für beide Herangehensweisen angewendet werden.

Bei der agglomerativen hierarchischen Clusterung werden im ersten Schritt an Hand der Distanzmatrix die Objekte ermittelt, die die kleinste Distanz aufweisen. Die gefundenen Objekte werden zur Startgruppe zusammengefasst. Die weitere Fusionierung hängt dann vom eingesetzten Verfahren ab. Einige dieser Verfahren werden nachfolgend beschrieben. Eine Gemeinsamkeit der Verfahren ist, dass die einmal vorgenommene Zuordnung eines Datensatzes zu einem Cluster während eines Auswertungsdurchgangs unveränderlich bleibt. Damit unterscheiden sich die Hierarchischen Verfahren grundsätzlich von dem unter Abschnitt 2.3.2 beschrieben Verfahren.

2.3.1.1 Single Linkage (Nearest-Neighbour-Verfahren / Nächstgelegener Nachbar)

Beim single linkage-Verfahren ist häufig eine sogenannte Kettenbildung zu beobachten. Die Verschiedenheit zweier Cluster wird über die kleinste Unähnlichkeit von je einem Objekt der beiden Cluster (vergl. Abbildung) bewertet. Es kann aufgrund von Brückenobjekten zu langgestreckten Clustern kommen. Diese Cluster können dann auch einzelne weit auseinander liegende Objekte enthalten. Das Verfahren ist auf die Bildung von wenigen großen Clustern optimiert und kann auch zur Isolierung von Ausreißern (d.h. Objekte, die eine große Unähnlichkeit zu allen anderen Objekten aufweisen) eingesetzt werden. Diese werden erst in den letzten Iterationsschritten an eine große Klasse angehängt und sind als solche aus dem Dendrogramm (vergl. Abschnitt 2.4) relativ leicht ablesbar.

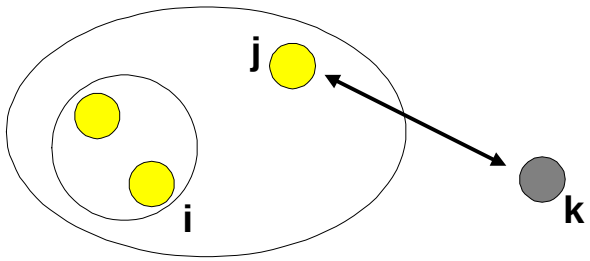

Abbildung 6: Schema des Single-Linkage-Verfahrens

2.3.1.2 Complete Linkage (Furthest-Neighbour-Verfahren / Entferntester Nachbar)

Das Complete-Linkage-Verfahren neigt zur Bildung kleiner Gruppen, da die Bewertung der Verschiedenheit zweier Klassen im Gegensatz zum Single-Linkage-Verfahren durch die maximale Distanz zwischen zwei Punkten erfolgt. Die Distanz zwischen zwei Clustern ist somit definiert als die längste Distanz zwischen einem Punkt im ersten und einem Punkt im zweiten Cluster (vergl. Abbildung). Ausreißer können mit diesem Verfahren nicht identifiziert werden, sie führen eher zu einer Verzerrung der Ergebnisse. Es empfiehlt sich daher, vor Anwendung des complete linkage-Verfahrens das single linkage-Verfahren anzuwenden und die so identifizierten Ausreißer zu eliminieren.

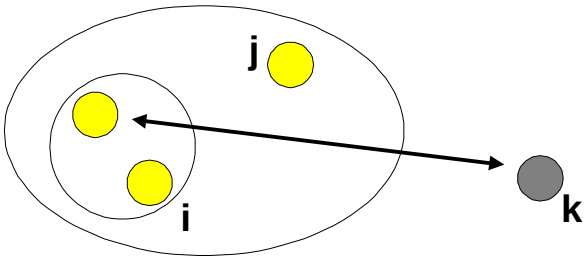

Abbildung 7: Schema des Complete-Linkage-Verfahrens

2.3.1.2.1 Centroid

Bei der Centroid-Methode wird die Distanz zwischen den Clustern aus der quadratischen Euklidischen Distanz ihrer Mittelwerte ermittelt. Das Centroid-Verfahren ist wesentlich robuster gegenüber Ausreißern als die bisher beschriebenen Verfahren. Es hat diesbezüglich jedoch immer noch Defizite gegenüber dem Average-Linkage- und dem Ward-Verfahren.

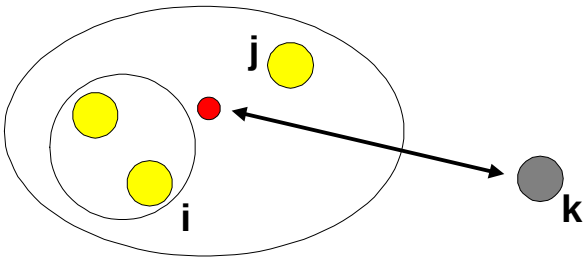

Abbildung 8: Schema des Centoid-Verfahrens

2.3.1.3 Average Linkage

Dieses Verfahren bestimmt die Distanz zwischen einem Element und einem Cluster als durchschnittliche Distanz (bzw. Ähnlichkeit) zwischen diesem Element und allen Elementen, die das Cluster bilden. Beim Average-Linkage-Verfahren kommt es zur Bildung von Clustern mit kleinen und nahezu gleichen Varianz (innerhalb des Clusters).

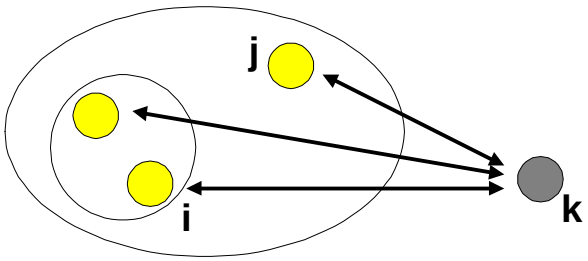

Abbildung 9: Schema des Average-Linkage-Verfahrens

2.3.1.4 Ward

Das Ward-Verfahren zeigt keine der bei den anderen Verfahren erwähnten Anomalien (Kettenbildung, Bildung kleiner Gruppen). Eine Besonderheit des Ward-Verfahren besteht darin, daß es als einziges der genannten Verfahren einen vorgegebenen Güteindex (Varianzkriterium) direkt optimiert. In jedem Iterationsschritt des Ward-Verfahrens werden die Klassen so zusammengefaßt, dass eine möglichst geringe Verschlechterung des Güteindexes vorliegt. Das Ward-Verfahren führt zur Bildung etwa gleich großer Klassen. Eine Erkennung von Ausreißern ist mit dem Ward-Verfahren ebenso wie mit dem Complete Linkage-Verfahren nicht möglich.

2.3.1.5 Ein kurzer Überblick über die genannten Hierarchischen Clusterverfahren

Tabelle 7: Überblick über die genannten Clusterverfahren

Verfahren	Eigenschaft	Monoton?	Bemerkung
Single Linkage	kontrahierend	ja	Kettenbildung
Complete Linkage	dillatierend	ja	kleine Gruppen
Centroid	konservativ	nein	
Average Linkage	konservativ	ja	
Ward	konservativ	ja	gleich große Gruppen

kontrahierend	=	Objekte werden verstärkt unterschiedlich großen Gruppen zugeordnet – Ausreißer erkennbar
dillatierend	=	Objekte werden verstärkt in etwa gleich große Gruppen zusammengefasst
konservativ	=	keine der oben genannten Tendenzen
Monoton ja	=	Heterogenitätsmaß (vergl. 2.4) steigt bei Verringerung der Gruppenanzahl monoton an
Monoton nein	=	Heterogenitätsmaß kann bei Verringerung der Gruppenanzahl auch abfallen an

2.3.2 K-Means Clustering

Das K-Means-Verfahren benötigt als Vorgabe die Anzahl von Partitionen in die der Datensatz geteilt werden soll. Diese Partitionsanzahl ist eine feste Vorgabe für einen Auswertdurchgang. Diese Clusteranzahl bleibt über den gesamten Prozess gleich. Die Zugehörigkeit eines Datensatzes zu einer Partition ist jedoch variabel. Während des Auswertvorgangs werden die Datensätze nach verfahrensbedingten Regeln zwischen den Partitionen verschoben bis ein vorgegebenes Optimum erreicht ist. Dieses ist ein iterativer alternierender Prozess. D.h. während der einzelnen Annäherungsschritte an das Optimum können Datensätze auch mehrfach die Partitionszugehörigkeit wechseln. Damit unterscheiden sich die Hierarchischen Verfahren grundsätzlich von dem unter Abschnitt 2.3.1 beschrieben Verfahren.

Am Anfang des iterativen alternierenden Prozesses steht die Bildung von *n* Startclustern (seed cluster). Diese Startcluster entstehen entweder durch *n* Zufallsauswahlen oder aber indem *n* Startcluster um *n* Dichtzentren gebildet werden. Danach werden den gebildeten Startclustern in weiteren Schritten Datensätze hinzugefügt. Dabei gelangt ein Verfahren zur Anwendung ähnlich dem unter Abschnitt 2.3.1.2.1 vorgestellten Centroid-Verfahen. Der Erfolg eines Zuordnungsschritts wird durch die Berechnung eines Streuungsmaßes für das neu gebildete Cluster geprüft. Dieses Verfahren wird so lange fortgeführt bis alle Datensätze zugeordnet sind (vergl. Abbildung 10).

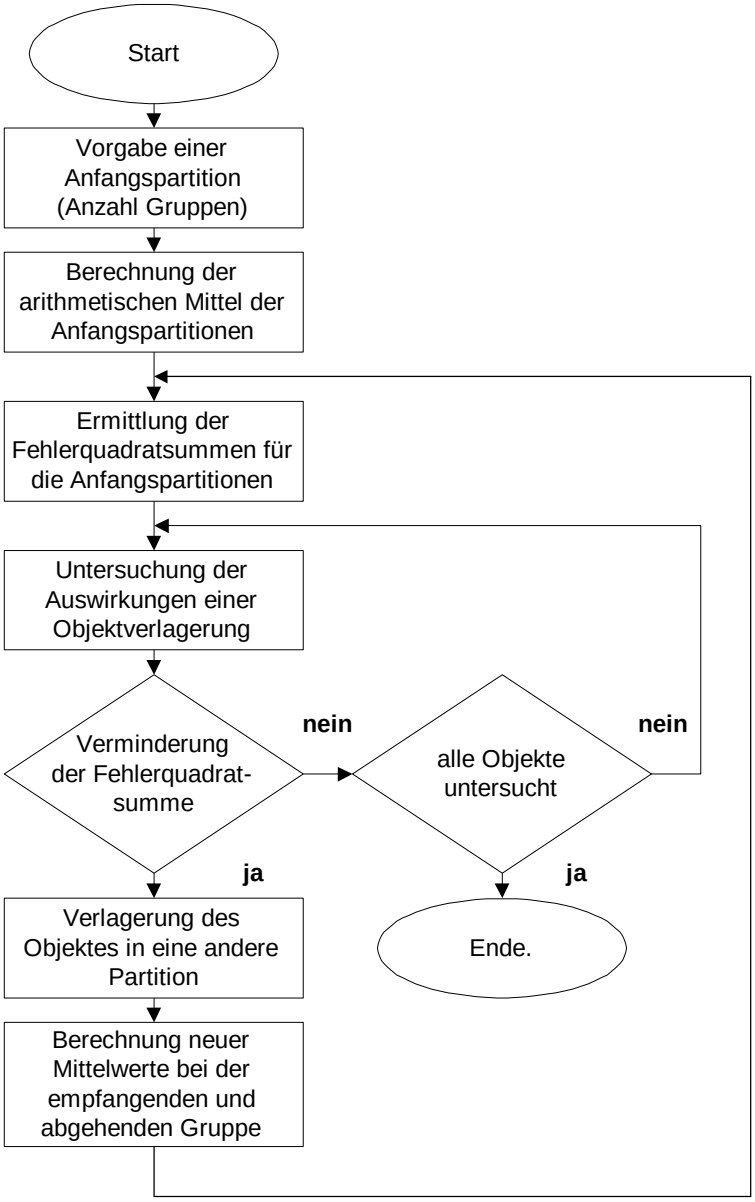

Abbildung 10: Ablaufschema des K-Means-Verfahrens
(nach Backhaus et al.: Multivariate Analysemethoden)

Das K-Means-Verfahren bietet sich für große Datensätze an, da keine speicherintensive Datenmatrix (vergl. Tabelle 2) benötigt wird. Darüber hinaus können mit dem K-Means-Verfahren die Ergebnisse der hierarchischen Clusteranalyse überprüft werden.

K-Means hat zwei Nachteile.

Es ist nicht immer möglich eine sinnvolle Festlegung der Anzahl von Startpartitionen zu treffen. Die Zahl der Startpartitionen muss meist relativ willkürlich festgelegt werden. Sinnvoll erscheint dann das Experimentieren mit unterschiedlicher Anzahl von Startpartitionen.

Außerdem besteht die Gefahr, dass das K-Means-Verfahen beim Erreichen lokaler Optima des Streuungskriteriums stoppt. Dieses kann eintreten, da nicht alle möglichen Kombinationen von Datensätzen und deren Zugehörigkeit zu den einzelnen Partitionen geprüft werden. Dieses erklärt sich dadurch, dass sich für *g* Gruppen und *m* Datensätze je Gruppe

g^m Kombinationen ergeben. In einem Fall von 4 Gruppen und 10 Objekten je Gruppe ergeben sich z.B. 4^{10} = 1.048.576 Kombinationen und damit Clustermöglichkeiten.

2.4 Die Darstellung von Clustern / Ergebnisse der Clusteranalyse

Im **Streudiagramm** / Scatterplot (vergl. Abbildung 11) werden zwei Variablen gegeneinander geplottet. Ein Datenpunkt markiert die Werte der zwei abgebildeten Variablen für einen Datensatz. In vielen Statistikprogrammen können die Datenpunkte im Scatterplot mit Hilfe von kategorialen Variablen eingefärbt werden. In JMP können gefundene Cluster (vergl. Abbildung 12) als kategoriale Variable für die Darstellung des Scatterplots verwendet werden.

Das **Dendrogramm** (vergl. Abbildung 12) visualisiert den Fusionierungsprozess einer Hierarchischen Clusteranalyse. Die Ziffern im Diagramm sind die Fallnummern.

Jeder Fusionierung wird durch die senkrechte Verbindung von zwei Datenpunkten dargestellt. In weiteren Fusionsschritten werden dann auch vorhandene Cluster mit weiteren Datenpunkten bzw. anderen Clustern fusioniert. Diese Fusionen werden ebenso durch senkrechte Verbindungen markiert. Dem Dendrogramm wird als Hilfsdiagramm (vergl. Abbildung 12 – blauer Graph) ein Plot des Heterogenitätsmaß zur Seite gestellt. Dieser Graph kann bei der Wahl der Clusteranzahl als Hilfe dienen. Je größer das Heterogenitätsmaß (Fehlerquadratsummen) wird desto unähnlicher werden die Datensätze innerhalb eines Cluster und desto gleicher werden die Cluster untereinander.

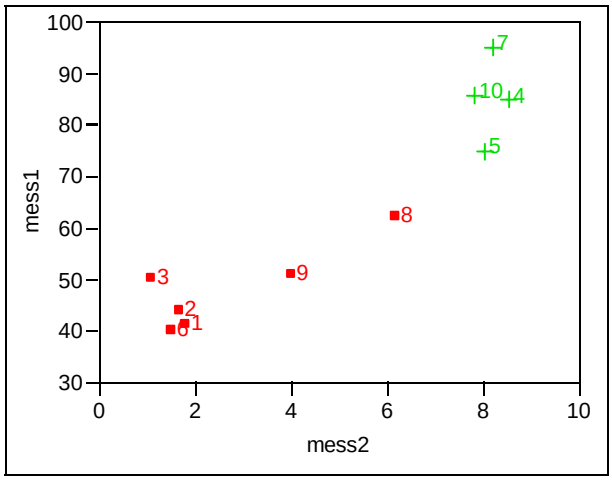

Abbildung 11: Streudiagramm / Scatterplot

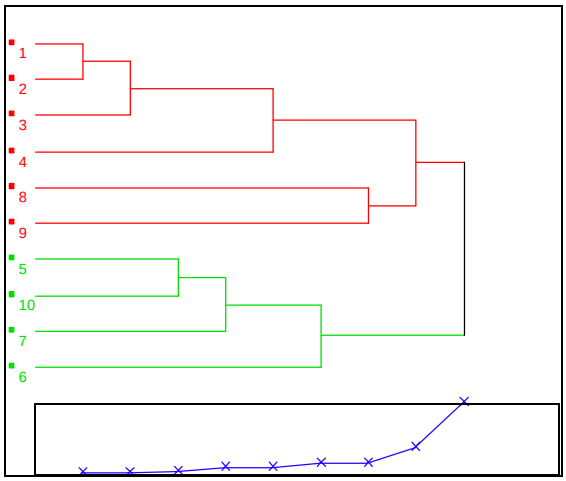

Abbildung 12: Dendrogramm

Ein Scatterplot kann nur zwei Variablen darstellen. Dieses Problem wird durch eine **Scatterplot Matrix** (vergl. Abbildung 13) gelöst. Der *n*-dimensionale Raum, gebildet durch *n* Variablen wird durch eine Matrix ähnlich einer Korrelationsmatrix zweidimensional aufgelöst. Jede Variable wird gegen jede anderen Variablen abgebildet. Die Lage der einzelner Cluster kann durch die Einfärbung der Datenpunkte wie beim einfachen Scatterplot visualisiert werden. Darüber hinaus werden Dichteellipsen für die gesamte Punktewolke eines Variablenpaares dargestellt.

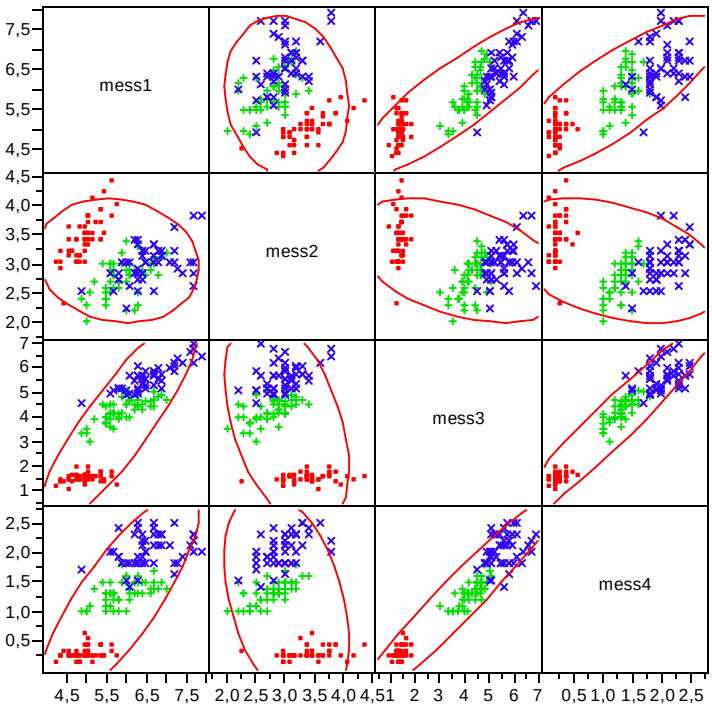

Abbildung 13: Scatterplot-Matrix

Beim Biplot (vergl. Abbildung 14) wird im Gegensatz zur Scatter-Matrix der n-dimensionale Raum durch Achsenverschiebungen zweidimensional projiziert. Die Achsen des n-dimensionalen Raumes werden auf die beiden Hauptkomponenten (Variablen, die die größte Streuung erklären) reduziert. Diese Darstellungsform ist typische für die K-Means-Clusteranalyse. Die Nummern und Kreise in der Abbildung markieren die Centroide der Startcluster eines K-Means-Durchlaufs.

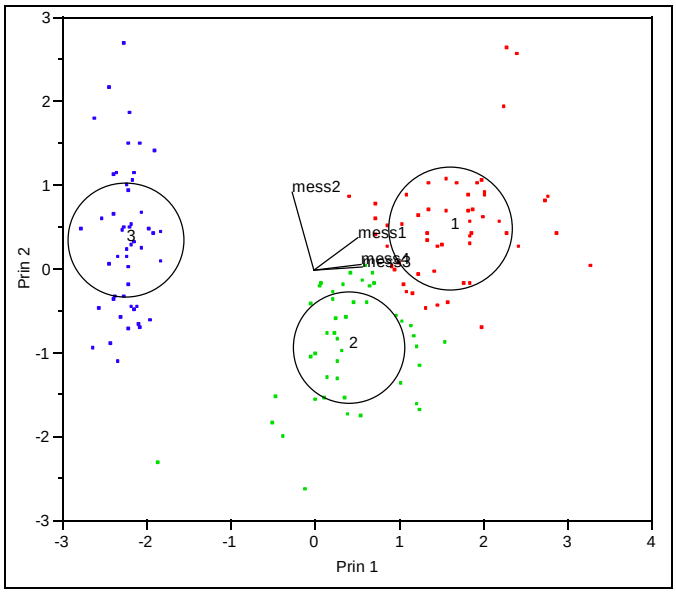

Abbildung 14: Biplot

Im Gegensatz zum Biplot ist die Interpretation des **Parallel Coord Plots** (Parallelkoordinaten-Plot) recht einfach. Es ist ebenso ein typisches Darstellungswerkzeug der K-Means Analyse.

Für jedes Cluster wird ein Plot erstellt. Dieser Plot zeigt auf der X-Achse die einzelnen für die Clusteranalyse herangezogenen Variablen als Kategorie und als Graph den Wert eines jeden Datensatzes für die jeweiligen Variablen. Die Daten werden ohne Skalen und unstandardisiert abgebildet. Für jede Variabel der Kategorieachse wird die Spannweite der Daten als Min und Max gesetzt. Weiterhin werden in einem extra Plot die Mittelwerte für die einzelnen Variablen über alle Cluster abgebildet.

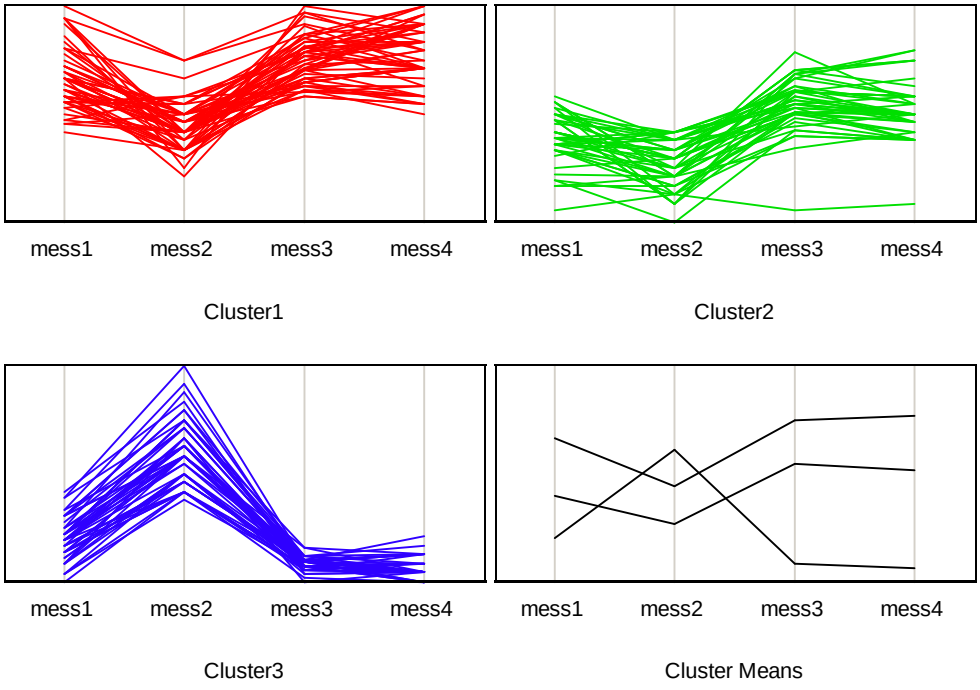

Abbildung 15: Parallel Coord Plots

3 Decision Trees / Classification Tree / Regression Trees

Tree-Verfahren sind rekursiv partitionierende Verfahren. Dargestellt wird die Beziehung von einzelnen oder mehreren Einflussgrößen (X-Variablen) auf eine abhängige (Y-)Variable (response variable). Einflussgrößen und abhängige Variablen können jegliches Skalenniveau (z.B. kategorial oder kontinuierlich) und auch Mischungen haben. Das Ergebnis der Partitionierung wird als Baum visualisiert. Partitionen werden dabei durch Blätter symbolisiert und die Äste entsprechen den Partitionierungsschritten. Diese Baumstruktur kann auch als Entscheidungsbaum bezeichnet bzw. genutzt werden.

Ein großer Vorteil von Entscheidungsbäumen ist, dass sie gut erklärbar und nachvollziehbar sind. Dies erlaubt dem Benutzer, das Ergebnis auszuwerten und Schlüsselattribute zu erkennen. Dies ist vor allem nützlich, wenn die Qualität der Daten nicht bekannt ist und auch sonst kein Ansatz für ein parametrisches Modell erkennbar ist. Die Regeln selber können ohne großen Aufwand in eine einfache Abfragesprache wir SQL übernommen werden. Das Verfahren ist in der Lage Ausreißer zu erkennen und sie in einem eigenen Blatt zu separieren. Große Datensätze können sehr schnell bearbeitet werden und das Ergebnis des Tree-Verfahrens kann zur Verbesserung von parametrischen Modellen dienen. Es ist jedoch nicht trivial einen Baum in ein parametrisch Modell umzuwandeln.

Bei X-Variablen mit einem kontinuierlichen Charakter erfolgt die Partitionierung durch Festlegung eines Trennwerts. Fälle mit einem größeren bzw. kleineren Wert als dem Trennwert bilden eine eigene Partition. Bei kategorialen X-Variablen erfolgt die Partitionierung durch Aufteilung der verschiedenen Kategorien in zwei Gruppen von Kategorien. Ordinale Variablen werden ebenso wie kategoriale Variablen behandelt.

Für Y-Variablen mit kontinuierlichem Charakter dient die Veränderung der Quadratsummen und für kategoriale und ordinalen Variablen die Veränderung des chi-quadrat-Wertes als Messgröße während des Splitvorgangs.

Das Tree-Verfahren läuft nach Auswahl einer abhängigen (response, X) Variablen und der unabhängigen X-Variablen automatisch bzw. in auslösbaren Schritten ab. Dabei entsteht die Schwierigkeit den Punkt zu finden an dem ein „optimaler“ Baum entstanden ist. Die „Optimalität“ des Baums lässt sich über die Gütekriterien vollständig, korrekt und möglichst klein definieren. Eine weitere Schwierigkeit ergibt sich aus der Tatsache, dass ein vollständiger Baum evtl. zu einem „Overfitting“ d.h. zu einer Überanpassung führt. Das Baummodell speichert in einem solchen Fall die Eigenarten des Trainingsdatensatzes und ist somit für den allgemeinen Fall innerhalb eines Produktionsprozesses nicht sinnvoll anwendbar. Der Baum muss evtl. manuell verändert werden.

Softwarepakete, die Tree-Verfahren im Funktionsumfang haben, liefern statistische Maßzahlen und Werkzeuge, die bei der manuellen Baumbildung assistieren.

Als erstes ist eine Maßzahl zur Beurteilung der Güte des Gesamtmodells zu nennen. Die Güte wird oft als Bestimmtheitsmaß r^2 ausgedrückt. Die Veränderung des Koeffizienten kann für jeden Partitionierungsschritt abgelesen werden. Der Wertebereich reicht von 0 bis 1, wobei der Wert 1 ein „optimales“ d.h. das bestmögliche Modell anzeigt.

Der automatische Split-Vorgang wählt immer den aktuell besten Split. Ein Split kann jedoch auch explizit an einer bestimmten Stelle im Baum ausgelöst werden. Weiterhin kann auch ein Split an einer bestimmten Stelle des Baums mit einer bestimmten Variablen veranlasst werden. Umgekehrt können Splits auch rückgängig (pruning) gemacht werden. Dieses kann ebenso automatisch und manuell an bestimmten Stellen erfolgen.

Für den Splitvorgang stehen in der Regel verschiedene Kriterien zur Auswahl: **Maximize Split Statistic** und **Maximize Significance**. Das Kriterium **Maximize Split Statistic** führt zur Bildung von eher gleich große Gruppen mit hohem Spliterfolg. Das Kriterium **Maximize Significance** führt in der Regel zu Gruppen die sich signifikant voneinander unterscheiden. Die Gruppengröße spielt dabei eine untergeordnete Rolle. Bei beiden Kriterien werden Ausreißer relativ schnell aussortiert.

4 Neuronale Netze

Neuronale Netze beziehen sich auf die Strukturen des Gehirns von Tieren und Menschen: Neuronen sind in der Art eines Netzes miteinander verknüpft. Biologische Neuronen reagieren auf elektrische oder chemische Reize. Neuronen haben üblicherweise mehrere Eingangsverbindungen sowie eine Ausgangsverbindung. Wenn die Summe der Eingangsreize einen gewissen Schwellenwert überschreitet, "feuert" das Neuron. Das bedeutet, daß ein Aktionspotential an seinem Axonhügel ausgelöst und entlang seines Axons weitergeleitet wird. Das ist das Ausgangssignal des Neurons. Über das Lernen in neuronalen Netzen gibt es verschiedene Theorien. Die erste neuronale Lernregel wurde von Hebb beschrieben (Hebb'sche Lernregel). Er formulierte 1949: „Wenn ein Axon der Zelle A...Zelle B erregt und wiederholt und dauerhaft zur Erzeugung von Aktionspotentialen in Zelle B beiträgt, so resultiert dies in Wachstumsprozessen oder metabolischen Veränderungen in einer oder in beiden Zellen, die bewirken, dass die Effizienz von Zelle A in Bezug auf die Erzeugung eines Aktionspotentials in B größer wird.“ Das bedeutet: Je häufiger ein Neuron A gleichzeitig mit Neuron B aktiv ist, umso bevorzugter werden die beiden Neuronen aufeinander reagieren ("what fires together that wires together"). Dies hat Hebb anhand von Veränderungen der synaptischen Übertragung zwischen Neuronen nachgewiesen.

Künstliche neuronale Netze (KNN) versuchen sich als Teilgebiet der Künstlichen Intelligenz an der computergestützten Simulation neuronaler Netze. Bis jetzt beschränkt sich die Anwendung von künstlichen neuronalen Netzen auf die Regelung von komplexen Prozessen z.B. in der Chemieindustrie, Mustererkennung, Sprachanalyse und anpassungsfähiger Software wie Virtuelle Agenten und KI-Robotern in Spielen.

Neuronale Netze haben im Vergleich zu anderen Gebieten der künstlichen Intelligenz ihren Anwendungsschwerpunkt immer genau dort, wo ein Computer etwas lernen soll bzw. auch durch einige ausgewählte Eingabemuster auf eine allgemeinere, abstraktere Form des Musters schließen können soll ("Generalisierung", z. B. Bild-/Gesichtserkennung).

Auch bei der Vorhersage von Veränderungen in komplexen Systemen werden KNNs unterstützend hinzugezogen, z.B. zur Früherkennung sich abzeichnender Tornados oder aber auch zur Abschätzung der weiteren Entwicklung wirtschaftlicher Prozesse.

Die beiden Hauptnachteile von KNNs sind

1. Das Trainieren von KNNs führt zu hochdimensionalen, nichtlinearen Optimierungsproblemen. Diese müssen i.allg. heuristisch gelöst werden. In der Praxis ist häufig nicht klar, ob man der global optimalen Lösung nahe kommt oder nicht.
2. KNNs neigen dazu, die Trainingsdaten einfach auswendig zu lernen (Overfitting). Wenn dies geschieht, können sie nicht mehr auf neue Daten verallgemeinern. Um Overfitting zu vermeiden, muß die Netzwerkarchitektur sehr vorsichtig gewählt werden. In der Praxis muss man viele Architekturen ausprobieren und hoffen, dass eine gut funktioniert.

5 Durchführung mit JMP an Hand von Beispielen

5.1 Clusteranalyse

5.1.1 Grundsätzliche Vorgehensweise

Nachdem eine Datei geöffnet wurde kann die Clusteranalyse über das Menü *Analyze* > *Multivariate Methods* > *Cluster* oder vom JMP Starter aus über den Reiter *Multivariate* erreicht werden. Es erschein eine Dialogbox (vergl. Abbildung 16) mit dem Titel *Clustering*.

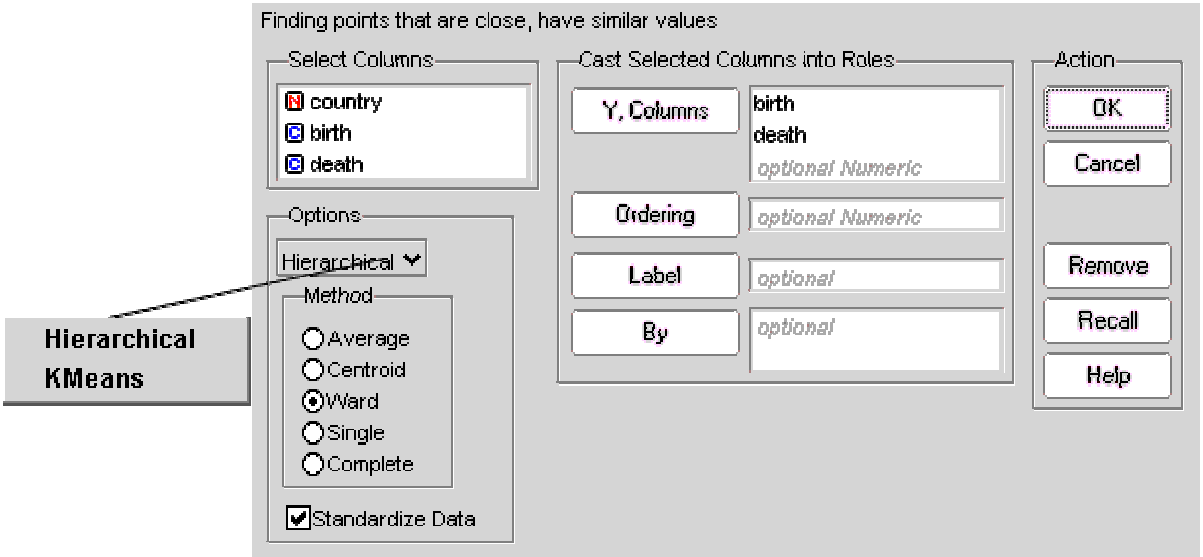

Abbildung 16: Dialogbox Clusteranalyse

Es können beliebig viele Y-Variablen zur Auswertung angewählt werden. Dieses erfolgt dadurch, dass entsprechende Variablen im Feld *Select Columns* mit der Maus angewählt werden und durch einen Klick auf den Button *Y,Columns* in das entsprechende Feld befördert werden. Die Y-Variablen müssen numerisch sein. Danach kann unter *Options*

zwischen *Hierarchical* und *K-Means clustering* gewählt werden. Je nach dem welche Option gewählt wurde ändert sich der Optionsbereich der Dialogbox.

Bei der Auswahl von Hierarchischer Clusteranalyse(vergl. 2.3.1 und folgende) kann weiterhin ein Fusionsalgorithmus gewählt werden.

Bei der Auswahl des K-Means-Verfahren (vergl. 2.3.2) muss die Anzahl der Cluster festgelegt werden.

Die Clusteranalyse wird in JMP mit der Vorgabe *Standardize Data* (vergl. 2.2) durchgeführt. Dieses kann durch Abwahl geändert werden.

Mit Klicken von *OK* startet die Auswertung. Das Ergebnisfenster sieht je nach verwendeter Methode unterschiedlich aus. Weitere Erläuterungen sind in den nächsten Abschnitten zu finden.

5.1.2 Beispiel für eine Hierarchisches Clusteranalyse

Das Beispiel zeigt Daten einer Messreihe zweier Messstellen (Datei: bsp2.jmp). Da der Datensatz nur zwei Variablen enthält, kann ein Scatterplot mit *Analyse > Fit Y by X* erstellt werden. Das Ergebnis sollte so aussehen:

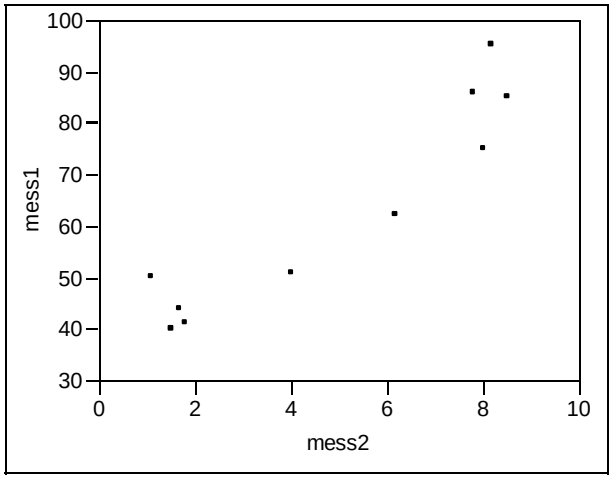

Abbildung 17: Scatterplot des Datensatzes bsp2.jmp

Der Scatterplot zeigt scheinbar zwei Clusterungen (unten links und oben rechts) und zwei Datensätzen die eine Brücke zwischen den Clustern bilden.

Als Parameter für die Clusteranalyse sollten folgende Parameter eingestellt werden:

- Y,Columns = mess1 und mess2
- Hierachical Clustering
- Methode = Single
- Standardize Data

Das Ergebnis sollte folgenden Abbildungen entsprechen. Falls die farbliche Markierung der Cluster nicht sichtbar ist kann diese erzeugt werden, in dem im Titelfeld *Hierarchical Clustering* die rechte Maustaste gedrückt wird und die Option *Color Clusters* gewählt wird. Die Verschiedenen Fenster (in diesem Fall *Hierarchical Clustering*, *Bivariate Fit of..* und die

Rohdatenmatrix) sind interaktiv miteinander Verknüpft. Durch ein Klick auf einen Datenpunkt des Scatterplots wird der gewählte Datenpunkt auch im Dendrogramm und in der Rohdatenmatrix sichtbar. Eine andere Reihenfolge ist ebenso möglich.

Das Single-Linkage-Verfahren zeigt das unter 2.3.1.1 beschriebene Verahalten. Der Datenpunkte 8 wird als letzter in das Cluster mit den Kreuzen aufgenommen und stellt sich somit evtl. als Ausreißer dar. Ähnliches gilt für den Datenpunkt 9 (im Scatterplot *südwestlich* von Punkt 8) im Cluster mit den Klötzchen.

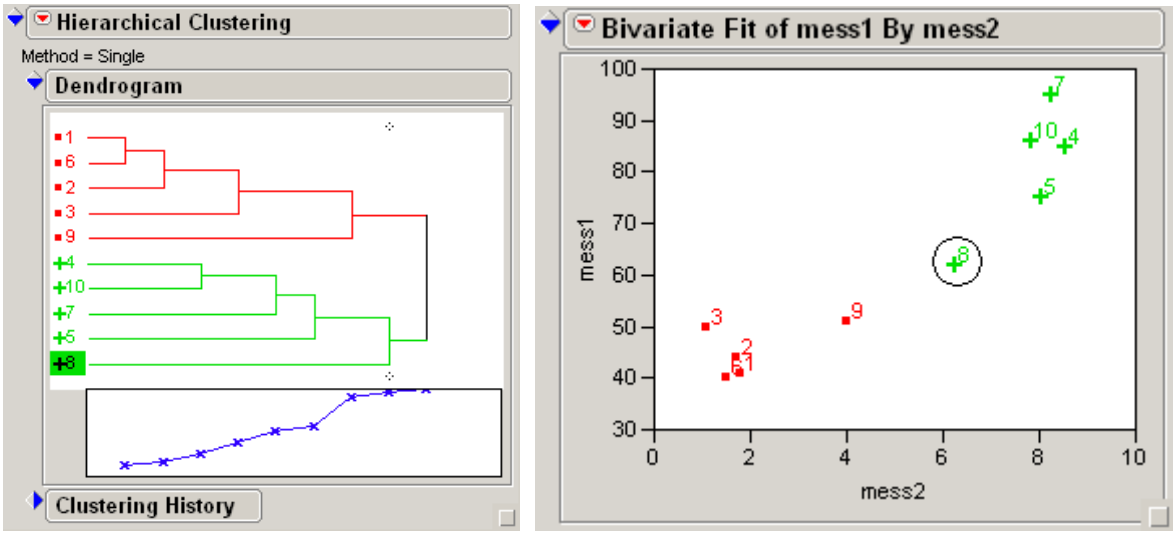

Abbildung 18: Dendrogramm, Methode *Single* (links) und Scatterplot (rechts); Datensatz 8 ist markiert

Im Gegensatz dazu verhält sich das Ward-Verfahren (vergl. 2.3.1.4).

- Y,Columns = mess1 und mess2
- Hierachical Clustering
- Methode = Ward
- Standardize Data

Datenpunkt 8 und Datenpunkt 9 werden erst in ein gemeinsames Cluster aufgenommen und dann in ein größeres Cluster (das mit den Klötzchen).

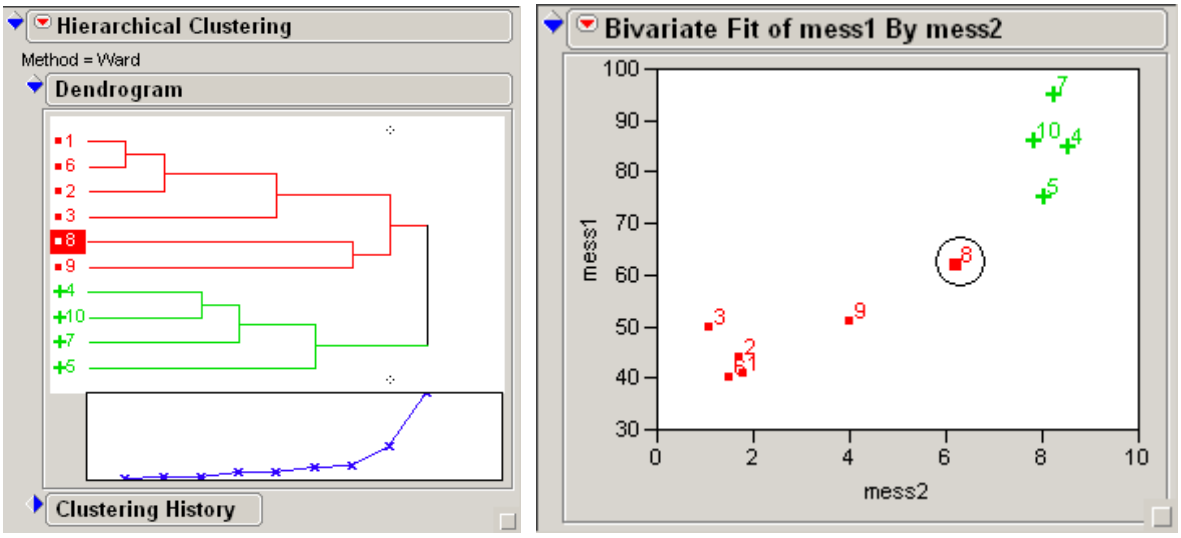

Abbildung 19: Dendrogramm, Methode *Ward* (links) und Scatterplot (rechts); Datensatz 8 ist markiert

Im diesem Beispiel scheint es sinnvoll die Clusteranzahl zu variieren. Dieses ist interaktiv möglich.

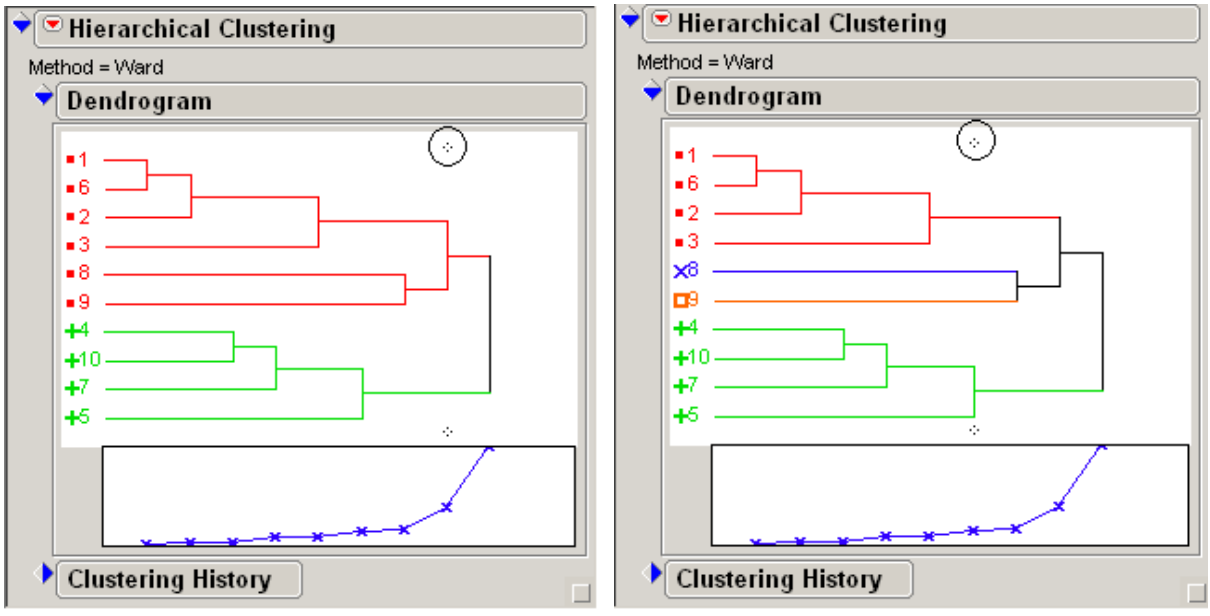

Abbildung 20: Veränderung der Clusteranzahl durch Verscheibung des *Diamanten*

Um die Clusterzahl zu variieren kann der in Abbildung 20 eingekreiste, innerhalb von JMP als Diamant bezeichnete Punkt mit der Maus (linke Taste halten) verschoben werden. Abbildung 20 zeigt von links nach rechts die Änderung von 2 auf 4 Cluster. Datenpunkte 8 und 9 bilden jeweils ein eigenes Cluster. Bei einer Clusteranzahl von 3 würden Datenpunkt 8 und 9 ein gemeinsames Cluster bilden. Bei der Wahl der Clusterzahl kann das Heterogenitätsmaß dargestellt im blauen Kurvenverlauf (vergl. 2.4) dienen.

Das Ergebnis der Clusteranalyse d.h. die Aufteilung in Cluster kann in JMP als neue Variable in den Datensatz gespeichert werden, so dass das Ergebnis für weitere Analysen zur Verfügung steht. Die Speicherung erfolgt darüber, dass in dem Titelfeld *Hierarchical Clustering* die rechte Maustaste gedrückt wird und die Option *Save Clusters* gewählt wird. Eine neue Variable mit dem Namen *Cluster* erscheint in der Datenmatrix. Diese kann umbenannt werden.

5.1.3 Beispiel für die K-Means-Clusteranalyse

Die K-Means-Clusteranalyse kann über das selbe Menü wie die Hierarchische Clusteranalyse aufgerufen werden (vergl. 5.1.1). Bitte nutzen Sie für dieses Beispiel den Datensatz bsp2.jmp. Alle weiteren Einstellungen sind wie in Abbildung 21 vorzunehmen.

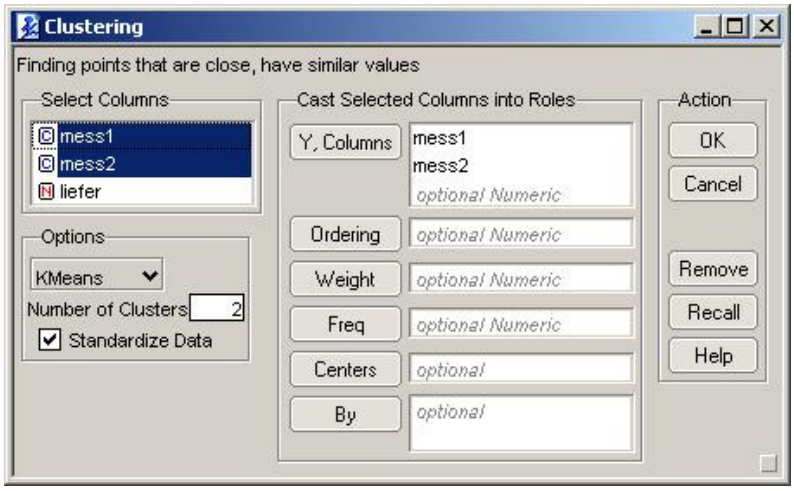

Abbildung 21: Einstellung für das K-Means-Verfahren mit dem Datensatz bsp2.jmp

Mit *OK* wird die Eingabe übernommen. Das sich öffnende Fenster (vergl. Abbildung 22 links) zeigt unter *Cluster Means* die Eigenschaften des Cluster Seed. Mit einem Klick auf OK (vergl. Abbildung 22 rechts) wird das iterative Verfahren angestoßen. Statistische Maßzahlen für die entstandenen Cluster werden ausgegeben.

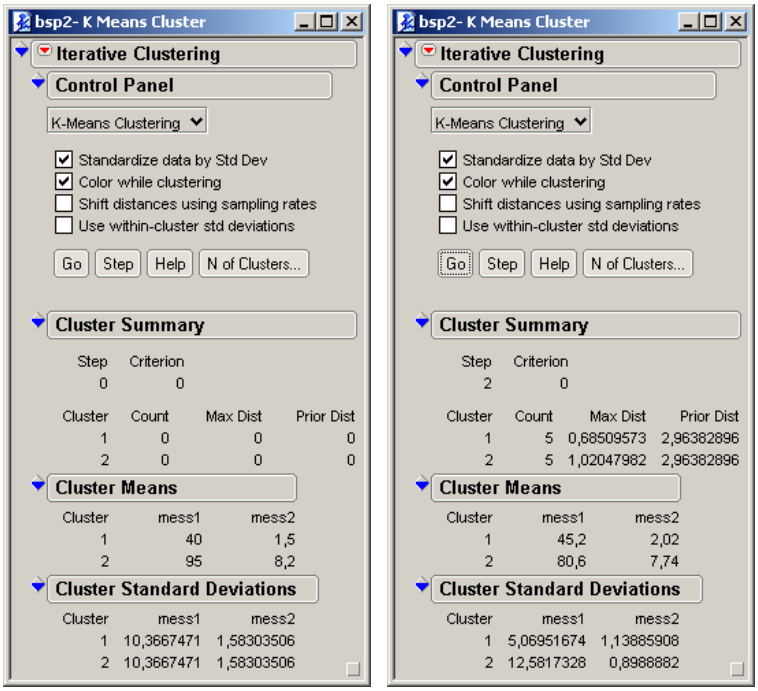

Abbildung 22: Startpunkt des K-Means-Verfahren links und Ergebnis rechts

Mit einem *Rechtsklick* auf *Iterative Clustering* können über das Kontextmenü die Grafikoptionen aufgerufen werden (vergl. Abbildung 24 und Abbildung 25). Die Veränderung der Clusteranzahl auf 3 zeigt das folgende Bild (vergl. Abbildung 25 und Abbildung 26).

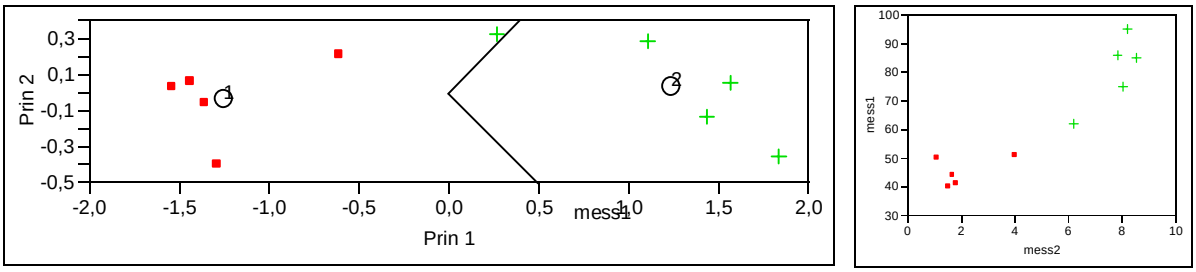

Abbildung 23: Biplot und Scatterplot des K-Means Verfahrens bei 2 Clustern (bsp2.jmp)

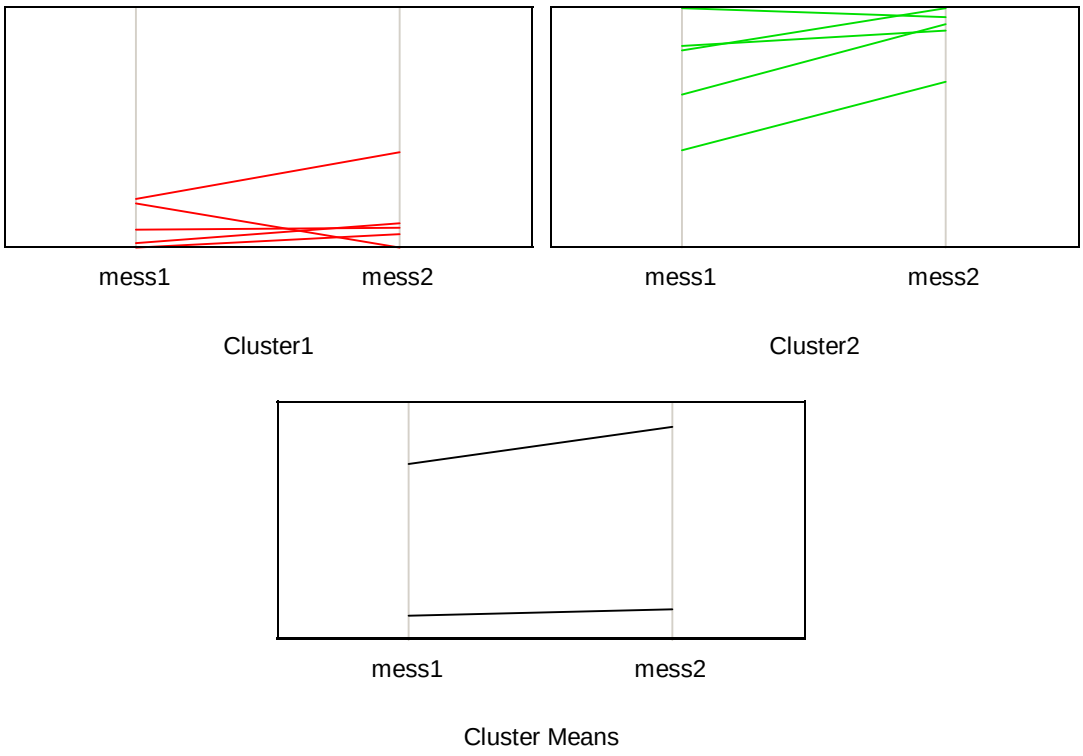

Abbildung 24: Parallel-Coordinate-Plots des K-Means Verfahrens bei 2 Clustern (bsp2.jmp)

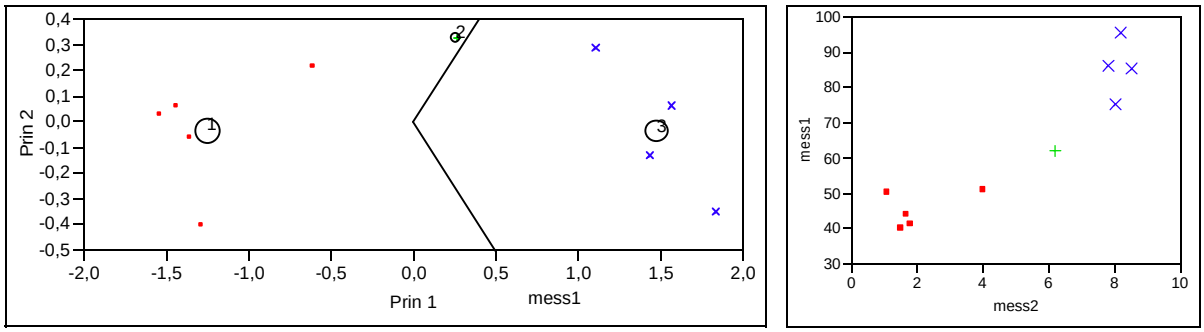

Abbildung 25: Biplot und Scatterplot des K-Means Verfahrens bei 3 Clustern
(Datensatz: bsp2.jmp)

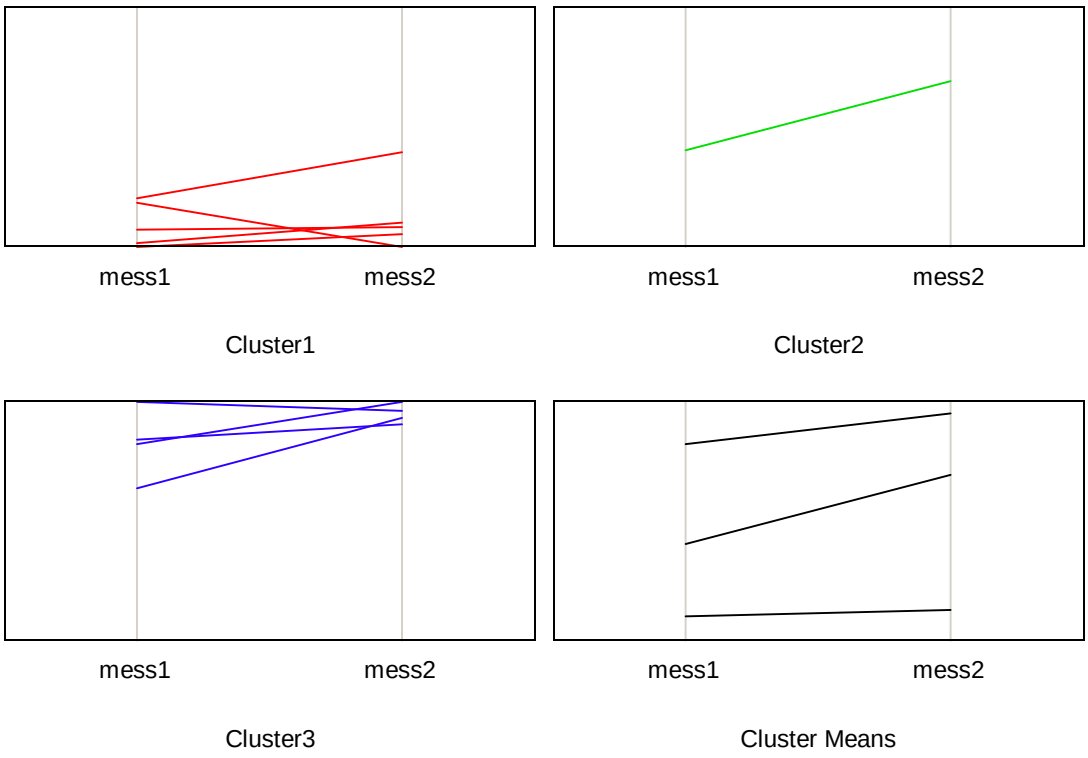

Abbildung 26: Parallel-Coordinate-Plots des K-Means Verfahrens bei 3 Clustern (Datensatz: bsp2.jmp)

Ein Beispiel mit einem größeren Datensatz (bsp1.jmp) und mehr Variablen zeigt Abbildung 13, Abbildung 14 und Abbildung 15.

Ebenso wie bei der Hierarchischen-Clusteranalyse kann das Ergebnis des K-Means-Verfahren als neue Variable in den Datensatz gespeichert werden, so dass das Ergebnis für weitere Analysen zur Verfügung steht. Die Speicherung erfolgt darüber, dass in dem Titelfeld *Iterative Clustering* die rechte Maustaste gedrückt wird und die Option *Save Clusters* gewählt wird. Eine neue Variable mit dem Namen *Cluster* erscheint in der Datenmatrix. Diese kann umbenannt werden.

5.2 Trees

5.2.1 Ein einfaches Beispiel

Ziel dieses einfachen Beispiels ist es, die Ergebnisse des Tree-Verfahren mit der Clusteranalyse und einem Verfahren der „klassischen Statistik“, dem t-Test, zu vergleichen.

Es wird wieder die Beispieldatei bsp2.jmp verwendet. Als erster Schritt wird einfach noch einmal eine Hierarchische Clusteranalyse mit der Fusionsmethode Single durchgeführt. In Abbildung 27 links ist das bekannte Ergebnis dargestellt. Das Ergebnis der Clusteranalyse sollte nun abgespeichert werden, und die Variable *Cluster* sollte in *liefer* (für z.B. Lieferant eines Produkts) umbenannt werden. Dabei entspräche die Zuordnung zu den Clustern in der Realität den unterschiedlichen Lieferanten. Zum Umbenennen von Variablen muss der Editiermodus für die Variable aktiviert werden. Dazu reicht ein Doppelklick auf den Variablennamen und die Eingabe des neuen Namens.

Das Ergebnis der Clusteranalyse, wird nun der Prüfung durch einen t-Test unterzogen.

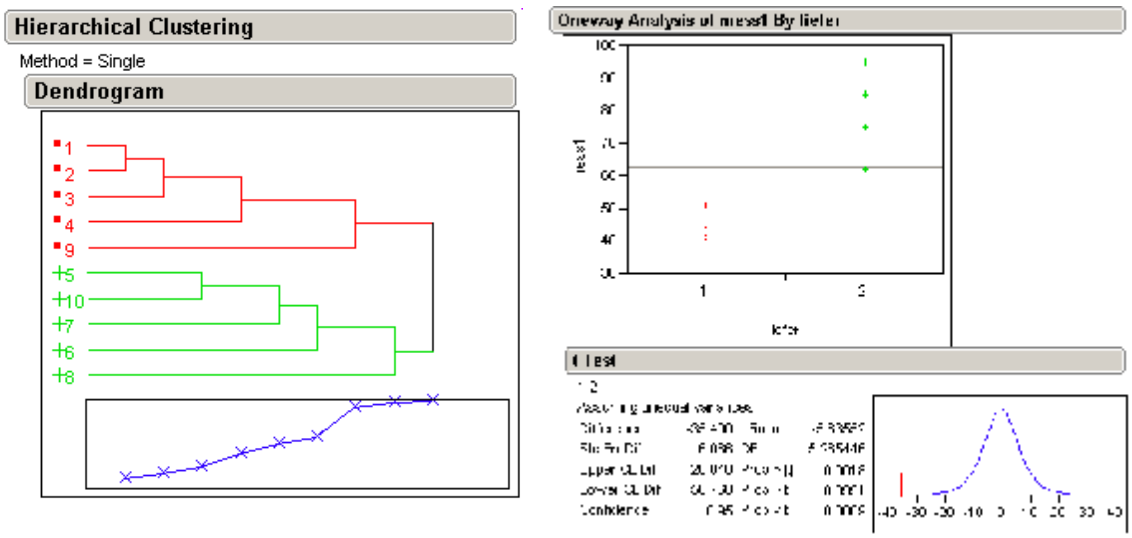

Abbildung 27: Clusteranalyse und Ergebnis eines t-Test

Der t-Test verbirgt sich hinter dem Menüpunkt *Analyze* > *Fit Y by X*. Wenn die Variable *mess1* als *Y, Response* und die Variable *liefer* als *X, Factor* festgelegt wird und die Aktion mit *OK* bestätigt wird, ergibt sich der Scatterplot der Abbildung 27 rechts mit dem Titel *Oneway Analyss of mess1 By liefer*. Das Ergebnis des t-Test kann erzeugt werden, indem mit der rechten Taste der Maus auf das Titelfeld geklickt wird und die Auswahl *t Test* getätigt wird. Weitere Ausgaben sind möglich. Als Ergebnis liefert der t-Test *Prob>|t|* kleiner 0,05. Dieses bedeutet, dass die gefundene Einteilung der Fälle in die zwei Cluster ein signifikantes Ergebnis ist.

Nun ist es Interessant, ob das Tree-Verfahren zu einem gleichen oder aber ähnlichen Ergebnisse kommt.

Das Tree-Verfahren kann über *Analyze* > *Modeling* > *Partition* aufgerufen werden. Im JMP-Starter ist es im Reiter *Model* über den Button *Partition* zu finden. Die Y,Response-Variable der Tree-Verfahren entspricht der abhängigen Variabel der „klassischen“ Statistik. Die Einstellungen sollten wie in Abbildung 28 erfolgen. Ein Klick auf OK ergibt die Partitionierung wie in Abbildung 29.

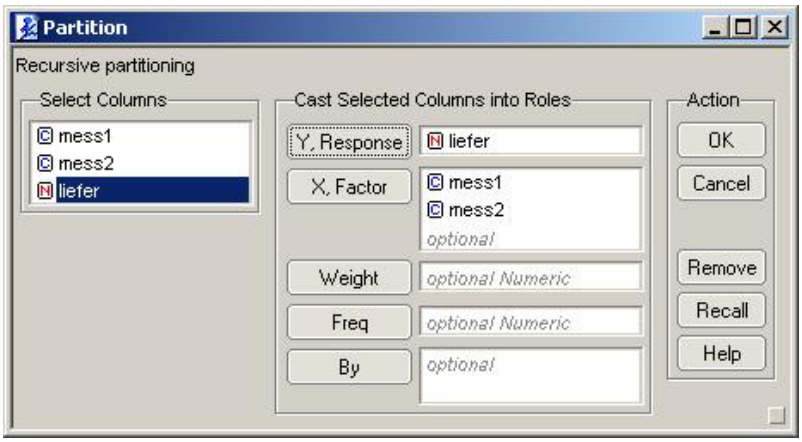

Abbildung 28: Eingabe der Variablen für das Tree-Verfahren in JMP

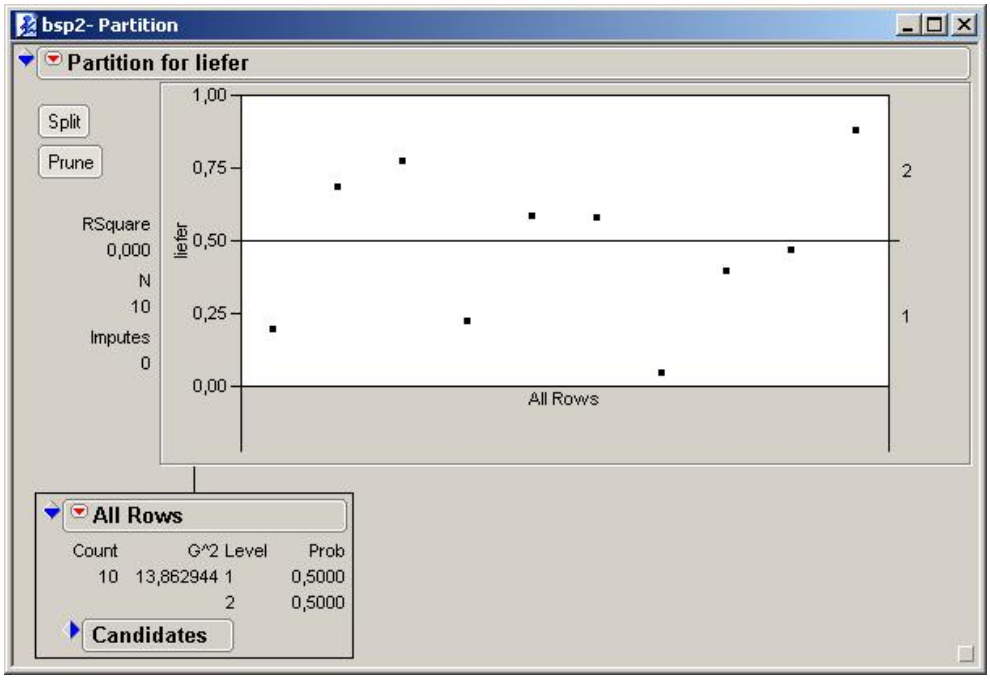

Abbildung 29: Anfangsdialog des Tree-Verfahrens in JMP

Die eigentliche Partitionierung erfolgt nun durch einen Klick auf den Button *Split* (vergl. Abbildung 29). Eine Partitionierung kann mit einem Klick auf den Button *Prune* rückgängig gemacht werden. Die Güte des Gesamtmodells wird bei jedem Partitionierungsschritt als *RSquare* berechnet.

Bei diesem einfachen Datensatz reicht ein Split aus um den Datensatz komplett aufzuteilen (vergl. Abbildung 30). Es wird ein Baum mit zwei Blättern gebildet, wobei sich die Variable mess1 als wichtigstes Split-Kriterium herausstellte. Der Split erfolgt bei *mess1>=62* bzw. *mess1<62*. RSquares (r^2) hat den Wert von 1 angenommen, weitere Splitvorgänge sind nicht möglich und nicht nötig. Indem Sie einzelnen Datensätze mit der Maus markieren können Sie interaktiv überprüfen, ob die Fallaufteilung des Tree-Verfahrens mit dem der Clusteranalyse übereinstimmt. Für dieses einfache Modell ist es der Fall.

Das Scatterplot des *Partition*-Dialogs ist ähnlich zu interpretieren wie das Biplot (vergl. Abbildung 14: Biplot). Die Gesamtstreuung des Datensatztes verursacht durch alle in die Auswertung einbezogenen Variablen wird zweidimensional dargestellt, die Response-Variable ist als Y-Achse abgebildet.

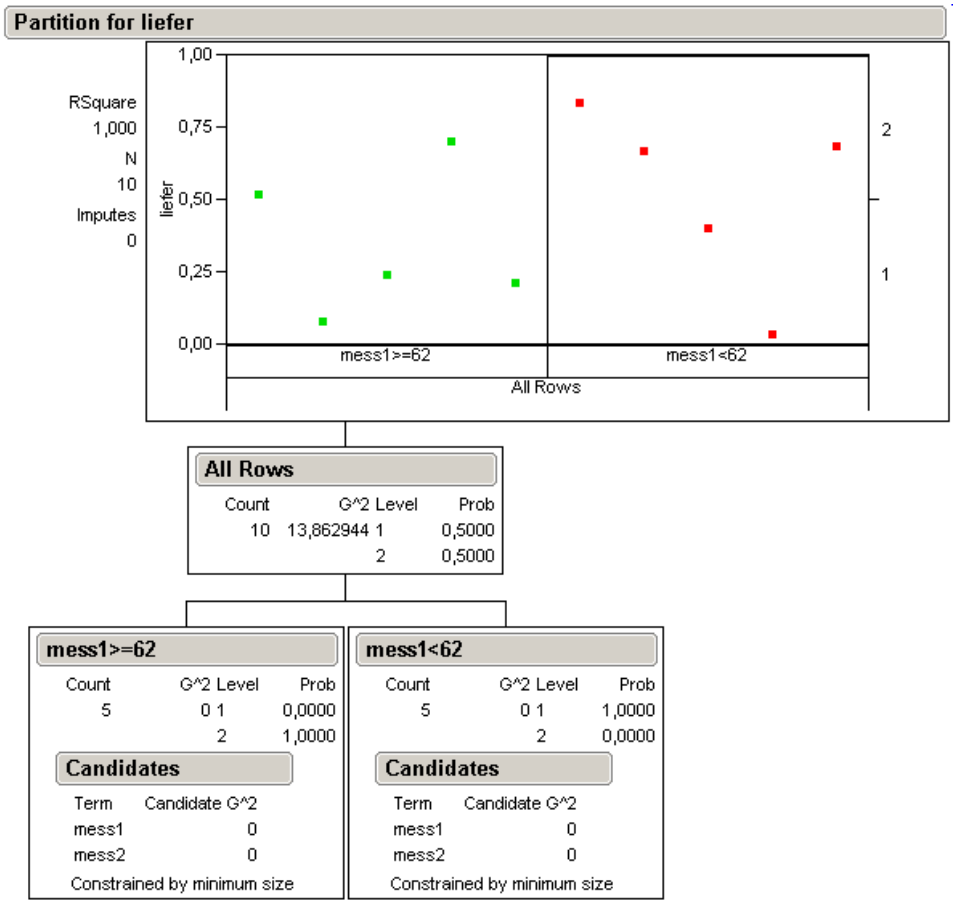

Abbildung 30: Ergebnis der Partitionierung des Beispieldatensatzes bsp2.jmp