Lisa STADLMÜLLER

Dimensionsreduktion und Klassifikation von Metabolomics-Daten

MASTERARBEIT

zur Erlangung des akademischen Grades einer Diplom-Ingenieurin

Masterstudium Operations Research und Statistik

Technische Universität Graz

Betreuer:
Univ.-Prof. Dipl.-Ing. Dr. techn. Ernst STADLOBER

Institut für Statistik

Graz, im Oktober 2011

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources and that I have explicitely marked all material which has been quotes either literally or by content from the used sources.

....
date (signature)

Danksagung

Mein besonderer Dank gilt Herrn Univ.-Prof. Dipl.-Ing. Dr. techn. Ernst Stadlober, Institut für Statistik der TU Graz, der mir diese Masterarbeit am Joanneum Research vermittelt und mich bei der vorliegenden Arbeit betreut und unterstützt hat.

Weiters möchte ich mich ganz herzlich bei Frau Dipl.-Ing. Ulrike Kleb und Frau Dipl.-Ing. Dr. Rose-Gerd Koboltschnig für ihre immer freundliche, umfassende fachliche und engagierte Betreuung bei der Erstellung dieser Arbeit bedanken.

Des weiteren gilt mein Dank Herrn Mag. Dr. Christoph Magnes, der mir bei der Erstellung dieser Masterarbeit mit seinem medizinisch-chemischen Fachwissen zur Seite gestanden ist.

Außerdem danke ich meinen Freunden und Studienkollegen, die mich während meines Studiums immer mit hilfreichen Tipps unterstützt haben.

Ganz besonders möchte ich mich bei Walter bedanken, der mich motivierte, für mich da war und mir immer das Gefühl gab, alle Hindernisse überwinden zu können.

Besonderer Dank gilt meiner Familie und hier vor allem meinen Eltern, die mir das Studium ermöglicht und mich immerwährend unterstützt haben sowie mir mit Ratschlägen zur Seite gestanden sind. Ihnen möchte ich diese Arbeit widmen.

Zusammenfassung

Im Bereich Metabolomics entstehen meist riesige Datensätze mit vielen Variablen (Features) und nur wenigen Objekten mit bekannter Gruppenzugehörigkeit. Angesichts der großen Datenmengen (in der vorliegenden Arbeit $n = 60$ Objekte und $p = 613$ Variablen) werden einerseits Verfahren zur Dimensionsreduktion vorgestellt, die zum Ziel haben, eine möglichst gute Gruppierung der Features zu erreichen, ohne viel Information für die Gruppenzugehörigkeit der Objekte zu verlieren. Andererseits ist das Ziel der Arbeit, Klassifikationsverfahren wie die *Partial-Least-Squares-Diskriminanzanalyse* (PLS-DA) anzuwenden, um Modelle zu finden, die die Gruppenzugehörigkeit der Objekte gut reproduzieren. Ein Hauptaugenmerk wird auf die Bestimmung jener Features, die für die Klassifikation von Bedeutung sind und diese Modelle beeinflussen, gelegt. Eine Anwendung der Kreuzvalidierung zeigt die Prognosetauglichkeit der gefundenen Modelle. Die Güte der Modelle wird anhand der Fehlklassifikationsrate bestimmt. Die Dimensionsreduktion und PLS-DA ermöglichen eine Reduktion auf weniger als 5 % der Variablen (20 – 30 Features). Erst dadurch wird die Voraussetzung für eine leichtere Interpretation der großen Datenmengen geschaffen.

Für die praktische Durchführung der Datenanalysen wurde das Statistik-Softwarepaket GNU R, Version 2.13.1, verwendet. Die Masterarbeit wurde in Zusammenarbeit mit der Joanneum Research Forschungsgesellschaft mbH erstellt.

Abstract

In the field of metabolomics, there often arise data sets with a huge number of variables (features) and only a few objects that have a known group affiliation. Given the large amount of data (in this master thesis $n = 60$ items and $p = 613$ variables), methods for dimension reduction are presented. They aim to achieve the best possible grouping of the features without losing much information about the group affiliation of the objects. Furthermore, the goal of this thesis is to describe classification methods such as the *Partial Least Squares Discriminant Analysis* (PLS-DA) in order to find models that reproduce the group affiliation of the objects well. The focus is on determining features that are important for the classification and that influence these models. An application of cross-validation shows the prediction capability of the models. The quality of the models is determined based on the percentage of incorrectly classified objects. The dimension reduction and PLS-DA enable a reduction to less than 5 % of the variables (20 – 30 features). Only thereby an easier interpretation of the large amounts of metabolomics data is possible.

For the practical implementation of the data analysis the statistical software package GNU R, version 2.13.1, was used. This thesis has been written in cooperation with Joanneum Research GmbH.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Abkürzungsverzeichnis

ANOVA Varianzanalyse (engl. *analysis of variance*)

%-CC prozentueller Anteil der Stichprobe, der richtig klassifiziert wird (engl. *percent of samples correctly classified*)

CoV Variationskoeffizient (engl. *coefficient of variation*)

DA Diskriminanzanalyse (engl. *discrimination analysis*)

DNA Desoxyribonukleinsäure

FTMS Fourier-Transformations-Massenspektrometrie (engl. *Fourier transform mass spectrometry*)

GC Gaschromatographie (engl. *gas chromatography*)

HK Hauptkomponente

HPLC Hochleistungsflüssigkeitschromatographie (engl. *high performance liquid chromatography*)

LC Flüssigkeitschromatographie (engl. *liquid chromatography*)

LDA lineare Diskriminanzanalyse

LOO engl. *leave one out*

LR Wahrscheinlichkeitsverhältnis (engl. *likelihood ratio*)

MAD absolute Medianabweichung (engl. *absolute median deviation*)

MCA engl. *metabolic control analysis*

MCD engl. *minimum covariance determinant*

mRNA Boten-RNA (engl. *messenger RNA*)

MS Massenspektrometrie (engl. *mass spectrometrie*)

NIPALS engl. *non-linear iterative partial least squares*

NMR Kernresonanzspektroskopie (engl. *nuclear magnetic resonance*)

PCA Hauptkomponentenanalyse (engl. *principal component analysis*)

PLS Partial-Least-Squares (engl. *partial least squares*)

PLS-DA Partial-Least-Squares-Diskriminanzanalyse (engl. *partial least squares discriminant analysis*)

PLS-R Partial-Least-Squares-Regression

QDA quadratische Diskriminanzanalyse

RNA Ribonukleinsäure

SN Signal-Rausch-Verhältnis (engl. *signal-to-noise ratio*)

1. Einleitung

1.1. Was ist Metabolomics

Der Begriff *Metabolomics* bezeichnet die Erforschung der Stoffwechselvorgänge einer Zelle. Nachstehend wird eine kurze Beschreibung der Zellabläufe gegeben, ausführliche Beschreibungen sind in Villas-Boas u. a. (2007) und Nielsen und Jewett (2007) nachzulesen.

Im Zellkern existieren die Desoxyribonukleinsäure (DNA) und die Ribonukleinsäure (RNA). Die Gene stellen Abschnitte von solchen DNA-Molekülen dar und sind Anleitungen zur Proteinbildung. Bei der *Transkription*, die eine Synthese zwischen RNA und DNA darstellt, wird die Information der DNA in Boten-RNA (mRNA) kopiert. Mit mRNA als Informationsträger wird die genetische Information aus dem Zellkern zu den Ribosomen[1] transportiert, wo sie bei der *Translation* in eine Aminosäurenkette und in Folge zu einem Protein umgesetzt wird. Manche Proteine katalysieren als Enzyme[2] biochemische Stoffwechselreaktionen zur Umwandlung von *Metaboliten*. Der Informationsfluss in einer Zelle ist in Abbildung 1.1 vereinfacht dargestellt, wobei es zwischen den einzelnen Substanzen weit mehr Vernetzungen gibt, als hier angedeutet werden.

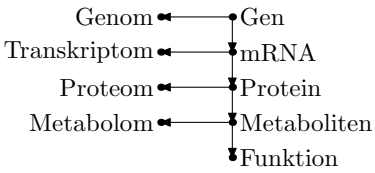

Abbildung 1.1.: Vereinfachte Darstellung des Informationsflusses in einer Zelle (vgl. Goodacre, 2005)

Das *Metabolom* beinhaltet die Gesamtheit aller Metaboliten, die nicht-genetische niedermolekulare Substrate bzw. Zwischenprodukte darstellen und durch Stoffwechselprozesse einer Zelle entstehen. Bei der Technologie von Metabolomics werden die Metaboliten vollständig identifiziert und quantifiziert, d. h. es kommt eine ganzheitliche Betrachtungsweise zum Einsatz.

[1] Ribosome befinden sich im Zellplasma und bestehen hauptsächlich aus Ribonukleinsäuren und Protein.

[2] Alle Enzyme sind Proteine.

Metabolomics entwickelte sich gemäß Griffiths (2008) in den vergangenen Jahren zu einem vielseitig anwendbaren Gebiet in der Industrie, Biologie, Medizin und den Umweltwissenschaften. Die Analyse der Metaboliten ermöglicht die Erforschung von Veränderungen beim Organismus in Folge von Krankheit, Vergiftung, genetischer Manipulation oder umweltbedingten Belastungen.

1.2. Motivation von Metabolomics aus medizinischer Sicht

Metabolomics ist ein mächtiges Werkzeug, mit dem der Status der Metaboliten im Organismus erfasst wird. Bereits kleine Abweichungen vom normalen Status können krankhafte Anomalien aufzeigen. Damit können Erkrankungen frühzeitig erkannt werden. Die Aufgabe liegt darin, das frühe Stadium einer Krankheit durch ein kombinatorisches Muster von Metaboliten in ihrem Massenspektrum zu erkennen (Hunter, 2009). Dies geschieht mit Hilfe von *Biomarkern*, für deren Identifikation Metabolomics einen wesentlichen Beitrag liefert. *Biomarker* sind messbare Produkte von Organismen, die als Indikatoren für Krankheiten oder zur Risikobewertung von Giftstoffen herangezogen werden (Lin u. a., 2006).

Mit dem Forschungsbereich Metabolomics lassen sich in der Medizin Fragestellungen aus den Bereichen *drug response*[3], *personalized medicine*, Wirkstoffentwicklung und Biotechnologie beantworten.

In der *drug response* wird die unterschiedliche Reaktion von Individuen (meist Ratten) auf dasselbe Medikament durch Verwendung der Massenspektrometrie festgestellt. Damit trägt Metabolomics einen wesentlichen Beitrag zum Verständnis von Krankheiten und zur besseren Entwicklung von Medikamenten in der Pharmaindustrie bei.

Bei der individualisierten Arzneimitteltherapie (*personalized medicine*) erhalten die Patienten Medikamente, die auf ihr Genmaterial und die Umwelteinflüsse des Einzelnen maßgeschneidert sind, in der vorhergesagt wirksamen Dosierung. Dabei liefert die Erforschung der Gene und Metaboliten neue Möglichkeiten, um die Medikamente besser auf den Patienten und seine Umgebung anzupassen.

Metabolomics findet jedoch nicht nur in der Medizin ihre Anwendung, sondern auch in der Toxikologie[4] und in der Nahrungsmittelindustrie. Mit Hilfe von Metabolomics werden in der Toxikologie die biochemischen Veränderungen der Organismen in Folge von Giftstoffbelastungen analysiert (vgl. Lin u. a., 2006). In der Nahrungsmittelindustrie wird mit Metabolomics sowohl die Qualität von Nährstoffen beurteilt als auch der Nährstoffgehalt von Lebensmitteln verbessert. Weiterführende Informationen zur

[3] *Drug response* bezeichnet das Ansprechen auf Medikamente.
[4] Toxikologie beschäftigt sich mit Giftstoffen und der Behandlung von Vergiftungen.

Anwendung von Metabolomics in der Toxikologie und Nahrungsmittelindustrie geben Hunter (2009) und Dunn u. a. (2005).

1.3. Metabolomics und Statistik

Die Datenanalyse von Metabolomics-Daten wird bei Brown u. a. (2005) in vier Kategorien unterteilt. Die Analyse kann mit univariaten oder multivariaten statistischen Methoden, *unsupervised learning*[5], *supervised learning*[6] oder systembasierenden Methoden, die MCA verwenden, erfolgen. Tabelle 1.1 gibt einige Methoden zur Datenanalyse entsprechend der vier Kategorien an.

Univariate und multivariate statistische Methoden	Unsupervised Learning	Supervised Learning	Systembasierende Methoden
Mittelwert	Hauptkomponenten-analyse (PCA)	Diskriminanz-analyse (DA)	Metabolic control analysis (MCA)
Standard-abweichung	Cluster-verfahren	Partial-Least-Squares (PLS)	
Variations-koeffizient		Künstliche neuronale Netze	
Korrelation u. Regression[7]			

Tabelle 1.1.: Statistische Analysemethoden von Metabolomics-Daten

Statistische Verfahren werden verwendet, um die hohe Qualität der Metabolomics-Daten sowie Interpretationen und Rückschlüsse zu bestätigen. Die nachstehende Beschreibung der verschiedenen statistischen Verfahren orientiert sich an Goodacre u. a. (2007) sowie an Brown u. a. (2005).

1.3.1. Univariate und multivariate statistische Methoden

Obwohl bei einem Metabolomics-Experiment multivariate Datenmengen entstehen, kann man univariate statistische Methoden anwenden, um Variablen einzeln zu betrachten, die sich zwischen verschiedenen Gruppen signifikant erhöhen oder kleiner werden. Typische univariate Anwendungen sind die Varianzanalyse (ANOVA), t-Tests

[5] Die Verfahren von *unsupervised learning* betrachten die Daten als Gesamtheit und versuchen allgemein gültige Regeln und Strukturen zu finden.

[6] Die *supervised-learning*-Methoden verwenden bereits bekannte Information, um eine Klassifikation der Daten durchzuführen.

[7] Die Methode *Korrelation und Regression* ist in Dillon und Goldstein (1984) nachzulesen.

und z-Tests. Diese parametrischen Verfahren setzen voraus, dass die Daten einer Normalverteilung genügen. Wenn die Normalverteilungsannahme bei den Daten nicht zutrifft, kann eine nicht-parametrische Methode wie der Kruskal-Wallis-Test angewandt werden.

Bevor die Daten einer multivariaten Analyse unterzogen werden, kann es hilfreich sein, die statistischen Eigenschaften wie Mittelwert und Varianz der Variablen bzw. Objekte zu untersuchen. Sinnvoll ist es auch, die Korrelationen zu betrachten. Da bei n Variablen bereits $\mathcal{O}(n^2)$ Kombinationen vorhanden sind, werden solche Analysen meist automatisiert durchgeführt. Ungewöhnlich große Varianzen können Ausreißer aufzeigen, die gegebenenfalls vom Datensatz entfernt werden müssen, um sinnvolle Schlussfolgerungen treffen zu können.

Klassische univariate statistische Verfahren sind bei großen Metabolomics-Datenmengen jedoch nicht ausreichend. Für eine zielführende Klassifikation werden multivariate statistische Methoden benötigt, die die Daten dazu verwenden, das bestmögliche Klassifikationsmodell zu bestimmen.

1.3.2. Supervised-learning-Methoden

Supervised-learning-Methoden werden verwendet, wenn sowohl über die Input- als auch über die Outputparameter – die beide zur Analyse verwendet werden sollen – Informationen bekannt sind. Damit ist es möglich, einem Modell ein bestimmtes Verhalten anzutrainieren. Ein typisches supervised-learning-Metabolomics-Beispiel ist das Klassenproblem. Dabei liegen zwei Klassen von Stichproben vor, nämlich Patienten mit einer bestimmten Krankheit und gesunde Kontrollpersonen. Ziel ist es, Biomarker der Inputparameter zu bestimmen, mit denen es möglich ist, die Stichprobe in die zwei Klassen aufzuteilen. Die Partial-Least-Squares-Diskriminanzanalyse (PLS-DA), die in Abschnitt 3.2.2 behandelt wird, ist ein bekannter Vertreter der supervised-learning-Methoden.

Die Methoden des *supervised learning* sind in ihrer Anwendung mächtiger als die *unsupervised-learning*-Methoden, da sie für die Erstellung des Modells auch auf die Informationen der Outputparameter zurückgreifen.

1.3.3. Unsupervised-learning-Methoden

Im Unterschied zu den *supervised-learning*-Methoden verwenden die *unsupervised-learning*-Methoden für die Analyse der Daten nur Informationen der Inputparameter. Die am weitesten verbreiteten Verfahren sind die Hauptkomponentenanalyse (PCA) und Clusterverfahren.

Clusterverfahren sind in der Praxis oft schwierig anzuwenden. Ein Nachteil von Clusterverfahren ist, dass sie eine Partition der Daten liefern, ohne jedoch die Zuverlässigkeit des Ergebnisses zu prüfen. Für eine ausführliche Beschreibung der Clusteranalyse wird auf Backhaus u. a. (2008, S. 395 ff.) verwiesen. Eine Beschreibung der Hauptkomponentenanalyse sowie ein Vergleich zur PLS-DA finden sich in Abschnitt 2.4.1.

2. Theoretischer Teil

2.1. Versuchsplanung

Einen wesentlichen Teil für die Erforschung des Metaboloms stellt die Aufstellung eines Versuchsplans dar. Nach der Durchführung der biologischen Experimente werden die Daten in drei Schritten analysiert. Nach der instrumentellen Messung werden zuerst die Rohdaten normiert. Dann werden diese modifizierten Daten auf die relevanten Input-Variablen reduziert, sodass anschließend eine statistische Analyse durchgeführt werden kann. Die genaue Vorgangsweise bei der Versuchsplanung von Metabolomics-Daten ist in Brown u. a. (2005) nachzulesen.

Bei einem Metabolomics-Experiment entstehen üblicherweise riesige Datenmengen, was eine präzise Planung desselben notwendig macht. Wie mit diesen riesigen Datenmengen umgegangen wird, zeigt Abbildung 2.1. Sie beinhaltet eine Darstellung des Informationsflusses bei einem Metabolomics-Experiment (vgl. Goodacre u. a., 2007).

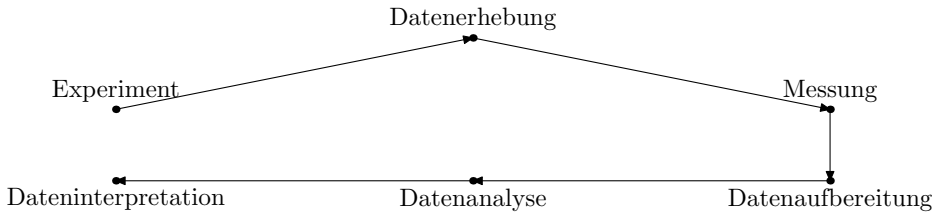

Abbildung 2.1.: Informationsfluss bei einem Metabolomics-Experiment

Bereits bei der Erhebung und Messung der Daten – noch vor der statistischen Analyse – gibt es unzählige Parameter auszuwählen. Als Statistiker ist man natürlich auf diese Voreinstellungen und Analysen, die im Vorfeld passieren, angewiesen.

Einige Methoden der Datenaufbereitung sowie Datenreduktion zeigt Tabelle 2.1.

Ziel der Datenanalyse ist, ein Schema in den Daten zu finden, das biologische Informationen liefert, mit denen man Hypothesen aufstellen kann. Diese Hypothesen werden in Folge getestet und gegebenenfalls neu formuliert.

Datenaufbereitung
Normierung der Daten mit internem Standard
Normierung der Daten durch Transformation
Erkennung von Fehlern des Messgerätes sowie von Abweichungen
Identifikation von Ausreißern in den Daten
Handhabung von fehlenden Daten
Datenreduktion
Anwendung der Datenanalyse nur auf einen gewissen Bereich der modifizierten Daten
Ausschließen von Variablen, deren Variation eine erlaubte Schranke überschreitet
Entfernung von Ausreißern, die z. B. mittels PCA entdeckt werden

Tabelle 2.1.: Methoden der Datenaufbereitung und Datenreduktion

2.1.1. Messung der Daten

Die Metaboliten sind chemische Größen und können mit Standardwerkzeugen der Chemie analysiert werden. Goodacre u. a. (2004) gibt einen Überblick über analytische Methoden wie Molekularspektroskopie, Massenspektrometrie (MS), Hochleistungsflüssigkeitschromatographie (HPLC) und Kernresonanzspektroskopie (NMR). Die Sensitivität, Auflösung sowie Reproduzierbarkeit der Massenspektrometrie kann erhöht werden, indem sie mit Gaschromatographie (GC) oder Flüssigkeitschromatographie (LC) kombiniert wird. Die Wahl des technischen Analyseverfahrens hängt von der Art der Stichprobe ab. NMR ist eine weniger empfindliche Methode und gibt umfassende Informationen über einen großen Bereich der Metaboliten. Damit liefert sie grobe Verallgemeinerungen und eignet sich für Massenanwendungen. Der große Vorteil von NMR besteht darin, dass diese Methode am lebenden Organismus durchgeführt werden kann. Im Gegensatz dazu weist MS eine größere Sensitivität auf und eignet sich besser für komplexe Stichproben mit wechselnden Konzentrationen (Hunter, 2009).

Die folgende Beschreibung der Massenspektrometrie geht auf Downard (2004) zurück. Bei der Massenspektrometrie wird die Masse von Teilchen mit einem Massenspektrometer gemessen. Der Massenspektrometer besteht aus drei Komponenten: der Ionenquelle, dem Massenanalysator und dem Ionendetektor. In der Ionenquelle werden die Moleküle ionisiert oder geladen und anschließend im Analysator gemäß ihrem Gewicht und ihrer Ladung im Vakuum getrennt. Der Ionendetektor misst die elektrischen Strömungen, die durch die sich bewegenden Ionen erzeugt werden.

In dieser Arbeit werden in Kapitel 3.3 nur Datensätze analysiert, die mit LC und einem hochauflösenden Fourier-Transformations-Massenspektrometer (FTMS) getrennt und gemessen wurden. LC-FTMS verbindet die Flüssigkeitschromatographie mit der hochauflösenden Fourier-Transformations-Massenspektrometrie. Hierbei stellt der Übergang von der Chromatographie zur Massenspektrometrie eine Herausforderung dar.

An der Schnittstelle muss das Lösungsmittel verdampfen, bevor die Ionen im Massen-Messgerät einer Spannung unterzogen werden.

2.2. Statistische Methoden der Datenaufbereitung

2.2.1. Ausreißerdiagnostik

Ausreißer lassen sich sowohl für die Features als auch für die Objekte bestimmen. In beiden Fällen kann man die *Mahalanobis-Distanz* zum Finden von Ausreißern in den Daten heranziehen.

Die Mahalanobis-Distanz stellt ein numerisches Maß dar, um multivariate Ausreißer erkennen zu können. Bei multivariaten Daten empfiehlt es sich, Ausreißer innerhalb jeder Gruppe zu bestimmen, wenn das Ziel in der Bestimmung von Gruppenunterschieden liegt. Die quadrierte Mahalanobis-Distanz ist unter der Normalverteilungsannahme Chi-Quadrat-verteilt (χ^2_p) mit p Freiheitsgraden (p gibt die Anzahl der Variablen an, wenn die Ausreißer bezüglich der Objekte bestimmt werden). Deshalb wird bei der Identifikation von multivariaten Ausreißern die kritische Grenze durch das $(1-\alpha)$-Quantil $\chi^2_{p,1-\alpha}$ der Chi-Quadrat-Verteilung festgelegt. Ausreißer erscheinen in den Daten mit einer großen Mahalanobis-Distanz vom Zentrum der Originaldaten. Objekte, deren quadrierte Mahalanobis-Distanz größer als der kritische Wert $\chi^2_{p,1-\alpha}$ ist, werden als Ausreißer identifiziert (Warner, 2008, und Tabachnick und Fidell, 1983).

Die Mahalanobis-Distanz zieht die Verteilung der Punkte im Variablenraum in Betracht und ist von der Skalierung der Variablen unabhängig. Sie ist als Distanz zwischen einem Beobachtungsvektor x_i und dem Datenzentrum $\overline{x}$ definiert als

$$\mathrm{d}_{\mathrm{Mahalanobis}} = \sqrt{(x_i - \overline{x})^T \cdot C^{-1} \cdot (x_i - \overline{x})},$$

wobei x_i den Vektor des i-ten Objektes angibt, das Zentrum der Daten als Vektor $\overline{x}$ der arithmetischen Mittel geschätzt wird und C die empirische Kovarianzmatrix darstellt.

Bei der Identifikation von Ausreißern ist entscheidend, wie gut das Zentrum und die Kovarianz der Daten geschätzt wurden. Die herkömmlichen Schätzer *Stichprobenmittel* $\overline{x}$ und *Kovarianzmatrix* sind aufgrund ihrer Sensibilität und leichten Verzerrbarkeit oft nicht gut geeignet, um Ausreißer zu entdecken. Varmuza und Filzmoser (2009) empfehlen, für die Berechnung der Mahalanobis-Distanz robuste Schätzer wie das Zentrum und die Kovarianzmatrix des MCD-Schätzers zu verwenden. Dabei steht MCD für *Minimum Covariance Determinant*.

Die Berechnung des MCD-Schätzers erfolgt folgendermaßen:

1. Zuerst wird diejenige h-elementige Teilmenge der Stichproben bestimmt, bei der die Determinante der Varianz-Kovarianzmatrix am kleinsten ist.

2. Der Lokalisations- bzw. Lageschätzer $\overline{x}_h$, die robuste Alternative zum Stichprobenmittel, ist in Folge als arithmetisches Mittel dieser Teilstichprobe definiert.

3. Die empirische Varianz-Kovarianzmatrix S_h dieser h-elementigen Teilstichprobe liefert den Schätzer für die Varianz-Kovarianzmatrix.

4. Anschließend wird für jeden Beobachtungsvektor x des Datensatzes die quadrierte Mahalanobis-Distanz

$$(x - \overline{x}_h)^T \cdot S_h^{-1} \cdot (x - \overline{x}_h)$$

basierend auf $\overline{x}_h$ und S_h bestimmt.

5. Alle Beobachtungsvektoren x, deren quadrierte Mahalanobis-Distanz größer als der kritische Wert $\chi^2_{p,0.975}$ ist, werden aus dem Datensatz entfernt. Wenn die Ausreißer bezüglich der Objekte bestimmt werden, bezeichnet p die Anzahl der Variablen.

Die Wahl der Teilstichprobengröße h bestimmt die Robustheit des MCD-Schätzers. Varmuza und Filzmoser (2009) schlagen die Wahl $h = 0.75n$ vor, wobei n die Gesamtanzahl der Stichproben angibt. Diese Wahl stellt einen guten Kompromiss zwischen nötiger Robustheit und Genauigkeit der Schätzer dar.

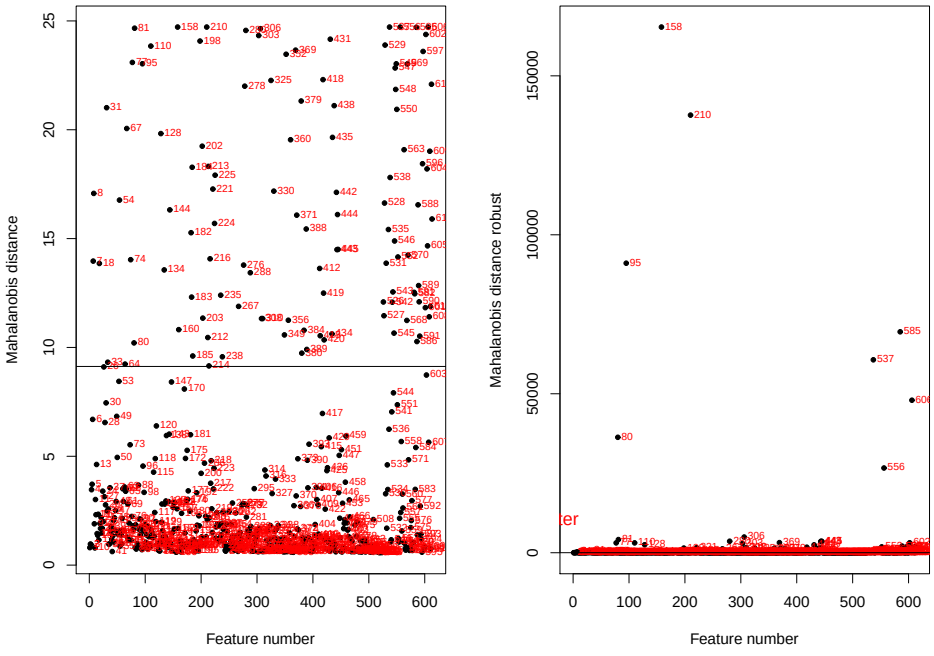

Abbildung 2.2.: Identifikation der Ausreißer mit Hilfe der Mahalanobis-Distanz

In Abbildung 2.2 werden mit der Mahalanobis-Distanz Ausreißer in einem Datensatz mit 613 Variablen, der in Abschnitt 3.1 genau beschrieben wird, identifiziert. In der

linken Grafik werden die Schätzer für das Zentrum und die Kovarianz nach der klassischen Variante, in der rechten Grafik nach der robusten Variante berechnet. Es ist klar ersichtlich, dass die Ausreißer bzgl. der Variablen mit der robusten Variante besser identifiziert werden können. Es ist generell oft der Fall, dass multivariate Ausreißer nur mit der robusten Variante der Mahalanobis-Distanz auffallen.

2.2.2. Datentransformation

Vor der Anwendung vieler multivariater Analysemethoden und vor allem für die Klassifikation ist es notwendig, die Datenmatrix entsprechend zu transformieren. Die folgenden Ausführungen über die Datentransformation orientieren sich hauptsächlich an Brereton (2010).

Die Datenvorbereitung stellt im Metabolomics-Bereich ein wesentliches Thema dar, da falsche Vorarbeit zu falschen Rückschlüssen bei den Daten führen kann. Es gibt drei prinzipielle Verfahren, um eine Datenmatrix, bei der sich die Variablen in den Spalten und die Objekte in den Zeilen befinden, zu skalieren:

- Transformation eines individuellen Eintrags der Matrix,
- Skalieren der Zeilen oder
- Skalieren der Spalten.

Gerade in der Metabolomics werden einige hundert Metaboliten gemessen, von denen manche sehr hohe Werte aufweisen können. Werden hier nicht die Spalten, also die Variablen, skaliert, so dominieren diese wenigen Variablen die gesamte weitere Analyse und die Schwankungen der anderen, kleineren Variablen hätten kaum Auswirkungen auf das Ergebnis.

Bei Metabolomics-Datensätzen im Speziellen wird vor den Analysen oft eine Standardisierung der Daten durchgeführt. Dabei werden die Daten zuerst um den Mittelwert zentriert und anschließend jede Spalte (bzw. Variable) durch ihre Standardabweichung dividiert, sodass der Eintrag x_{ij} der $(n \times p)$-Datenmatrix X zu

$$z_{ij} = \frac{x_{ij} - \overline{x}_j}{s_j}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, p,$$

transformiert wird, wobei s_j die empirische Standardabweichung der Variable j angibt. Üblicherweise wird die empirische Standardabweichung mit

$$s_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \overline{x}_j)^2}$$

berechnet. Mit der Standardisierung ist gewährleistet, dass alle Variablen einen ähnlichen Einfluss auf die Klassifizierung ausüben.

Eine weitere Überlegung zur Standardisierung wird notwendig, wenn die Daten in zwei Datensätze aufgeteilt werden, wie es bei der Kreuzvalidierung in Abschnitt 2.5.2 der Fall ist. Es gibt in diesem Fall zwei Ansätze, um die Daten zu standardisieren:

1. Der gesamte Datensatz (also sowohl Trainingsdaten als auch Testdaten) wird standardisiert, indem der gemeinsame Mittelwert und die gemeinsame Standardabweichung des vollständigen Datensatzes verwendet werden.

2. Es wird nur der Trainingsdatensatz standardisiert und der Testdatensatz wird dann mit dem Mittelwert und der Standardabweichung des Trainingsdatensatzes normiert.

Die erste Standardisierungsmethode liefert meist eine bessere Abgrenzung zwischen den unterschiedlichen Gruppen. Statistisch gesehen ist die zweite Standardisierungsmethode korrekter, aber andererseits auch komplexer und mit mehr Aufwand verbunden, wenn viele Vorhersagen der Gruppenzugehörigkeit getroffen werden sollen. In Abschnitt 2.5.2 wird bei der Kreuzvalidierung ausschließlich die zweite Standardisierungsmethode angewandt.

2.3. Methoden zur Dimensionsreduktion

Vor Beginn der Dimensionsreduktion ist es sinnvoll, zu untersuchen, ob zwischen den verschiedenen Variablen Zusammenhänge bestehen und wie stark diese sind. Dafür wird der *empirische Korrelationskoeffizient*

$$\mathrm{r}(X,Y) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\overline{y})^2}} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2}}$$

herangezogen, wobei $x_1,\ldots,x_n$ und $y_1,\ldots,y_n$ die Ausprägungen zweier unterschiedlicher Zufallsvariablen X und Y bezeichnen.

Notation 1. *Zwei Variablen X und Y gelten als* hoch korreliert*, falls der Betrag ihres empirischen Korrelationskoeffizienten* $\mathrm{r}(X,Y)$ *einen bestimmten Schwellwert t, $t \in [0,1]$, überschreitet, d. h.* $|\mathrm{r}(X,Y)| > t$.

Im folgenden Abschnitt wird mit Hilfe der Korrelationen zwischen den Variablen (Features) eine Teilung der Variablen bzw. Features in verschiedene Gruppen, die sogenannten Cluster, vorgenommen. Dafür wird die Korrelationsmatrix als Graph modelliert, wobei zwischen der Variablen X und der Variablen Y nur dann eine Kante existiert, wenn $|\mathrm{r}(X,Y)| > t$. Die Variablen selbst stellen die Knoten dar.

Ziel ist es, in einem Graphen Teilgraphen bzw. Cluster so zu bilden, dass die Korrelationen innerhalb der Cluster maximiert und die Korrelationen zwischen den Clustern

minimiert werden. In Folge sollen alle Korrelationen innerhalb der Cluster betragsmäßig größer als t und alle Korrelationen zwischen den Clustern betragsmäßig kleiner oder gleich t sein. Diese beiden Bedingungen werden in Forderung 1 zusammengefasst.

Forderung 1. *In einem Graphen sollen Teilgraphen bzw. Cluster so gebildet werden, dass*

a) die Korrelationen innerhalb der Cluster betragsmäßig maximiert bzw. größer als ein Schwellwert t werden,

b) die Korrelationen zwischen den Clustern betragsmäßig minimiert bzw. kleiner oder gleich einem Schwellwert t werden.

Beide Teile aus Forderung 1 sind für einen bestimmten Schwellwert t aber nicht immer gleichzeitig erfüllbar, wie Beobachtung 1 zeigt.

Beobachtung 1. *Nicht für jeden Graphen G gibt es eine Teilung, sodass für einen bestimmten Schwellwert t alle Korrelationen innerhalb der Teilgraphen (Cluster) betragsmäßig größer als t sind und alle Korrelationen zwischen den Teilgraphen (Cluster) betragsmäßig höchstens so groß wie t sind.*

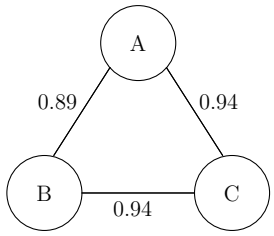

Abbildung 2.3.: Gegenbeispiel: Es gibt keinen optimalen Algorithmus, der bei einem Schwellwert von 0.9 die Korrelationen innerhalb der Cluster maximiert und zwischen den Clustern minimiert

Beweis. Gegeben sei der Graph G aus Abbildung 2.3. Angenommen, der Schwellwert für hoch korrelierte Zufallsvariablen beträgt $t = 0.9$.

Sei $u_i,\ i = 1, 2, 3$, eine gültige Partitionierung der Knotenmenge von G in i Teilmengen.

Für $i = 1$ wird der Graph G nicht geteilt und es liegt ein 3-er Cluster vor. In diesem Fall ist $\mathrm{Kor}(A, B) = 0.89$, d. h. eine Korrelation innerhalb des Clusters ist geringer als 0.9.

Für $i = 2$ wird der Graph G in zwei Teilgraphen unterteilt. Angenommen, die Teilgraphen sind $\{B, C\}$ sowie $\{A\}$. Die Forderung, die Korrelation innerhalb der Cluster zu maximieren, ist mit $\mathrm{Kor}(B, C) = 0.94$ erfüllt. Jedoch beträgt die Korrelation zwischen Knoten C des einen Clusters und Knoten A des anderen Clusters auch 0.94, wodurch nicht alle Korrelationen zwischen den Teilgraphen kleiner als t sind.

Die Argumentation bei einer Unterteilung in die Teilgraphen $\{A, C\}$ sowie $\{B\}$ erfolgt analog. Bei der Gruppierung $\{A, B\}$ und $\{C\}$ liegt innerhalb des Clusters $\{A, B\}$ mit 0.89 eine zu geringe Korrelation vor.

Für $i = 3$ stellt jeder Knoten einen Teilgraphen dar; die Kanten bezeichnen die Korrelationen zwischen den Teilgraphen. Jetzt sind zwei Korrelationen zwischen verschiedenen Teilgraphen größer als 0.9, nämlich $\mathrm{Kor}(A, C) = 0.94$ und $\mathrm{Kor}(B, C) = 0.94$. □

2.3.1. Algorithmus zur Dimensionsreduktion

Die Dimensionsreduktion wird mit Hilfe eines Graphen modelliert. Die Features stellen die Knoten dar, die Kantengewichte entsprechen den Korrelationen zwischen den Features.

Die Korrelation zwischen zwei Features, die in unterschiedlichen Clustern liegen, soll gemäß Forderung 1 klein sein, während die Korrelation zwischen Features des gleichen Clusters groß werden soll. Beide Forderungen können für gewisse Konstellationen nicht gleichzeitig erfüllt werden, wie mit Abbildung 2.3 gezeigt wurde.

2.3.1.1. Erster Ansatz mit Zusammenhangskomponenten

Beim ersten Ansatz soll die Korrelation zwischen Elementen unterschiedlicher Cluster kleiner als ein Schwellwert t sein; gleichzeitig soll die Clustergröße auch nicht größer als notwendig sein.

Das ist algorithmisch leicht umzusetzen. Aus dem Graphen werden alle Kanten mit Kantengewicht kleiner als der Schwellwert t gelöscht und Breitensuche angewandt, um die Zusammenhangskomponenten zu finden. Die genaue Umsetzung findet sich in Abschnitt 3.2.1.2.

Die Forderung 1 b, dass alle Korrelationen zwischen den Clustern betragsmäßig kleiner als ein bestimmter Schwellwert t sein sollen, ist bei Betrachtung der Zusammenhangskomponenten eines Graphen erfüllt. Da bei der Modellierung der Korrelationsmatrix als Graph nur dann eine Kante zwischen zwei Knoten existiert, wenn deren Korrelation größer als t ist, ist garantiert, dass es zwischen Knoten aus unterschiedlichen Zusammenhangskomponenten nur Korrelationen kleiner als t gibt.

Beobachtung 2. *Unter Berücksichtigung der in Forderung 1 b gestellten Bedingung stellt die Anzahl der Zusammenhangskomponenten in dem betrachteten Graphen eine untere Schranke für die Anzahl der Cluster dar.*

Beweis. Angenommen, es liegen in einer optimalen Clusterung weniger Cluster vor, als es Zusammenhangskomponenten gibt. Dann gibt es in mindestens einem Cluster

mindestens zwei Features, deren korrespondierende Knoten im Graphen in unterschiedlichen Zusammenhangskomponenten liegen. Da es zwischen den beiden Knoten keinen Weg gibt, kann ein Schnitt der Kardinalität Null gefunden werden, der die beiden Knoten trennt. Das bedeutet, der betrachtete Cluster kann entlang dieses Schnittes in zwei Teilcluster C_1 und C_2 zerlegt werden, sodass alle Paare (c_1, c_2), $c_1 \in C_1$, $c_2 \in C_2$, eine Korrelation kleiner als der Schwellwert t aufweisen. Die dadurch entstandene neue Aufteilung in Cluster ist – gemessen an der Bedingung in Forderung 1 b – besser. □

2.3.1.2. Dimensionsreduktion unter Berücksichtigung der Forderung 1

Der zweite Ansatz soll garantieren, dass die Korrelationen innerhalb der Cluster größer als ein Schwellwert t werden, wobei die Cluster nicht kleiner als notwendig sein sollen. Diese Forderung ist algorithmisch nicht ganz einfach umzusetzen. Man kann große vollständige Graphen suchen, indem man z. B. greedy-artig vorgeht und in jedem Schritt das Max-Clique-Problem löst.

Bei der „Luxusvariante“, die auf beide Bedingungen aus Forderung 1 Rücksicht nimmt, werden ausgehend von den Zusammenhangskomponenten minimale Schnitte im Graphen gefunden. Dabei wird am Anfang von jedem Kantengewicht der Schwellwert t subtrahiert. Wird ein Schnitt mit negativem Wert gefunden, so kann eine Trennung des Clusters entlang dieses Schnittes als Verbesserung betrachtet werden. Algorithmisch ist diese Variante sehr aufwändig, da das Min-Cut-Problem häufig zu lösen ist, das allein schon eine zeitintensive Implementierung mit sich zieht. Eine Beschreibung für diese Variante findet sich in Algorithmus 1.

Algorithmus 1 Erste Heuristik zur Clusterung der Features für einen bestimmten Schwellwert t

1. Berechne die Korrelationsmatrix der Features und ziehe von dieser Matrix den Schwellwert t ab. Speichere diese Matrix auf K ab.
2. Setze in K alle Einträge mit Wert kleiner Null auf Null und speichere die neue Matrix auf M ab.
3. Bestimme anhand der Matrix M die Zusammenhangskomponenten im korrespondierenden Graphen mit Breitensuche. Speichere diese Partitionierung der Knotenmenge (Features) in P ab.
4. Solange $P \neq \emptyset$, betrachte für eine zufällige Komponente Q aus P den zugehörigen Graphen anhand der Matrix K und bestimme den minimalen Schnitt U.
5. Falls $|U| < 0$, dann teile Q in die zwei Teilmengen Q_1 und Q_2 entlang des Schnittes U. Damit erhält man $P = P \setminus \{Q\} \cup \{Q_1, Q_2\}$. Gehe zu 4.
6. Falls $|U| \geq 0$, lösche die Komponente Q aus P, d. h. $P = P \setminus \{Q\}$ und gebe Q aus. Gehe zu 4.

Eine zusätzliche Möglichkeit besteht darin, die negativen Kantengewichte mit a sowie die positiven Kantengewichte mit $(1 - a)$ unterschiedlich zu gewichten. Für $a = 0$,

d. h. die negativen Kantengewichte werden mit Null gewichtet, existieren keine negativen Schnitte und man bekommt als Clusterung die Zusammenhangskomponenten. Bei $a = 1$ existieren keine positiven Kantengewichte und man erhält vollständige Graphen als Cluster.

2.3.1.3. Heuristischer Ansatz

Eine Heuristik zur Dimensionsreduktion, die einfach zu implementieren ist und kleine Cluster bevorzugt, ist in Algorithmus 2 angegeben. Als Input wird dem Algorithmus die Datenmatrix, die in den Zeilen die Objekte und in den Spalten die Variablen enthält, sowie der Schwellwert t übergeben.

Algorithmus 2 Weitere Heuristik zur Clusterung der Features für einen bestimmten Schwellwert t

1. Berechne und speichere für jedes Feature a die Anzahl der Features, mit denen es eine Korrelation größer als t aufweist, in `kor_matr[a]`. Speichere diese Features, die mit a eine Korrelation größer als t aufweisen, im Vektor `l_kor`.
2. Bestimme für alle Paare in `l_kor` die Korrelation zwischen den zugehörigen Features und speichere das Minimum in `kor_matr2[a]`.
3. Erstelle den Null-Vektor `selected_features`.
4. Wähle zuerst alle Features aus, die mit keinem anderen Feature eine höhere Korrelation als t haben, und setze den zugehörigen Eintrag in `selected_features` auf Eins (d. h. diese Features wurden ausgewählt).
5. **while** Es existieren noch Features, deren zugehöriger Eintrag in `selected_features` Null ist **do**
6. Bestimme all jene Features, für die der dazugehörige Eintrag in `kor_matr` minimal ist.
7. Wähle von diesen Features eines aus (bezeichne es mit j), dessen zugehöriger Eintrag in `kor_matr2` maximal ist. Setze `selected_features[j]` auf Eins.
8. Eliminiere alle zu j hoch korrelierten Features und setze für diese Features den Wert in `selected_features` auf $-j$.
9. Update von `kor_matr` (Vereinfachung von Schritt 1) und `kor_matr2` (wie in Schritt 2), gehe zu 5.
10. **end while**

Als Output gibt die Funktion eine Liste zurück. Der erste Eintrag beinhaltet die Liste `selected_features`, bei der die ausgewählten Features leicht zu erkennen sind, während der zweite Eintrag die Anzahl der selektierten Features zurückgibt. Die genaue Implementation von Algorithmus 2 wird in Abschnitt 3.2.1.1 beschrieben.

Algorithmus 2 gibt abhängig vom gewählten Schwellwert t eine unterschiedliche Anzahl von Clustern zurück. Ziel ist es nun, für jeden dieser Cluster einen Repräsentanten zu finden, der für den jeweiligen Cluster charakteristisch ist. Erst durch Bestimmung

solcher Repräsentanten ist die Dimensionsreduktion in den Daten vollständig durchgeführt.

Als Repräsentant wird jenes Feature ausgewählt, das zu allen anderen Features desselben Clusters im Mittel die höchste Korrelation aufweist. Zusätzlich zum Repräsentanten werden für jeden Cluster weitere Kenngrößen berechnet, die im Programmpaket R in Tabellenform wiedergegeben werden können:

1. Repräsentant des Clusters,
2. Anzahl der Features im Cluster,
3. die im Cluster vorkommende kleinste und größte Korrelation zwischen zwei unterschiedlichen Features desselben Clusters,
4. die mittlere Korrelation im Cluster zwischen zwei unterschiedlichen Features und
5. der Median der Korrelationen zwischen zwei unterschiedlichen Features des Clusters.

Wenn der Cluster nur aus einem einzigen Feature besteht, wird als Repräsentant automatisch dieses Feature angesehen und die Berechnung der Korrelationen für diesen Cluster (Punkte 3 – 5) ist überflüssig.

Mit diesen Kenngrößen lässt sich eine gute Einschätzung der gefundenen Cluster treffen. Erfahrungsgemäß liegt die mittlere Korrelation pro Cluster über dem gewählten Schwellwert t, obwohl dies für einzelne Paare desselben Clusters nicht unbedingt erfüllt sein muss. Es lässt sich genau bestimmen, wie viele Cluster Featurepaare mit einer Korrelation unter dem gewählten Schwellwert t enthalten. Algorithmus 2 ist in dieser Hinsicht sicher nicht optimal, findet jedoch bei dem in dieser Arbeit behandelten Datensatz nur wenige Cluster, die Forderung 1 a verletzen.

2.3.2. Verschiedene Modellierungsvarianten anhand eines Beispiels

Als Beispiel zur Modellierung der verschiedenen Algorithmen aus Abschnitt 2.3.1 wird ein verkleinerter Datensatz mit neun Variablen betrachtet. Wenn für die dazugehörige Korrelationsmatrix

```
      vec1  vec2  vec3  vec4  vec5  vec6  vec7  vec8  vec9
vec1  1.00  0.11  0.54  0.13  0.84 -0.32  0.68 -0.07  0.46
vec2  0.11  1.00  0.64 -0.14  0.12  0.01  0.25 -0.01  0.04
vec3  0.54  0.64  1.00 -0.06  0.75 -0.28  0.80 -0.38  0.37
vec4  0.13 -0.14 -0.06  1.00  0.24  0.66  0.18 -0.09 -0.28
vec5  0.84  0.12  0.75  0.24  1.00 -0.20  0.92 -0.25  0.26
vec6 -0.32  0.01 -0.28  0.66 -0.20  1.00 -0.09 -0.24 -0.57
vec7  0.68  0.25  0.80  0.18  0.92 -0.09  1.00 -0.26  0.11
```

```
vec8 -0.07 -0.01 -0.38 -0.09 -0.25 -0.24 -0.26  1.00 -0.45
vec9  0.46  0.04  0.37 -0.28  0.26 -0.57  0.11 -0.45  1.00
```

der Schwellwert für korrelierte Zufallsvariablen auf $t = 0.5$ gesetzt wird, resultiert die nachstehende Matrix, bei der ein Eintrag Eins ist, wenn die Korrelation betragsmäßig größer als $t = 0.5$ ist. Somit wird deutlich, zwischen welchen Variablen im Graphen eine Kante existiert. Der dazugehörige Graph ist in Abbildung 2.4 dargestellt.

```
     vec1 vec2 vec3 vec4 vec5 vec6 vec7 vec8 vec9
vec1    1    0    1    0    1    0    1    0    0
vec2    0    1    1    0    0    0    0    0    0
vec3    1    1    1    0    1    0    1    0    0
vec4    0    0    0    1    0    1    0    0    0
vec5    1    0    1    0    1    0    1    0    0
vec6    0    0    0    1    0    1    0    0    1
vec7    1    0    1    0    1    0    1    0    0
vec8    0    0    0    0    0    0    0    1    0
vec9    0    0    0    0    0    1    0    0    1
```

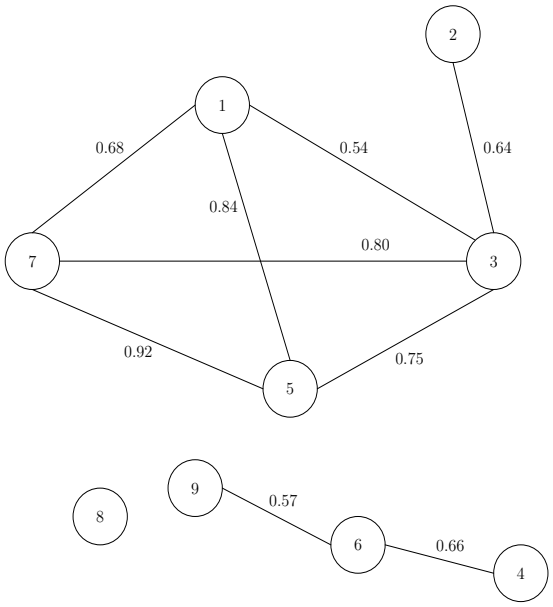

Abbildung 2.4.: Beispiel für einen Graphen, der aus einer Korrelationsmatrix erstellt wird

Die Zusammenhangskomponenten sind in diesem Fall leicht erkennbar. Diese sind in Abbildung 2.5 rot eingezeichnet. Für die Reduktionsvariante mit Zusammenhangskomponenten ergibt sich bei diesem Beispiel, dass die neun Variablen auf drei Cluster reduziert werden können. Die drei resultierenden Cluster sind $\{1, 2, 3, 5, 7\}$, $\{4, 6, 9\}$ und $\{8\}$. Mit den Zusammenhangskomponenten ist Forderung 1 b vollständig erfüllt.

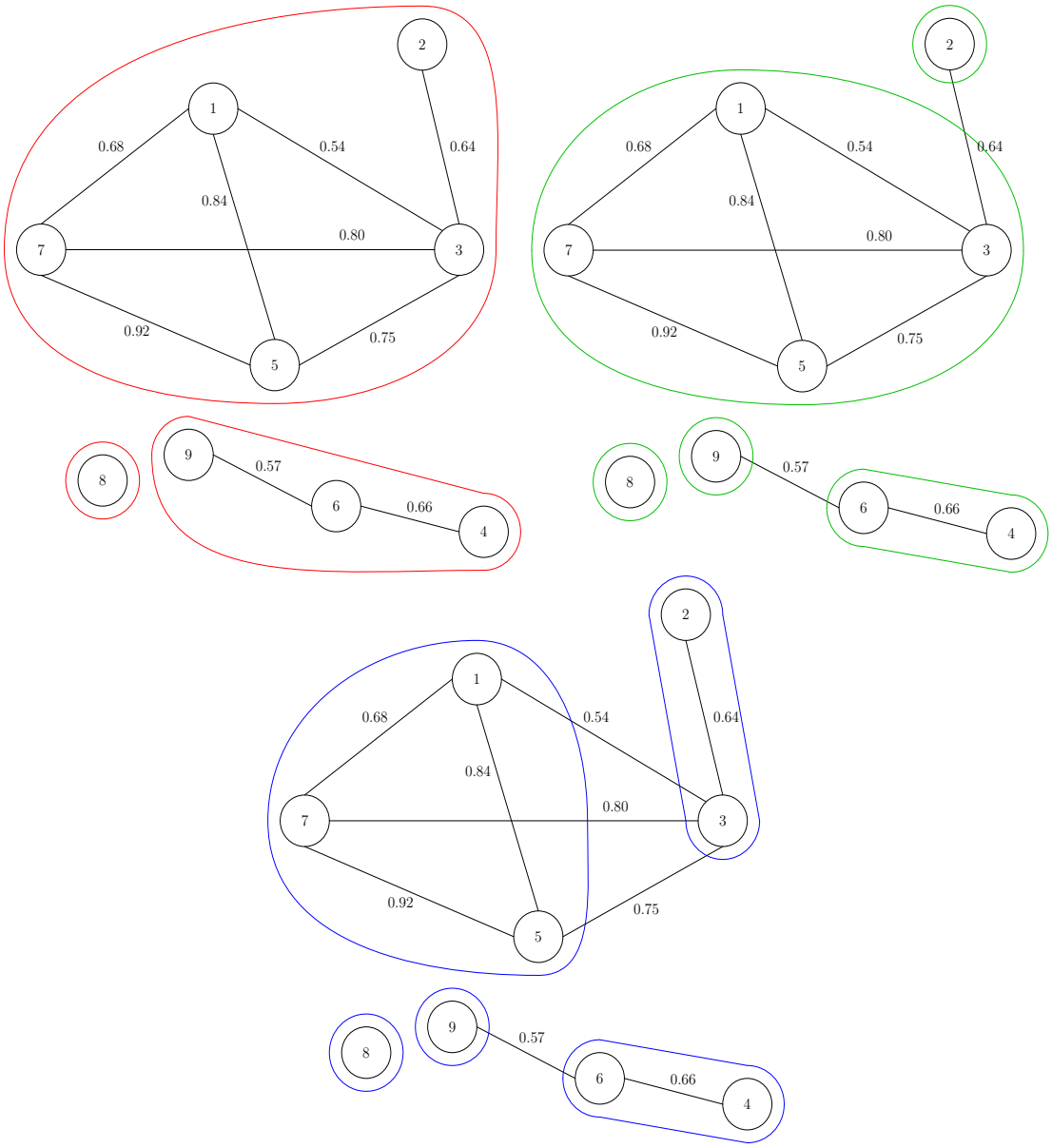

Abbildung 2.5.: Clusterlösungen des Beispielgraphen für die drei unterschiedlichen Algorithmusvarianten:
Variante mit Zusammenhangskomponenten: rote Cluster
Clusterbildung mit Algorithmus 1: grüne Cluster
heuristischer Korrelationsalgorithmus: blaue Cluster

Wird die Modellierung mit Algorithmus 1 durchgeführt, so resultieren in Abbildung 2.5 die grünen Cluster als gültige Partitionierung. Insgesamt erhält man bei der Variante aus Algorithmus 1 fünf Cluster, nämlich $\{1, 3, 5, 7\}$, $\{4, 6\}$, $\{2\}$, $\{8\}$ und $\{9\}$. Bei diesem Beispiel ist somit Forderung 1 a vollständig erfüllt.

Führt man die Clusterung gemäß Algorithmus 2 durch, so erhält man auch fünf unterschiedliche Cluster des Graphen aus dem obigen Beispiel. Die zugehörigen Cluster sind in Abbildung 2.5 blau eingezeichnet und lauten $\{1, 5, 7\}$, $\{4, 6\}$, $\{2, 3\}$, $\{8\}$ und $\{9\}$.

Dieses Beispiel demonstriert, dass die Anzahl der Zusammenhangskomponenten entsprechend Beobachtung 2 unter Berücksichtigung von Forderung 1 b tatsächlich eine untere Schranke für die Anzahl der Cluster darstellt.

2.4. Klassifikationsverfahren

2.4.1. PCA-Diskrimination

Die Hauptkomponentenanalyse zählt gemäß Antoniewicz u. a. (2006) zu den *unsupervised*-Methoden und wird dazu verwendet, die wirkliche Dimension der Daten zu erkennen, redundante Daten zu identifizieren und die Daten in einer reduzierten Dimension wiederzugeben.

Bei großen Datenmengen können sehr viele redundante Daten auftreten. Deshalb ist eine gute Beschreibung der Daten oft nur bei kleineren Dimensionen der Daten möglich. Die Hauptkomponentenanalyse wird als Datenreduktionsmethode dazu verwendet, um die wahre Dimension der Daten deutlich zu machen. Dies geschieht, indem ein kleinerer Dimensionsraum erzeugt wird, der von neuen künstlichen Variablen aufgespannt wird. Diese neuen Variablen stellen Linearkombinationen der ursprünglichen Variablen dar und erklären so viel Varianz wie möglich.

Die folgende Beschreibung der auf Pearson (1901) zurückgehenden Hauptkomponentenanalyse orientiert sich an Pruscha (2006) und Fahrmeier u. a. (1996).

Metabolomics-Daten können in multivariater Form durch eine $(n \times p)$-Datenmatrix $X = (x_1, \ldots, x_p)$ angegeben werden, wobei n die Anzahl der Objekte (Beobachtungen) und p die Anzahl der Variablen (Metaboliten, Features) bezeichnet. Dabei ist der n-dimensionale Vektor der i-ten Variablen durch

$$x_i = \begin{pmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{pmatrix}, \quad i = 1, \ldots, p,$$

gegeben. Dies führt zur Darstellung

Var. x_1	Var. x_2	$\dots$	Var. x_i	$\dots$	Var. x_p
x_{11}	x_{12}	$\dots$	x_{1i}	$\dots$	x_{1p}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
x_{k1}	x_{k2}	$\dots$	x_{ki}	$\dots$	x_{kp}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
x_{n1}	x_{n2}	$\dots$	x_{ni}	$\dots$	x_{np}

der $(n \times p)$-Datenmatrix X.

Die p Variablen korrelieren bei Metabolomics-Daten typischerweise miteinander. Deshalb werden bei der Hauptkomponentenanalyse q ($q < p$) neue künstliche Variablen (die sogenannten Hauptkomponenten) eingeführt, die untereinander unkorreliert und somit redundanzfrei sind. Die ursprünglichen Variablen $x_1, \dots, x_p$ können als Linearkombinationen dieser neuen künstlichen – auch latent genannten – Variablen dargestellt werden. Bei der PCA werden die Hauptkomponenten (HK) gemäß ihres Erklärungsgehaltes für die Varianz der Reihe nach absteigend geordnet. Die erste Hauptkomponente erklärt den größten Anteil der Stichprobenvarianz, während die zweite, zur ersten unkorrelierte HK, einen maximalen Anteil der Restvarianz enthält, usw. Insgesamt kann bei der PCA die gesamte Varianzstruktur der p Variablen durch die p Hauptkomponenten beschrieben werden. Die ersten $q < p$ Hauptkomponenten beinhalten ein Maximum der Gesamtvarianz.

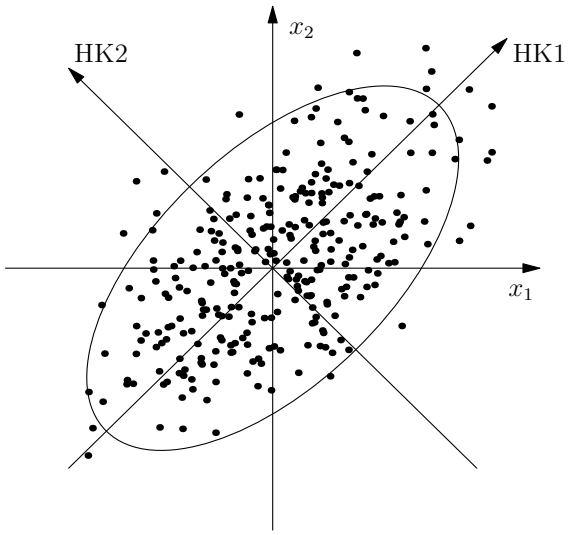

Abbildung 2.6.: Ellipsoide Punktwolke für den zweidimensionalen Raum

Geometrisch gesehen sind die q Hauptkomponenten die Hauptachsen durch die ellipsoidförmige Punktwolke, die durch n Beobachtungen im q-dimensionalen Raum gebildet wird. Bei multivariat normalverteilten Beobachtungen nimmt die Punktwolke

tatsächlich so eine mehrdimensionale ellipsoide Form an. Grafisch veranschaulicht wird die ellipsoide Punktwolke für den q-dimensionalen Raum mit $q = 2$ in Abbildung 2.6 (Pruscha, 2006, S. 268 ff.).

Die $(n \times p)$-Datenmatrix X kann bei der Hauptkomponentenanalyse aus den Hauptkomponenten (den *Scores*) y sowie der *Ladungsmatrix* Λ rekonstruiert werden. Im Allgemeinen werden $q \leq \min(n, p)$ HK für die Berechnung verwendet. Für $q = \min(n, p)$ gilt

$$\underset{(n\times p)}{X} = \underset{(n\times q)}{Y} \cdot \underset{(q\times p)}{\Lambda^T}$$

als Darstellung der PCA in Matrixschreibweise. Die Matrix $Y = X \cdot \Lambda$ stellt die Score-Matrix dar, während Λ die Ladungsmatrix angibt. Bei der Berechnung der Hauptkomponenten in den Abschnitten 2.4.1.1 und 2.4.1.2 wird davon ausgegangen, dass ebenso viele Hauptkomponenten q extrahiert werden, wie es Variablen p gibt.

Werden nur wenige HK für die Berechnung verwendet, so resultiert die sogenannte *approximierte* X-Matrix, die ein geringeres Rauschen aufweist. Würden alle möglichen $\min(n, p)$ HK für die Berechnung verwendet werden, so entspräche die Fehlermatrix E der Nullmatrix. Es gelten folgende Zusammenhänge:

$$X_{\text{appr}} = Y \cdot \Lambda^T, \quad X = Y \cdot \Lambda^T + E, \quad E = X - X_{\text{appr}}.$$

Grafisch wird der Zusammenhang zwischen der ursprünglichen Datenmatrix X und der approximierten Matrix X_{appr} bei Verwendung von q HK in Abbildung 2.7 veranschaulicht (vgl. Varmuza und Filzmoser, 2009, S. 62). Dabei bezeichnet Λ die Ladungsmatrix, Y die Matrix der PCA-Scores und E die Fehlermatrix.

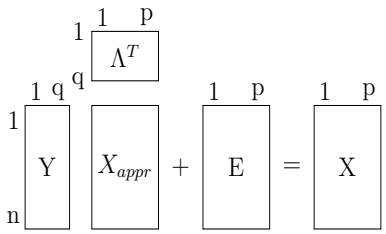

Abbildung 2.7.: Veranschaulichung der Zusammenhänge bei einer PCA

Bei der Hauptkomponentenanalyse können grundsätzlich zwei Berechnungsmethoden unterschieden werden:

- die Berechnung mit Hilfe der Kovarianzmatrix und
- die Berechnung anhand der Korrelationsmatrix.

Die Hauptkomponenten sind bei beiden Berechnungsarten im Normalfall bis auf das Vorzeichen eindeutig bestimmt, aber nicht identisch. Hierzu ist festzuhalten, dass bei der Berechnung mit der Korrelationsmatrix eine Standardisierung erfolgt. Dies geschieht, indem die Daten vor der Analyse auf einen Mittelwert von Null transformiert

und auf gleiche Varianz skaliert werden. Die Hauptkomponentenmethode ist *skalenabhängig*. Daher sind die aus den standardisierten Variablen gewonnenen Hauptkomponenten nicht identisch mit jenen aus den unstandardisierten Variablen. Es resultieren aus den zwei Berechnungsarten nicht dieselben Linearkombinationen und es gibt auch kein zweckmäßiges Verfahren, um die erste Linearkombination in die zweite Linearkombination überzuführen. Eine Hauptkomponentenanalyse mit der Kovarianzmatrix ist nur dann sinnvoll, wenn alle Variablen dieselbe interpretierbare Maßeinheit aufweisen und die Einheiten von Bedeutung sind. Wenn jedoch die Maßeinheiten der Variablen untereinander nicht vergleichbar sind, ist es naheliegend, standardisierte Variablen zu verwenden und die Hauptkomponentenanalyse mit der Korrelationsmatrix durchzuführen. Dadurch werden die Daten relativiert, die Einheiten verlieren an Bedeutung und die Hauptkomponentenanalyse wird auch bei nicht vergleichbaren Maßeinheiten anwendbar (Flury und Riedwyl, 1983, S. 119 ff.).

Zusätzlich zur Berechnungsmethode wird bei der Hauptkomponentenanalyse zwischen verschiedenen Analysetechniken unterschieden. Die gebräuchlichste Technik ist die *R-type*-PCA, bei der die Einträge der Korrelations- bzw. Varianz-Kovarianzmatrix für die Varianzen/Kovarianzen/Korrelationen *zwischen den Variablen* stehen. Die *R-type*-PCA wird immer dann angewandt, wenn ein Datensatz mit mehr Objekten als Variablen vorliegt. Dillon und Goldstein (1984) geben insgesamt sechs Alternativmethoden zur *R-type*-PCA an. Die am weitesten verbreitete Alternativmethode stellt die *Q-type*-PCA dar. Bei der *Q*-Analyse werden die Zeilen und Spalten der Datenmatrix vertauscht. Durch diese Transformation entsprechen die Elemente der Korrelationsmatrix den Korrelationen *zwischen den Individuen*. Dadurch ist eine Gruppierung der Individuen möglich und die PCA kann auch bei Datensätzen durchgeführt werden, bei denen mehr Variablen als Objekte vorliegen.

2.4.1.1. Berechnung der Hauptkomponenten aus der Kovarianzmatrix

Die Berechnung der Hauptkomponenten in den Abschnitten 2.4.1.1 und 2.4.1.2 orientiert sich an Pruscha (2006, S. 270 ff.).

Die Kovarianz zweier n-dimensionaler Vektoren $u = (u_1, \ldots, u_n)^T$ und $v = (v_1, \ldots, v_n)^T$ wird durch die empirische Kovarianz

$$\mathrm{Cov}(u, v) = s_{u,v} = \frac{1}{n-1} \sum_{k=1}^{n} (u_k - \overline{u})\,(v_k - \overline{v}), \quad k = 1, \ldots, n,$$

erwartungstreu geschätzt. Die Varianz ist durch

$$\mathrm{Var}(u) = s_u^2 = \frac{1}{n-1} \sum_{k=1}^{n} (u_k - \overline{u})^2$$

gegeben.

Für die $(n \times p)$-Datenmatrix $X = (x_1, \ldots, x_p)$ sei dann die $(p \times p)$-Kovarianzmatrix $S = \mathrm{Cov}(X, X) = (s_{ij})$ durch

$$s_{ij} = \mathrm{Cov}(x_i, x_j) = \frac{1}{n-1} \sum_{k=1}^{n} (x_{ki} - \overline{x}_i)\,(x_{kj} - \overline{x}_j), \quad i, j = 1, \ldots, p,$$

mit $\overline{x}_i = \frac{1}{n} \sum_{k=1}^{n} x_{ki}$ und $\overline{x}_j = \frac{1}{n} \sum_{k=1}^{n} x_{kj}$ gegeben. Die Kovarianzmatrix S wird als invertierbar vorausgesetzt.

Die p Eigenwerte $\lambda_j,\ j = 1, \ldots, p$, der positiv definiten Kovarianzmatrix S seien nach absteigender Größe geordnet:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p > 0.$$

Die Hauptachsen sind normierte, orthogonale p-dimensionale Vektoren

$$a_1 = (a_{11}, \ldots, a_{1p})^T, \ldots, a_p = (a_{p1}, \ldots, a_{pp})^T,$$

die die p Eigenvektoren der Kovarianzmatrix S bilden und Länge Eins aufweisen. Es ergeben sich die Eigenwertgleichungen

$$S \cdot a_j = \lambda_j a_j, \quad j = 1, \ldots, p, \qquad a_j^T \cdot a_{j'} = \begin{cases} 1 & \text{für } j = j' \\ 0 & \text{für } j \neq j' \end{cases}.$$

Das mit der k-ten Zeile der $(n \times p)$-Datenmatrix X assoziierte Objekt wird durch den $(p \times 1)$-Spaltenvektor $\overline{x}_k = (x_{k1}, \ldots, x_{kp})^T$ dargestellt. Mittels Skalarprodukt können die k-ten Einträge des zur j-ten Hauptkomponente gehörenden Score-Vektors y_j mit

$$y_{kj} = x_{k1}a_{1j} + x_{k2}a_{2j} + \ldots + x_{kp}a_{pj} = \overline{x}_k^T \cdot a_j, \quad k = 1, \ldots, n, \quad j = 1, \ldots, p,$$

berechnet werden. Für das k-te Zeilenobjekt ergeben sich somit die p Hauptkomponenten als lineare Funktionen von $\overline{x}_k^T$.

Mit der $(n \times p)$-Datenmatrix $X = (x_1, \ldots, x_p)$ werden diese p Hauptkomponenten als n-dimensionale Vektoren durch

$$y_1 = X \cdot a_1, \ldots, y_p = X \cdot a_p \tag{2.1}$$

realisiert.

Aus Gleichung (2.1) ergeben sich die Grundgleichungen der Hauptkomponentenanalyse in Matrixschreibweise

$$Y = X\Lambda,$$
$$X = Y\Lambda^T.$$

Die erste Gleichung gibt an, wie für eine Matrix X von Beobachtungsvektoren die dazugehörigen Vektoren der HK berechnet werden. Die zweite Gleichung stellt die Matrix X mit Hilfe der Score-Matrix Y der Hauptkomponenten dar.

Aufgrund der Orthogonalität der Ladungsmatrix $\Lambda = (a_1, \ldots, a_p)$ gilt:

$$\Lambda^T = \Lambda^{-1}.$$

Als wichtigste Eigenschaften der p HK $y_1, \ldots, y_p$ werden

$$\mathrm{Var}(y_j) = a_j^T \cdot S a_j = \lambda_j a_j^T a_j = \lambda_j, \quad j = 1, 2, \ldots, p,$$

$$\mathrm{Cov}(y_j, y_{j'}) = a_j^T S a_{j'} = \lambda_{j'} a_j^T a_{j'} = 0 \quad \text{für } j \neq j'$$

angegeben. Die Hauptkomponenten y_j, $j = 1, \ldots, p$, haben Varianz λ_j und sind unkorreliert.

Die erste Hauptkomponente ist daher jene Linearkombination der $x_1, \ldots, x_p$, die maximale Varianz besitzt. Die zweite HK ist zur ersten unkorreliert und weist maximale Restvarianz auf usw. Wegen der Unkorreliertheit der p HK $y_1, \ldots, y_p$ gilt

$$\mathrm{Var}(y_1 + \ldots + y_p) = \lambda_1 + \ldots + \lambda_p = \mathrm{Spur}(S).$$

Die erklärte Varianz der j-ten HK y_j, $j = 1, \ldots, p$, wird durch ihren prozentuellen Beitrag

$$\frac{\lambda_j}{\mathrm{Spur}(S)} \cdot 100$$

zur Gesamtvarianz angegeben. Die Korrelation der i-ten Beobachtungsvariablen x_i und der j-ten HK $y_j = X \cdot a_j$ berechnet sich mit

$$\mathrm{r}(x_i, y_j) = \frac{\mathrm{Cov}(x_i, y_j)}{\sqrt{\mathrm{Var}(x_i)\,\mathrm{Var}(y_j)}} = \frac{\sqrt{\lambda_j}}{s_{x_i}} a_{ji}, \quad s_{x_i} = \sqrt{s_{ii}}, \quad i, j = 1, \ldots, p,$$

da $\mathrm{Cov}(x_i, y_j) = \epsilon_i^T S a_j = \lambda_j a_{ji}$ (ϵ_i stellt den i-ten p-dimensionalen Einheitsvektor dar). Der skalierte Koeffizient a_{ji} gibt also die Korrelation der i-ten Variable x_i mit der j-ten HK y_j an.

2.4.1.2. Berechnung der Hauptkomponenten aus der Korrelationsmatrix

Wenn die Variablen x_i, $i = 1, \ldots, p$, unterschiedliche Dimensionen aufweisen, verwendet man für die Berechnung der HK anstelle der Kovarianzmatrix S die Korrelationsmatrix R. Die Korrelationsmatrix R ist die Kovarianzmatrix der standardisierten Variablen. Die standardisierten Variablen werden mit

$$x_{ki}^* = \frac{x_{ki} - \overline{x}_i}{s_{x_i}}, \quad k = 1, \ldots, n,$$

berechnet, sodass

$$x_i^* = \begin{pmatrix} x_{1i}^* \\ \vdots \\ x_{ni}^* \end{pmatrix}, \quad i = 1, \ldots, p,$$

als standardisierter Vektor der i-ten Beobachtungsvariablen gilt. Dabei gibt s_{x_i} die Standardabweichung der Variablen x_i an. Die Kovarianzmatrix der standardisierten Matrix $X^* = (x_1^*, \ldots, x_p^*)$ ergibt sodann die $(p \times p)$-Korrelationsmatrix R.

Es werden die Eigenwerte und Eigenvektoren der als invertierbar vorausgesetzten Korrelationsmatrix R analog zum vorherigen Abschnitt 2.4.1.1 berechnet. Die zugehörigen Hauptkomponenten $y_j,\ j = 1, \ldots, p$, erhält man auch in analoger Weise zum vorherigen Abschnitt.

Wird die PCA als spezielle Methode der Faktorenanalyse aufgefasst (bei der Faktorenanalyse werden $q < p$ HK extrahiert[8]), so

- wird bei der Berechnung anstatt der Kovarianzmatrix S die Korrelationsmatrix R verwendet und
- werden die Eigenvektoren a_j auf die Länge $\sqrt{\lambda_j}$ anstatt auf die Länge Eins normiert.

Die p positiven Eigenwerte $\lambda_j,\ j = 1, \ldots, p$, der Korrelationsmatrix R seien nach absteigender Größe geordnet und die Hauptachsen seien normierte p-dimensionale Vektoren

$$a_1 = (a_{11}, \ldots, a_{1p})^T, \ldots, a_p = (a_{p1}, \ldots, a_{pp})^T,$$

die die p orthogonalen Eigenvektoren der Korrelationsmatrix R bilden und Länge $\sqrt{\lambda_j},\ j = 1, \ldots, p$, aufweisen. Es ergeben sich die Eigenwertgleichungen

$$R \cdot a_j = \lambda_j a_j, \quad j = 1, \ldots, p, \qquad a_j^T \cdot a_{j'} = \begin{cases} \lambda_j & \text{für } j = j' \\ 0 & \text{für } j \neq j' \end{cases}.$$

Mit der standardisierten $(n \times p)$-Datenmatrix $X^* = (x_1^*, \ldots, x_p^*)$ werden die p Hauptkomponenten als n-dimensionale Vektoren durch

$$y_1 = \frac{1}{\lambda_1} X^* a_1, \ldots, y_p = \frac{1}{\lambda_p} X^* a_p \tag{2.2}$$

berechnet. Als Eigenschaften der HK werden

$$\mathrm{Var}(y_j) = 1, \qquad \mathrm{Cov}(y_i, y_j) = 0,\ i \neq j, \qquad \mathrm{Var}(y_1 + \ldots + y_p) = p = \mathrm{Spur}(R)$$

angegeben, d. h. die p Hauptkomponenten sind unkorreliert und haben Varianz Eins.

[8] Eine genaue Ausführung der Unterschiede zwischen Hauptkomponentenanalyse und Faktorenanalyse findet sich in Pruscha (2006, S. 278 ff.).

Die Korrelation des i-ten Vektors x_i^* der standardisierten Variablen und der j-ten Hauptkomponente y_j ist gleich der i-ten Komponente von a_j:

$$\mathrm{r}(x_i^*, y_j) = \mathrm{Cov}(x_i^*, y_j) = a_{ji}.$$

Es sei $\Lambda = (a_1, \ldots, a_p)$ die $(p \times p)$-Matrix der Hauptachsen, die die normierten Eigenvektoren a_j als Spalten enthält. Für die Ladungsmatrix Λ gelten folgende Eigenschaften:

- $\Lambda^T \cdot \Lambda = \mathrm{Diag}(\lambda_j)$ und
- $\Lambda \cdot \mathrm{Diag}\left(\frac{1}{\sqrt{\lambda_j}}\right)$ ist eine orthogonale Matrix.

Beide Eigenschaften sind wegen der Normierung der orthogonalen Eigenvektoren a_j auf Länge $\sqrt{\lambda_j}$ sofort ersichtlich. Aufgrund der Diagonalisierbarkeit der Korrelationsmatrix R lassen sich die Eigenwertgleichungen $Ra_j = \lambda_j a_j$ als

$$R \cdot \Lambda = \Lambda \cdot \mathrm{Diag}\left(\lambda_j\right)$$

schreiben und die Gleichung (2.2) der Hauptkomponenten lautet in Matrixschreibweise

$$Y = \mathrm{Diag}\left(\frac{1}{\lambda_j}\right) X^* \Lambda.$$

2.4.1.3. Aspekte der PCA

Die Hauptkomponentenanalyse basiert auf dem Konzept von Eigenvektoren und Eigenwerten. Die Linearkombination der ursprünglichen Variablen, aus der die j-te Hauptkomponente resultiert, hat Koeffizienten, die den Elementen des zu dem j-ten Eigenwert λ_j der Korrelationsmatrix R gehörenden Eigenvektor entsprechen. Da im Allgemeinen die Korrelationsmatrix R nicht bekannt ist, muss diese bei der PCA mit Hilfe einer gegebenen Stichprobe geschätzt werden. Typischerweise wird die empirische Korrelationsmatrix R (bzw. empirische Kovarianzmatrix S) verwendet.

Raykov und Marcoulides (2008) geben an, dass die Summe der Varianzen der ursprünglichen Variablen gleich der Summe der Varianzen der Hauptkomponenten ist und diese der Summe aller Eigenwerte der Korrelationsmatrix entspricht.

$$\mathrm{Var}\left(y_1\right)+\mathrm{Var}\left(y_2\right)+\ldots+\mathrm{Var}\left(y_p\right) = \mathrm{Var}\left(x_1\right)+\mathrm{Var}\left(x_2\right)+\ldots+\mathrm{Var}\left(x_p\right) = \lambda_1+\lambda_2+\ldots+\lambda_p.$$

Somit wird die Varianz in den Daten auf die Hauptkomponenten aufgeteilt. Der Quotient

$$r_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}$$

gibt den Anteil an der Gesamtvarianz an, den die erste Hauptkomponente erklärt. Analog dazu gibt der Quotient

$$r_q = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_q}{\lambda_1 + \lambda_2 + \cdots + \lambda_p} \tag{2.3}$$

den Anteil an der Gesamtvarianz an, den die ersten q ($q < p$) Hauptkomponenten enthalten.

2.4.1.4. Bestimmung der Anzahl der benötigten Hauptkomponenten

Bei der Hauptkomponentenanalyse gibt es verschiedene Kriterien, die besagen, wie viele Hauptkomponenten es zu extrahieren gilt.

Eine Möglichkeit besteht darin, die Hauptkomponenten so lange zu extrahieren, bis der Quotient in (2.3) einen bestimmten Anteil an erklärter Varianz übersteigt. Ein mögliches Ziel könnte z. B. sein, mindestens 80 % der Gesamtvarianz zu erklären, bevor keine weiteren Hauptkomponenten mehr bestimmt werden.

Kaiser (1960) besagt mit seinem Kriterium, dass keine Hauptkomponenten mehr extrahiert werden, sobald eine Hauptkomponente gefunden wird, deren Eigenwert kleiner als der Mittelwert der Eigenwerte der Kovarianzmatrix ist. Bei der Analyse mit Hilfe der Korrelationsmatrix gilt es demnach, so lange Hauptkomponenten zu extrahieren, solange die Korrelationsmatrix Eigenwerte größer als Eins aufweist. In diesem Fall sind aufgrund der Normierung die Varianzen aller *standardisierten* Variablen gleich Eins. Danach entspricht die Anzahl der zu extrahierenden HK der Zahl der HK mit Eigenwerten größer Eins. Der Eigenwert ist ein Maß für den Varianzerklärungsbeitrag der jeweiligen Hauptkomponente im Vergleich zur Varianz aller Variablen. Das Kaiser-Kriterium basiert auf der Tatsache, dass eine Hauptkomponente, deren Varianzerklärungsgehalt über alle Variablen gerechnet kleiner als Eins ist, weniger Varianz als eine einzelne Variable erklärt (Backhaus u. a., 2008).

Zusätzlich zu diesen beiden Kriterien gibt es den von Cattell (1966) eingeführten Scree-Plot. Dabei werden die Eigenwerte grafisch der Größe nach abnehmend dargestellt und die Werte durch Geraden verbunden. An der Stelle, an der die Differenz der Eigenwerte zwischen zwei Hauptkomponenten am größten ist, entsteht ein Knick (auch *Ellbow* genannt). Der Punkt im Knick gibt die Anzahl der zu extrahierenden Hauptkomponenten an. Die rechts vom Knick liegenden Eigenwerte nähern sich asymptotisch der Abszisse an. Hauptkomponenten mit den kleinsten Eigenwerten werden demnach für die Varianzerklärung als unbedeutsam angesehen. Ein Nachteil des Scree-Plots ist, dass er oft nicht eindeutige Aussagen liefert. Wenn die Eigenwerte ähnliche Differenzen aufweisen, lässt sich z. B. kein eindeutiger Knick ermitteln (Backhaus u. a., 2008).

2.4.1.5. Anteil erklärter Variabilität pro Feature

Das PCA-Modell ist durch die Gleichung

$$X = {}_qY \cdot {}_q\Lambda^T + {}_qE$$

gegeben, wobei ${}_qY$ und ${}_q\Lambda$ Matrizen mit q Spalten in Abhängigkeit von der Anzahl q der extrahierten Hauptkomponenten sind. Je mehr HK im Modell verwendet werden, desto besser ist die Approximation der Datenmatrix X und desto kleiner werden die Einträge der Fehlermatrix ${}_qE$.

Der folgende Abschnitt über den Anteil erklärter Variabilität pro Feature orientiert sich an Varmuza und Filzmoser (2009). Ziel dieses Abschnittes ist es anzugeben, wie gut jede Variable durch das Hauptkomponentenmodell bei Verwendung von q HK erklärt wird. Für jede Variable kann der Fehler durch die Summe der quadrierten Spaltenelemente der Fehlermatrix ${}_qE$

$$\sum_{i=1}^{n} {}_qe_{ij}^2 = \sum_{i=1}^{n} \left(x_{ij} - {}_q\overline{y}_i^T \cdot {}_q\overline{a}_j\right)^2, \quad j = 1, \ldots, p,$$

gewonnen werden. Dabei geben ${}_qe_{ij}$ die Einträge der Matrix ${}_qE$ an. Mit ${}_q\overline{y}_i$ wird die zum i-ten Objekt der Matrix ${}_qY$ gehörende Zeile als $(q \times 1)$-Spaltenvektor dargestellt und mit ${}_q\overline{a}_j$ die als $(q \times 1)$-Spaltenvektor geschriebene j-te Zeile der Matrix ${}_q\Lambda$ bezeichnet.

Wird diese Größe durch die Summe der quadrierten Spaltenelemente der Matrix X dividiert, ergibt dies ein Maß für die unerklärte Varianz jeder einzelnen Variable. Wird dieser Term von Eins subtrahiert, so resultiert das Maß ${}_q\Lambda_j^2$ für den erklärten Varianzanteil jeder Variablen bei Verwendung von q HK. Das Maß

$${}_q\Lambda_j^2 = 1 - \frac{\sum_i {}_qe_{ij}^2}{\sum_i x_{ij}^2}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, p,$$

für den Anteil erklärter Variabilität pro Variable bzw. Feature nimmt Werte zwischen Null und Eins an.

Für eine fix gewählte Anzahl an verwendeten HK ist es wünschenswert, dass jede Variable des Modells so gut wie möglich erklärt wird. Dennoch kann es vorkommen, dass einzelne Variablen im Gegensatz zu den übrigen Variablen eine nur sehr geringe erklärte Varianz aufweisen. Um dem entgegenzuwirken, sollte in so einem Fall die Anzahl der verwendeten HK im Modell erhöht werden.

In R lässt sich die erklärte Varianz für jede Variable mit der im Package `chemometrics` implementierten Funktion `pcaVarexpl` berechnen und grafisch durch einen Barplot veranschaulichen. Die Berechnung der erklärten Variabilität pro Feature anhand eines Beispiels findet sich in Abschnitt 3.3.8.

2.4.1.6. PCA-Diagnostik

Datenreduktion kann mit einer PCA am besten durchgeführt werden, wenn sich die (mehrdimensionalen) Daten ellipsoidförmig symmetrisch um ein Zentrum verteilen. Für stark schiefsymmetrische Datensätze ist die PCA als Werkzeug zur Dimensionsreduktion nicht gut geeignet. Im Allgemeinen erklären die HK bei schiefsymmetrischen Daten weniger an Varianz und auch die Krümmung kann nicht gut wiedergegeben werden. In solchen Fällen ist es empfehlenswert, die Daten vor der Dimensionsreduktion entsprechend zu transformieren, um im Anschluss mit der PCA eine Dimensionsreduktion durchzuführen.

Die Hauptkomponentenanalyse ist gegenüber Ausreißern in den Daten sensitiv. Die klassischen nicht-robusten Varianzmaße werden durch Ausreißer verzerrt. Da die PCA-Komponenten gemäß den Richtungen der maximalen Varianz extrahiert werden, werden sie logischerweise von Ausreißern beeinflusst. Um der Verzerrung entgegenzuwirken, empfehlen Varmuza und Filzmoser (2009), die Hauptkomponenten in Richtung von robusten Varianzmaßen zu maximieren, die von Ausreißern nicht verzerrt werden. Solche robusten Varianzmaße werden in Kapitel 2.2.1 als MCD-Schätzer eingeführt. Mit der robusten Variante erklären die HK die Variabilität in den Daten ohne die Ausreißer und stellen so eine zuverlässige Information dar.

Die Vorteile der robusten PCA können wie folgt zusammengefasst werden:

- Die resultierenden Richtungen der Ladungsvektoren sind wie bei der klassischen PCA zueinander orthogonal.
- Bei der robusten Variante wird im Gegensatz zur klassischen ein robustes Varianzmaß maximiert.
- Die Score-Plots der robusten PCA veranschaulichen die Struktur in den Daten besser als jene der klassischen Variante, da sie von Ausreißern weniger beeinflusst werden.
- Die Identifikation von Ausreißern erweist sich bei der robusten Variante mit sogenannten Diagnoseplots einfacher; bei der klassischen PCA werden oft nur extreme Ausreißer in den Daten erkannt.

Die robuste PCA-Variante unter Verwendung der MCD-Schätzer ist jedoch nur anwendbar, wenn zumindest doppelt so viele Objekte wie Variablen vorliegen. Liegen mehr Variablen als Objekte vor, ist laut Varmuza und Filzmoser (2009) eine robuste PCA mit Hilfe einer *Projection Pursuit*[9] durchzuführen. Die Idee dahinter lässt sich auf den klassischen Ansatz bei der PCA zurückführen. Es werden die Richtungen der Hauptkomponenten bestimmt, indem ein geeignetes Varianzmaß, unter Berücksichtigung der Orthogonalität zu den bereits vorher bestimmten HK, maximiert wird.

[9] *Projection Pursuit* ist ein statistisches Verfahren, bei dem mehrdimensionale Daten auf eine Hyperebene projiziert werden.

Beim *Projection-Pursuit*-Algorithmus wird die Richtung der Hauptkomponente entlang der maximalen robusten Varianz der projizierten Daten bestimmt. Eine ausführliche Beschreibung der robusten Hauptkomponentenanalyse in Verbindung mit dem Projection-Pursuit-Ansatz lässt sich bei Croux u. a. (2007) finden.

Es gibt zwei Arten von Ausreißern unter den Beobachtungen bzw. Objekten, zwischen denen bei der PCA unterschieden wird:

- orthogonale Ausreißer und
- Hebelpunkte.

Orthogonale Ausreißer sind Punkte, die eine große orthogonale Distanz zu dem von den Hauptkomponenten aufgespannten Raum aufweisen. Diese große Entfernung ist bei Projektion auf den PCA-Raum jedoch nicht sichtbar. Bei *Hebelpunkten* hingegen weist die Projektion auf den PCA-Raum eine große Entfernung vom Zentrum auf, was als große *Score-Distanz* bezeichnet wird. Bei den Hebelpunkten wird zusätzlich noch angegeben, ob es sich um *gute* oder *schlechte* Hebelpunkte handelt. Gute Hebelpunkte weisen zwar eine große Score-Distanz auf, haben jedoch eine kleine orthogonale Distanz zum PCA-Raum. Schlechte Hebelpunke weisen sowohl eine große Score-Distanz als auch eine große orthogonale Distanz auf und können die Schätzung des PCA-Raumes durch ihre sogenannte Hebelwirkung beeinflussen.

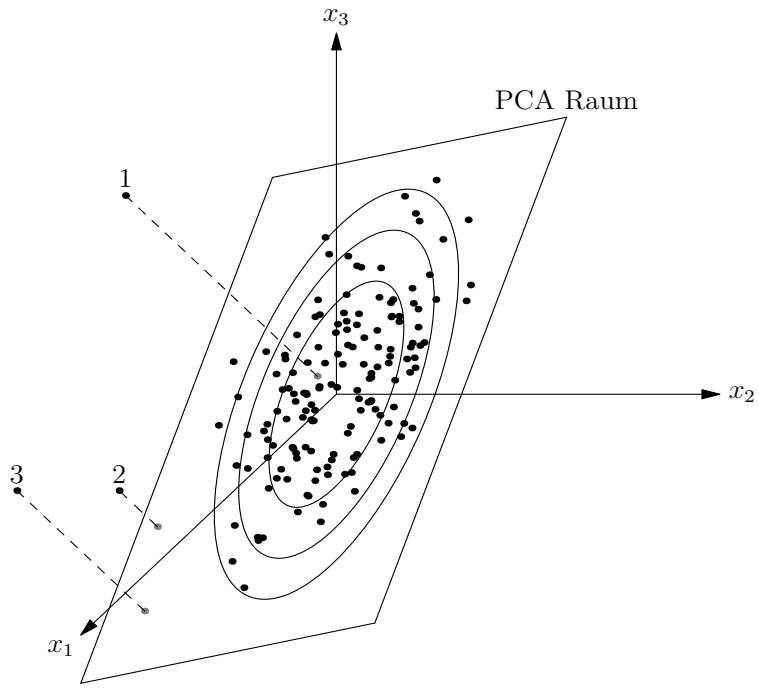

Abbildung 2.8.: Unterschiedliche Ausreißertypen bei einer PCA

In Abbildung 2.8 findet sich eine Darstellung der unterschiedlichen Ausreißertypen, die die klassische PCA beeinflussen können (vgl. Varmuza und Filzmoser, 2009). In dieser

Abbildung sind dreidimensionale Daten gegeben, deren Dimension auf den Raum der ersten zwei HK reduziert wird. Die meisten Objekte liegen ellipsoidförmig im PCA-Raum, der von den ersten beiden HK aufgespannt wird. Drei Objekte liegen jedoch auffallend weit weg. Objekt 1 weist eine große orthogonale Distanz zum PCA-Raum auf und stellt damit einen orthogonalen Ausreißer dar. Der dazugehörige projizierte Punkt im PCA-Raum fällt hingegen nicht auf und liegt nahe dem Zentrum. Objekt 2 stellt einen guten Hebelpunkt dar. Die Projektion dieses Punktes ist weit vom Zentrum entfernt (große Score-Distanz), die orthogonale Distanz ist dagegen minimal. Objekt 3 hat sowohl eine große Score-Distanz als auch eine große orthogonale Distanz und stellt daher einen schlechten Hebelpunkt dar.

Um Ausreißer in den Daten zu bestimmen, ist es nun naheliegend, die orthogonale Distanz sowie die Score-Distanz für jedes Objekt zu bestimmen. Die quadrierte Score-Distanz SD_i des i-ten Objektes wird mit

$$\mathrm{SD}_i^2 = \sum_{k=1}^{q} \frac{y_{ik}^2}{v_k}, \quad i = 1, \ldots, n,$$

berechnet, wobei q die Anzahl der HK angibt, die den PCA-Raum aufspannen, y_{ik} die Einträge der Score-Matrix Y darstellen und v_k die Varianz der k-ten HK angibt. Falls die Daten mehrheitlich multivariat normalverteilt sind, können die quadrierten Score-Distanzen durch eine Chi-Quadrat-Verteilung mit q Freiheitsgraden approximiert werden. Ein möglicher kritischer Wert für die quadrierte Score-Distanz wäre das 97.5-%-Quantil mit $\chi^2_{q,0.975}$. Alle Beobachtungen, deren quadrierte Score-Distanz über dem kritischen Wert liegt, werden als Ausreißer angesehen.

Die orthogonale Distanz OD_i des i-ten Objektes wird mit

$$\mathrm{OD}_i = ||\overline{x}_i - \Lambda \cdot \overline{y}_i||, \quad i = 1, \ldots, n,$$

berechnet, wobei $\overline{x}_i$ die zum i-ten Objekt gehörende Zeile der zentrierten Datenmatrix X als $(p \times 1)$-Spaltenvektor darstellt, Λ die $(p \times q)$-Ladungsmatrix der q HK ist und $\overline{y}_i$ die zum i-ten Objekt der $(n \times q)$-Score-Matrix Y gehörende Zeile als $(q \times 1)$-Spaltenvektor darstellt (bei der Verwendung von q HK).

Die Werte $(\mathrm{OD})^{\frac{2}{3}}$ verhalten sich annähernd normalverteilt, wobei μ und σ der Normalverteilung durch den Median und die absolute Medianabweichung (MAD) der Werte $(\mathrm{OD})^{\frac{2}{3}}$ geschätzt werden. Der kritische Wert für die orthogonale Distanz ergibt sich dann als

$$\left(\mathrm{med}(\mathrm{OD}^{\frac{2}{3}}) + \mathrm{MAD}(\mathrm{OD}^{\frac{2}{3}}) \cdot z_{0.975}\right)^{\frac{3}{2}},$$

wobei $z_{0.975} = 1.96$ das 97.5-%-Quantil der Standardnormalverteilung bezeichnet.

Der MAD einer Stichprobe gibt die absolute Medianabweichung (engl. absolute median deviation) an und berechnet sich mit

$$\mathrm{MAD}(x_1, \ldots, x_n) = \mathrm{med}\left(|x_i - \tilde{x}|\right) \cdot c,$$

wobei $\tilde{x}$ den Median der Stichprobe bezeichnet und die Konstante

$$c = 0.6745^{-1} = (z_{0.75})^{-1}$$

so gegeben ist, dass für eine standardnormalverteilte Stichprobe der MAD ein konsistenter Schätzer für die Standardabweichung σ ist (vgl. Varmuza und Filzmoser, 2009, S. 78 ff.).

Die praktische Berechnung der Score-Distanz und orthogonalen Distanz anhand eines Beispiels findet sich in Abschnitt 3.3.8.

2.4.1.7. Variablenselektion

Die Hauptkomponentenanalyse stellt eine Methode dar, um die Dimension im ursprünglichen Datenraum zu reduzieren und dabei nur möglichst wenig an Information zu verlieren. Al-Kandari und Jolliffe (2005) erwähnen, dass die HK als Linearkombinationen der Originalvariablen oft wesentlich schlechter zu interpretieren sind als die Ausgangsdaten. Außerdem gehen in die HK im Allgemeinen *alle* Variablen zu einem festgelegten Anteil ein.

Um eine einfache Interpretation zu erhalten, ist es aus diesem Grund häufig günstiger, anstelle von p Variablenkombinationen mit maximalem Informationsgehalt besser q der p Originalvariablen selbst auszuwählen, die für die gesamten Daten am repräsentativsten sind. Es gibt zahlreiche unterschiedliche Kriterien zur Variablenselektion, die bei Al-Kandari und Jolliffe (2005) nachzulesen sind. Einige Kriterien basieren auf den Ladungen der HK und ein paar ziehen die Korrelation zwischen den Variablen und den HK als Selektionskriterium heran. Nachfolgend sind zwei Kriterien zur Selektion von $q < p$ Variablen angegeben, die auf einer PCA basieren und die die Variablen anhand der Ladungen in der Ladungsmatrix Λ selektieren.

1. Die Ladungen aller Variablen der ersten q HK werden in aufsteigender Reihenfolge geordnet. Die q Variablen (ohne Wiederholung), die die höchsten Ladungen aufweisen, bleiben im Modell erhalten.

2. In der Ladungsmatrix Λ, die die Eigenvektoren der Varianz-Kovarianz- bzw. Korrelationsmatrix enthält, werden ihre Maximaleinträge bestimmt. Hat der j-te Spaltenvektor von Λ seinen größten Eintrag in Zeile i, so

   - wird die i-te Variable ausgewählt, wenn j klein ist ($j < q$) und die i-te Variable in Folge mit einem hohen Gewicht in die zugehörige j-te HK eingeht.

   - wird die i-te Variable vom Datensatz eliminiert, wenn j nahe bei p liegt (d. h. für die letzten $(p - q)$ HK werden jene Variablen eliminiert, die die größten Ladungen für diese HK aufweisen).

Um genau q Variablen auszuwählen, wird dieses Verfahren q-mal iterativ durchgeführt, bis genau q Variablen selektiert bzw. $(p-q)$ Variablen aus dem Datensatz entfernt wurden.

Eine andere Methode zur Variablenselektion mit Hilfe der Korrelationen wurde bereits in Abschnitt 2.3.1.3 mit Algorithmus 2 eingeführt. Mit den Korrelationen werden die Variablen zu Clustern geformt und von jedem Cluster eine Variable als Repräsentant ausgewählt.

2.4.2. PLS-Diskriminanzanalyse

Die Partial-Least-Squares-Diskriminanzanalyse (PLS-DA) ist ein weit verbreitetes statistisches Verfahren in der Chemometrik und stellt für die Klassifikation einen erheblichen Fortschritt dar. Sie ist ein lineares Regressionsverfahren und dient zum Finden von wenigen Linearkombinationen der Originalvariablen, den sogenannten latenten Variablen bzw. PLS-Hauptkomponenten. Der größte Vorteil von PLS-DA ist die anwendbare Flexibilität – vor allem bei Datensätzen, bei denen die Anzahl der Variablen die Anzahl der Objekte weit übersteigt. Die PLS-DA dient wie eine lineare Diskriminanzanalyse (LDA) oder eine quadratische Diskriminanzanalyse (QDA) als Klassifikator. Der Vorteil ist jedoch, dass die PLS-DA im Gegensatz zur LDA oder QDA auch Datensätze mit mehr Variablen als Objekten klassifizieren kann. Das Hauptaugenmerk liegt auf der richtigen Wiedergabe der Gruppentrennungen. Ziel dabei ist es, ein automatisiertes Modell zu erstellen, um die Unterschiede zwischen den Gruppen zu bestimmen (vgl. Brereton, 2010).

Der PLS-DA liegt eine Partial-Least-Squares-Regression (PLS-R) zugrunde, mit deren Hilfe eine Klassifikation der Objekte durchgeführt wird. Die Response-Matrix Y ist kategorial und gibt dabei die Klassenzugehörigkeit der Objekte an. Bei der PLS-DA werden jene PLS-Komponenten gesucht, die die beste Trennung der unterschiedlichen Klassen gewährleisten (Vinzi u. a., 2010).

Die PLS-DA versucht, zwischen der $(n \times m)$-Matrix X und der $(n \times k)$-Matrix Y einen passenden Zusammenhang zu finden. Die Matrix Y besteht nur aus den Werten 0 und 1, welche die Klassenindizes der Objekte der Matrix X darstellen. Das bedeutet, dass die Einträge von Y binär codiert sind. Binärcodierung wird meist in der Klassifikation verwendet, sobald $k > 2$ Klassen vorliegen. Für jedes Objekt wird in Folge eine y-Variable für jede Gruppe definiert, indem

$$y_{ij} = \begin{cases} 1 & \text{wenn Objekt } i \text{ in Gruppe } j, \\ 0 & \text{wenn Objekt } i \text{ kein Element von Gruppe } j \end{cases}$$

festgesetzt wird. Die resultierende $(n \times k)$-Matrix Y kann dann für die Klassifikation sowie für ein multivariates Regressionsmodell verwendet werden (Varmuza und Filzmoser, 2009).

Das Klassifikationsmodell wird durch Anwendung einer Partial-Least-Squares-Regression (PLS-R) auf die Matrix Y trainiert. Die ursprünglichen Variablen der Matrix X werden bei der PLS-R in neue latente Variablen, die PLS-Hauptkomponenten, umgerechnet, die schließlich für die Regression verwendet werden. Nachdem ein solches Klassifikationsmodell erstellt wurde, ist es möglich, sowohl die Klassenzuordnung für die Originaldaten vorherzusagen, als auch für unbekannte Daten Vorhersagen über die Gruppenzugehörigkeit zu treffen.

Während der Analyse wird auf die Matrizen X und Y jeweils eine Hauptkomponentenanalyse angewandt. Es resultieren die Gleichungen

$$X = TP^T + E,$$
$$Y = UQ^T + F,$$

wobei Q und P Ladungsmatrizen (vgl. Kapitel 2.4.1 der Hauptkomponentenanalyse) darstellen. Die Beziehungen und Dimensionen der Matrizen werden in Abbildung 2.9 veranschaulicht, wobei p die Anzahl der PLS-Hauptkomponenten angibt. Das Matrizenprodukt von T und P approximiert die Datenmatrix X, während das Matrizenprodukt von U und Q die Klassifizierungsmatrix Y annähert.

Der erklärte Varianzanteil jeder PLS-Hauptkomponente a lässt sich in Anlehnung an die erklärte Varianz der Hauptkomponenten bei der Hauptkomponentenanalyse berechnen. Dafür werden die Quadratsummen der Vektoren t_a und p_a der a-ten PLS-Komponente miteinander multipliziert (t_a und p_a bezeichnen Spaltenvektoren der $(n \times p)$-Matrix T bzw. der $(m \times p)$-Ladungsmatrix P):

$$\kappa_a = \left(\sum_{i=1}^{n} t_{ia}^2 \right) \left(\sum_{j=1}^{m} p_{ja}^2 \right).$$

Die Summe aller Werte κ_a für jede PLS-Hauptkomponente ungleich Null ergibt genau die Quadratsumme der Originaldaten. Im Gegensatz zur PCA muss der Wert von κ_a für nachfolgende Hauptkomponenten nicht notwendigerweise kleiner werden, da die PLS-Analyse nicht nur die Datenmatrix X, sondern auch Y für die Analyse einbezieht.

$$\underset{(n\times m)}{X} = \underset{(n\times p)}{T}\,\underset{(p\times m)}{P^T} + \underset{(n\times m)}{E}$$
$$U = B \cdot T \qquad \updownarrow$$
$$\underset{(n\times k)}{Y} = \underset{(n\times p)}{U}\,\underset{(p\times k)}{Q^T} + \underset{(n\times k)}{F}$$

Abbildung 2.9.: Grafische Veranschaulichung der Beziehung bei einer PLS

Die Matrizen T und U der Scores werden iterativ so bestimmt, dass zwischen ihnen eine maximale Korrelation herrscht. Dies geschieht, indem zwischen den Scores T

der latenten Variablen von X und den Scores U der latenten Variablen von Y das Regressionsmodell

$$U = BT$$

erstellt wird. Dabei ist B eine Diagonalmatrix und die Scores in T und U werden so bestimmt, dass sie maximal miteinander korrelieren. Die dazugehörigen Ladungen sind durch die Matrizen P und Q gegeben.

PLS ist ein Verfahren der Dimensionsreduktion, bei dem Richtungen im Raum von X gesucht werden, in denen X und Y maximal miteinander korrelieren und auch eine hohe Varianz vorhanden ist (vgl. Kapitel 2.4.1 der Hauptkomponentenanalyse).

In der Literatur gibt es zahlreiche Algorithmen für eine PLS-Analyse, bei denen im Allgemeinen nicht exakt dieselben Scores und Ladungen resultieren, obwohl die Vorhersagen für die Klassifikation übereinstimmen. Eine sehr gängige Möglichkeit zur Bestimmung der PLS-Komponenten ist iterativ über den *non-linear-iterative-Partial-Least-Squares*-Algorithmus (NIPALS). In der Literatur werden häufig auch die Bezeichnungen PLS1-Algorithmus sowie PLS2-Algorithmus verwendet, die beide auf dem NIPALS beruhen. PLS1 bedeutet, dass die Matrix Y nur aus einer Spalte besteht, welche für die PLS-Regression verwendet wird. Bei der PLS-DA besteht Y aus mehreren Spalten. In diesem Fall wird für die multikategorielle Klassifikation der PLS2-Algorithmus angewandt, der eine Zerlegung der Matrizen X und Y wie in Abbildung 2.9 liefert. Eine genaue Beschreibung des PLS2-Algorithmus findet sich in Brereton (2010, S. 298 ff.).

Bei der Bestimmung der PLS-Hauptkomponenten fließt im Gegensatz zur PCA sehr wohl Information über die Regressandenmatrix Y ein. Die PLS-Hauptkomponenten werden so bestimmt, dass X und Y miteinander korrelieren, indem die Kovarianz zwischen den Matrizen U und T maximiert wird.

2.4.2.1. Abgrenzung von PLS zu PCA

Der Unterschied bei der Berechnung der PLS-Komponenten und den Hauptkomponenten einer PCA besteht darin, dass die Berechnung der PLS-Komponenten sowohl auf der Matrix X als auch auf der Matrix Y basiert. Im Gegensatz dazu wird bei der PCA die Information für die neuen latenten Variablen nur aus der Datenmatrix X genommen. Bei der Hauptkomponentenanalyse wird die Varianz der Scores maximiert, während bei der PLS die Kovarianz zwischen den Scores und der Response maximiert wird. Das Ziel, Y über die Matrix X erklären zu können, entspricht genau der Maximierung der Kovarianz.

Wird außerdem ein Vergleich zwischen den aus der Hauptkomponentenanalyse gewonnenen Hauptkomponenten und den PLS-Hauptkomponenten angestellt, so liegt aufgrund der Optimalität der PCA-Hauptkomponenten im Sinne einer maximalen Varianzextraktion die erklärte Varianz der PLS-Hauptkomponenten stets unterhalb jener der PCA-Hauptkomponenten. Vor allem bei den ersten Komponenten können

erhebliche Unterschiede im Ausmaß der erklärten Varianz auftreten. Im Allgemeinen ist der Unterschied in Summe gesehen aber nicht allzu groß, da nach drei bis vier extrahierten Komponenten beide Methoden über 80 % der Information in X wiedergeben (Henrion und Henrion, 1995).

Der Vorteil von PLS liegt im Vergleich zur PCA darin, dass die Methode auch ohne Weiteres bei Datensätzen angewandt werden kann, bei denen mehr Variablen als Objekte vorliegen – wie es bei Metabolomics-Datensätzen meistens der Fall ist. Außerdem entsteht als Nebenprodukt der PLS-DA eine Klassifikation der Objekte, bei der schon die Gruppenzugehörigkeit eingeht. Bei der PCA muss z. B. noch eine Diskriminanzanalyse angewendet werden, um eine Klassifikation in Abhängigkeit der unterschiedlichen Gruppen zu erhalten.

2.4.2.2. Bestimmung der Anzahl der benötigten Komponenten

Um bei einem Klassifikationsmodell mit k Gruppen eine Trennung der verschiedenen Gruppen zu erhalten, sind $(k-1)$-dimensionale Hyperebenen notwendig. Daher werden bei einer PLS-DA zum Trennen von k Gruppen mindestens $k-1$ PLS-Hauptkomponenten benötigt. Als Abschätzung gilt, dass k Hyperebenen zumindest $k-1$ Mengen und maximal 2^k Mengen trennen.

Bei Modellen mit sehr vielen Variablen erweist sich die Wahl einer optimalen Anzahl von Komponenten als sinnvoll. Es gibt viele Möglichkeiten, die optimale Anzahl von extrahierten PLS-Hauptkomponenten zu bestimmen. Bei Verwendung von zu wenigen PLS-Komponenten geht signifikante Information verloren, während es bei Verwendung von zu vielen Komponenten zu einer Überanpassung des Modells und in Folge zu falschen Klassenvorhersagen kommen kann (Antoniewicz u. a., 2006).

In Klassifikationsmodellen wird üblicherweise jeder Klasse ein Integer-Wert zugeordnet. Wenn der Klasse A z. B. der Wert +1 zugeordnet ist, ist es naheliegend, dass ein Objekt dieser Klasse A zugeordnet wird, wenn die PLS-DA einen numerischen Wert nahe Eins liefert. Das Interesse besteht darin, eine passende Anzahl von PLS-Hauptkomponenten zu bestimmen, die den Prozentsatz der korrekt klassifizieren Objekte (%-CC) optimiert. Den größten Erfolg erzielt das Entfernen eines Teils des Datensatzes. Anschließend wird eine PLS-Analyse auf den verbleibenden Datensatz angewandt und beobachtet, wie gut die vom Datensatz entfernten Stichproben vorhergesagt werden. Die Signifikanz einer Hauptkomponente kann am besten mit der Vorhersage eines unbekannten Objektes getestet werden. Dahinter liegt ein ähnliches Prinzip wie zum Testen der Güte eines Modells. Brereton (2010) gibt als klassischen Ansatz zum Optimieren der Anzahl der verwendeten PLS-Hauptkomponenten die Kreuzvalidierung an, die in Abschnitt 2.5.2 beschrieben wird.

2.4.2.3. Ausreißerproblematik und Diagnostik

Um mit Ausreißern in den Daten umzugehen, gibt es bei der PLS-DA ähnliche Möglichkeiten wie bei der PCA. In Abschnitt 2.4.1.6 wurden die unterschiedlichen Ausreißertypen behandelt, die auch bei einer PLS-DA auftreten und das Modell bzw. die Vorhersage beeinflussen können. Um einer solchen Verzerrung durch Ausreißer vorzubeugen, ist es wiederum möglich, robuste Analysevarianten zu wählen. Bei der PLS-DA besteht die Möglichkeit, anstatt der empirischen Kovarianz als Schätzer für die Kovarianz zwischen den Scores der Matrix X und den Scores der Matrix Y eine robuste Schätzung zu verwenden.

Um Ausreißer in Bezug auf die unterschiedlichen Klassen zu bestimmen, erweist sich die Kreuzvalidierung, die in Abschnitt 2.5.2 eingeführt wird, am geeignetsten. Bei der Kreuzvalidierung kann genau ermittelt werden, aus welcher Klasse die Objekte stammen, die als Ausreißer auffallen, und in welche Klasse sie eingeordnet werden würden.

2.4.2.4. Variablenselektion

Oft liegt das Augenmerk auch darauf, welche Variablen für die Klassifikation bei der PLS-DA am einflussreichsten sind. Eine Feature- bzw. Variablenselektion ist sinnvoll, um jene Variablen zu bestimmen, die für die Klasseneinteilung die größte Verantwortung tragen. In der Biomedizin bzw. Chemometrik werden solche identifizierten Variablen *Biomarker* genannt.

Meist wird versucht, die Variablen in Zusammenhang mit dahinterliegenden chemischen Substanzen zu interpretieren. Mit einer PLS-DA-Analyse ist es möglich, chemische Substanzen zu bestimmen, die in einer Gruppe weiter verbreitet sind als in den anderen Gruppen. Häufig wird auch untersucht, ob es überhaupt solche Biomarker gibt, die zwischen den unterschiedlichen Gruppen variieren. Die Schwierigkeit bei einem mehrklassigen Datensatz liegt darin, dass es Biomarker gibt, die nicht nur für eine einzige Gruppe, sondern für eine spezielle Kombination von Gruppen charakteristisch sind.

Die Methoden der Variablenselektion stimmen für eine PLS-DA weitgehend mit den bei einer PCA angewandten Methoden aus Abschnitt 2.4.1.7 überein. Als Kriterium für die Variablenselektion könnte der (absolute) Wert der Koeffizienten der Ladungsmatrix herangezogen werden. Die Ladungen informieren über die Bedeutung der ursprünglichen Variablen in X für die neuen latenten Variablen bzw. HK. Im Unterschied zur PCA, bei der die Ladungen ausgehend von der Matrix X bestimmt werden, geben bei der PLS-DA die Ladungen sehr wohl auch eine Beziehung zu Y an (vgl. Brereton, 2010).

2.5. Modellvalidierung

2.5.1. Bestimmung der Modellgüte

Die folgende Beschreibung über die Bewertung der Modellgüte orientiert sich an Brereton (2010, S. 315 ff.).

Die Modellgüte eines Modells wird bei Klassifikationsverfahren durch die Fehlklassifikationsrate angegeben. Je geringer die Fehlklassifikationsrate ist, desto besser ist das verwendete Modell für die Vorhersage der Gruppenzugehörigkeit. Die Fehlklassifikationsrate lässt sich am besten durch die sogenannte Konfusionsmatrix bzw. Wahrheitsmatrix (engl. *confusion matrix*) angeben.

Im Zwei-Klassen-Fall wird mit der Wahrheitsmatrix zwischen *falsch positiver* und *falsch negativer* Klassifikation unterschieden. Jedes Objekt wird entweder korrekt vorhergesagt und der richtigen Gruppe zugeordnet oder nicht. Im zweidimensionalen Fall wird in den Spalten die tatsächliche Klassenzugehörigkeit der Objekte und in den Zeilen die vorhergesagte Klassenzugehörigkeit eingetragen. Es entsteht eine (2×2)-Kontingenztafel, deren mögliche Fälle in Tabelle 2.2 dargestellt wird. In dieser Tabelle wird die Zugehörigkeit zur Gruppe A als erwünscht (positiv) definiert.

	Gruppe A (wahre Klasse)	Gruppe B (wahre Klasse)	
Gruppe A (vorhergesagte Klasse)	richtig positiv (TP)	falsch positiv (FP)	TP + FP
Gruppe B (vorhergesagte Klasse)	falsch negativ (FN)	richtig negativ (TN)	FN + TN
	TP + FN	FP + TN	

Tabelle 2.2.: (2×2)-Kontingenztafel

Im *richtig positiven* sowie *richtig negativen* Fall stimmen die Vorhersagen, in den beiden anderen Fällen liegt ein Fehler vor. Im Speziellen ergeben sich folgende vier Kombinationen:

- *richtig positiv* (TP)*:* Anzahl der Objekte, die der Klasse A angehören und auch der Klasse A zugehörig klassifiziert werden,
- *richtig negativ* (TN)*:* Anzahl der Objekte, die der Klasse B angehören und auch Klasse B zugeordnet werden,
- *falsch positiv* (FP)*:* Anzahl der Objekte, die in Wirklichkeit der Klasse B angehören, aber fälschlicherweise der Klasse A zugeordnet werden,
- *falsch negativ* (FN)*:* Anzahl der Objekte, die in Wirklichkeit der Klasse A angehören, aber fälschlicherweise als der Klasse B zugehörig ausgewiesen werden.

Die Häufigkeiten jeder dieser vier Kombinationen aus tatsächlicher und vorhergesagter Klasse werden anschließend in die Wahrheitsmatrix eingetragen. Aus den Häufigkeiten lassen sich die Prozentsätze für die vier Möglichkeiten sowie die Likelihood-Ratios (LR) berechnen, wie sie in Tabelle 2.3 angegeben werden.

	A (wahr)	B (wahr)	Likelihood-Ratio	Wahrscheinlichkeit
A (vorherg.)	$\%\text{-TP} = \frac{100 \cdot \text{TP}}{\text{TP}+\text{FN}}$	$\%\text{-FP} = \frac{100 \cdot \text{FP}}{\text{FP}+\text{TN}}$	$\text{LR}^+ = \frac{\%\text{-TP}}{\%\text{-FP}}$	$P^+ = \frac{\%\text{-TP}}{\%\text{-TP}+\%\text{-FP}}$
B (vorherg.)	$\%\text{-FN} = \frac{100 \cdot \text{FN}}{\text{TP}+\text{FN}}$	$\%\text{-TN} = \frac{100 \cdot \text{TN}}{\text{FP}+\text{TN}}$	$\text{LR}^- = \frac{\%\text{-FN}}{\%\text{-TN}}$	$P^- = \frac{\%\text{-FN}}{\%\text{-FN}+\%\text{-TN}}$
%-CC d. Kl.	$\%\text{-CC}_A = \%\text{-TP}$	$\%\text{-CC}_B = \%\text{-TN}$		

Tabelle 2.3.: Berechnungsmöglichkeiten bei einer (2×2)-Kontingenztafel

Der gesamte Prozentsatz der richtig klassifizierten Objekte wird mit

$$\%\text{-CC} = 100 \cdot \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$$

berechnet.

Dieser gesamte %-CC unterscheidet sich im Normalfall von den $\%\text{-CC}_A$ und $\%\text{-CC}_B$ der einzelnen Klassen, wenn in den Klassen die Anzahl von Objekten unterschiedlich ist.

Mit *Sensitivität* wird die Wahrscheinlichkeit bezeichnet, mit der ein Objekt der Gruppe A auch tatsächlich der Gruppe A zugeordnet wird. Die *Spezifität* gibt die Wahrscheinlichkeit an, dass ein Objekt der Gruppe B auch wirklich als zur Gruppe B gehörig eingestuft wird.

Es gibt zwei verschiedene Arten von *Likelihood-Ratios* (LR), die häufig in der Medizin bzw. in klinischen Studien angewandt werden.

- Der *positive* Likelihood-Ratio $\text{LR}^+ = \text{LR}^A$ bezeichnet für eine Stichprobe das Verhältnis der Chance, dass die Stichprobe der Gruppe A zugeordnet wird, wenn sie in Wirklichkeit der Gruppe A angehört, zu der Chance für die Zuordnung zur Gruppe A, wenn die Stichprobe eigentlich aus der Gruppe B stammt. Der positive Likelihood-Ratio gibt somit das Verhältnis zwischen der Anzahl *richtig positiver* Klassifikationen und der Anzahl *falsch positiver* Klassifikationen an.

$$\text{LR}^+ = \frac{\text{Sensitivität}}{1-\text{Spezifität}} = \frac{\frac{\text{TP}}{\text{TP}+\text{FN}}}{\frac{\text{FP}}{\text{FP}+\text{TN}}} = \frac{\%\text{-TP}}{\%\text{-FP}}.$$

- Der *negative* Likelihood-Ratio $\text{LR}^- = \text{LR}^B$ gibt das Verhältnis zwischen *falsch negativen* und *richtig negativen* Klassifikationen an. Er ist ein Maß dafür, um

wie viel mal häufiger ein Objekt der Gruppe B zugeordnet wird, obwohl es aus Gruppe A stammt, als es B zugeordnet wird, wenn es tatsächlich aus Gruppe B ist.

$$\mathrm{LR}^- = \frac{1 - \text{Sensitivität}}{\text{Spezifität}} = \frac{\frac{\mathrm{FN}}{\mathrm{TP}+\mathrm{FN}}}{\frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}}} = \frac{\%\text{-FN}}{\%\text{-TN}}.$$

Mit dem positiven LR kann die Wahrscheinlichkeit

$$P^+ = \frac{\mathrm{LR}^+}{\mathrm{LR}^+ + 1} = \frac{\frac{\%\text{-TP}}{\%\text{-FP}}}{\frac{\%\text{-TP}+\%\text{-FP}}{\%\text{-FP}}} = \frac{\%\text{-TP}}{\%\text{-TP}+\%\text{-FP}}$$

angegeben werden, dass ein Objekt tatsächlich der Gruppe A angehört, wenn es zur Gruppe A klassifiziert wird. Analog dazu gibt die Wahrscheinlichkeit

$$P^- = \frac{\mathrm{LR}^-}{\mathrm{LR}^- + 1} = \frac{\%\text{-FN}}{\%\text{-FN}+\%\text{-TN}}$$

an, dass ein Objekt aus Gruppe A stammt, wenn es zur Gruppe B zugeordnet wird.

Wenn bei der Zuordnung mehr als zwei Klassen vorliegen, wird das Prinzip der Konfusionsmatrix auf eine $(G \times G)$-Matrix erweitert, wobei G die Anzahl der möglichen Klassen angibt. In der Diagonale werden die richtig klassifizierten Stichproben eingetragen. Da meist die Anzahl der Objekte in den Klassen unterschiedlich ist, werden nicht die Häufigkeiten in der Konfusionsmatrix eingetragen, sondern die dazugehörigen Klassen-Wahrscheinlichkeiten, die sich spaltenweise auf Eins summieren. Ein Beispiel für eine solche Konfusionsmatrix mit vier möglichen Klassen ist in Tabelle 2.4 dargestellt.

	A	B	C	D		A	B	C	D
A	60	11	0	7	A	80.00 %	18.33 %	0.00 %	7.37 %
B	10	45	3	0	B	13.33 %	75.00 %	4.29 %	0.00 %
C	2	4	58	3	C	2.67 %	6.67 %	82.86 %	3.16 %
D	3	0	9	85	D	4.00 %	0.00 %	12.86 %	89.74 %
$\sum$	75	60	70	95					

Tabelle 2.4.: Beispiel einer Konfusionsmatrix mit mehreren Klassen

Eine solche mehrdimensionale Konfusionsmatrix gibt Auskunft darüber, welche der Klassen häufig miteinander verwechselt werden und welche nicht. In Tabelle 2.4 ist erkennbar, dass die Klassen A und B öfters miteinander verwechselt werden als die Klassen C und D. Insgesamt ergibt sich ein %-CC von

$$\%\text{-CC} = \frac{60 + 45 + 58 + 85}{300} = \frac{248}{300} = 0.82667 \approx 82.7\,\%,$$

der sich auf alle im Datensatz vorkommenden Klassen bezieht. Für die einzelnen Klassen ergeben sich als Wahrscheinlichkeit der richtigen Vorhersage die jeweiligen klassenspezifischen %-CC-Werte mit

$$\%\text{-CC}_A = \frac{60}{75} = 80\,\%, \quad \%\text{-CC}_B = \frac{45}{60} = 75\,\%,$$
$$\%\text{-CC}_C = \frac{58}{70} \approx 83\,\%, \quad \%\text{-CC}_D = \frac{85}{95} \approx 89\,\%.$$

Bei der Bestimmung der Modellgüte gibt es bei mehrklassigen Datensätzen je nach Priorität der einzelnen Klassen mehrere Möglichkeiten. Sind alle Klassen für die Vorhersage gleich wichtig, so wird man als geeignetes Kriterium für die Modellgüte den gesamten %-CC-Wert zu maximieren versuchen.

Spielt lediglich die Vorhersage einer einzigen Klasse eine Rolle, so wird jenes Modell bevorzugt, das diese eine Klasse am besten vorhersagt. Dabei wird die Konfusionsmatrix für diese eine Klasse extra berechnet und danach angegeben, wie gut diese eine Gruppe im Gegensatz zum Rest der Daten vorhergesagt wurde. In Anlehnung an das Beispiel aus Tabelle 2.4 findet sich die dazugehörige Konfusionsmatrix der Klasse A in Tabelle 2.5.

	A	¬ A		A	¬ A
A	60	18	A	80 %	8 %
¬ A	15	207	¬ A	20 %	92 %

Tabelle 2.5.: Fortsetzung des Beispiels einer Konfusionsmatrix für die Klasse A

Es besteht auch die Möglichkeit, bei mehreren Klassen den Vorhersagefehler bzw. den Anteil korrekt klassifizierter Objekte mit Kosten zu gewichten. Seien a, b, c und d die jeweiligen Kostenparameter, dann besteht ein geeignetes Kriterium zur Bestimmung der Modellgüte in der Maximierung der Kostenfunktion

$$\%\text{-CC}_{\text{gew}} = a \cdot \%\text{-CC}_A + b \cdot \%\text{-CC}_B + c \cdot \%\text{-CC}_C + d \cdot \%\text{-CC}_D.$$

Je nach Wahl der Parameter a, b, c und d wird die Wahl des Modells in Abhängigkeit der Vorhersage der einzelnen Klassen getroffen. Wird ein Parameter Null gesetzt, so ist die dazugehörige Klasse bei der Vorhersage irrelevant. Werden die Parameter im vorigen Beispiel als prozentueller Anteil der Gruppengröße mit

$$a = \frac{75}{300}, \quad b = \frac{60}{300}, \quad c = \frac{70}{300}, \quad d = \frac{95}{300}$$

gewählt, so liefert dies genau die Maximierung der Gesamtanzahl von korrekt klassifizierten Objekten.

2.5.2. Kreuzvalidierung

Kreuzvalidierung wird häufig verwendet, um die Güte des Klassifikationsmodells zu testen und die Anzahl der im Modell verwendeten PLS-Hauptkomponenten zu optimieren. Dafür wird der Datensatz in Test- und Trainingsdaten geteilt und am Schluss eine Fehlerrate für die Vorhersage des Modells geliefert. Hierbei ist wichtig, dass das Objekt, dessen Zugehörigkeit vorhergesagt werden soll, niemals für die Entwicklung des Modells herangezogen wird. Für die Klassifikation wird dann jenes Modell verwendet, dass bei der Kreuzvalidierung den größten %-CC-Wert aufweist. Die Bestimmung eines passenden Modells anhand von Kreuzvalidierung durch Betrachtung der unterschiedlichen Fehlerprozentsätze wird in Kapitel 3.3.3 durchgeführt.

Zuerst wird der gesamte Datensatz in zwei Datensätze geteilt, nämlich in einen Trainingsdatensatz und einen Testdatensatz. Aufbauend auf dem Trainingsdatensatz wird ein passendes Modell erstellt, während der Testdatensatz ausschließlich dazu dient, die Güte des erhaltenen Modells zu testen.

Klassifizierungsprobleme sind für Metabolomics-Daten laut Westerhuis u. a. (2008) sehr aufwändig, da viele Variablen und nur wenige Objekte vorliegen. Oft können mehrere Lösungen gefunden werden, die die Datensätze in Klassen teilen. Häufig tritt jedoch der Fall ein, dass ein Modell zwar die Trainingsdaten gut klassifiziert, für zukünftige Stichproben jedoch eine schlechte Klasseneinteilung trifft.

Es gibt mehrere Methoden, um eine Kreuzvalidierung durchzuführen. Die am häufigsten verwendete ist laut Brereton (2010) die *Leave-One-Out*-Kreuzvalidierung (LOO). Bei dieser Methode wird jedes Objekt ein einziges Mal vom Datensatz entfernt und das Modell für die verbleibenden Objekte aufgestellt. Beispielsweise wird bei 100 Objekten eine PLS-DA auf 99 Objekte angewandt und ein passendes Modell aufgestellt. Anschließend wird analysiert, wie gut das verbleibende einzelne Objekt vorhergesagt wird. Die Vorgehensweise bei der LOO-Kreuzvalidierung für einen Datensatz mit I Objekten ist folgendermaßen:

1. Wähle zuerst Objekt 1 ($= i$) des Datensatzes aus.

2. Führe eine PCA- oder PLS-Analyse mit den verbleibenden $I-1$ Objekten durch (abhängig davon, welches Objekt i vom Datensatz entfernt wurde, gibt es unterschiedliche Scores- und Ladungsmatrizen).

3. Gib eine Gruppenzugehörigkeitsvorhersage für das i-te Objekt an.

4. Wähle das nächste Objekt im Datensatz, das ausgelassen werden soll, und gehe zu Schritt 2. Wiederhole diese Vorgangsweise, bis alle Objekte einmal vom Datensatz entfernt worden sind.

Die Signifikanz der einzelnen PLS-Hauptkomponenten wird am besten getestet, indem mit dem Modell noch unbekannte Stichproben vorhergesagt werden. Zu den Hauptaufgaben der PLS-Analyse zählt die Bestimmung jener im Modell benötigten PLS-Komponenten, die das beste Klassifikationsergebnis liefern. Der klassische Ansatz, um die Anzahl der benötigten PLS-Komponenten zu optimieren, ist durch Anwendung einer Kreuzvalidierung. Dabei wird eine Teilmenge der Daten entfernt, eine PLS-DA auf dem verbleibenden Datensatz angewandt und dann die Vorhersage der aus dem Datensatz entfernten Objekte betrachtet. Die Idee dahinter ist, dass die ersten HK wesentliche Informationen für die Struktur der Daten beinhalten, während die restlichen Hauptkomponenten hauptsächlich Rauschen darstellen. Brereton (2010) teilt die Hauptkomponenten in solche, die die Struktur des Modells wiedergeben (die ersten q Hauptkomponenten), und jene, die nur Rauschen modellieren.

Die Tatsache, dass sich der %-CC leicht durch ein oder zwei Objekte beeinflussen lässt, stellt einen Nachteil der Kreuzvalidierung dar. Das Klassifikationsmodell muss basierend auf dem Trainingsdatensatz optimiert werden. Bei iterativer Durchführung der Kreuzvalidierung werden wiederholt neue Trainingsdatensätze bestimmt, woraus unterschiedliche Modelle resultieren. Jede dieser unterschiedlichen Aufspaltungen in Trainings- und Testdatensätze liefert eine andere optimale Anzahl an PLS-Hauptkomponenten. Einen Nachteil stellt hier die Abhängigkeit des Klassifikationsmodells von den Objekten des Trainingsdatensatzes dar. Die Lösung hängt von der Iteration bzw. der Aufspaltung in Trainings- und Testdatensatz ab. Deswegen ist es laut Brereton (2010) schwer, eine stabile Lösung für die Anzahl der Hauptkomponenten zu finden.

3. Anwendung auf mehrklassige Metabolomics-Datensätze

3.1. Datenbeschreibung

Metabolomics-Daten sind in multivariater Form durch eine $(n \times p)$-Matrix gegeben, wobei n die Anzahl der Objekte und p die Anzahl an Variablen (Metaboliten, Features) bezeichnet. Bei den meisten Metabolomics-Experimenten sind laut Goodacre u. a. (2007) viel weniger Objekte als Metaboliten bzw. Variablen vorhanden.

Diese Arbeit behandelt zwei Datensätze SN10 sowie SN200, die jeweils $n = 60$ Objekte beinhalten, jedoch eine unterschiedliche Anzahl an Features (Variablen) aufweisen. Alle Features stellen kleinmolekulare Stoffe eines Naturproduktes dar. Die Filterung der ursprünglichen Anzahl von 1899 Features erfolgt mit Hilfe des Signal-Rausch-Verhältnisses (SN). Das SN-Verhältnis ist ein Maß für die Qualität der Messtechnik. Um bedeutende Information aus der Messung gewinnen zu können, muss sich die Messung deutlich vom Hintergrundrauschen abheben. Das bedeutet, dass der SN-Wert ausreichend groß sein sollte. Beim Datensatz SN10 wurden Peaks herausgefiltert, die ein Signal-Rausch-Verhältnis von über 10 aufweisen; damit sind für diesen Datensatz $p = 613$ Features übrig geblieben. Bei diesem Datensatz wurde nicht sehr stark selektiert, weshalb Störfaktoren wie Rauschen vorkommen können. Beim zweiten Datensatz SN200 liegt eine feinere Selektierung vor und es sind nur jene 243 Features gegeben, die ein größeres Signal-Rausch-Verhältnis als 200 haben.

Die 60 Objekte des in dieser Arbeit behandelten Datensatzes sind in acht Klassen mit unterschiedlichen Gruppengrößen aufgeteilt, die in Tabelle 3.1 angeführt sind. Es gibt zwei Klassen mit je 13 Objekten, vier Klassen mit je acht Objekten und zwei Klassen mit nur einem Objekt. Die Klassen werden für die statistischen Analysen in Abschnitt 3.3 von Eins bis Acht durchnummeriert. Eine genaue Aufschlüsselung der insgesamt 60 Objekte befindet sich in Anhang C.

Die Gruppe *Blank* beinhaltet Blindproben, die zeitgleich unter denselben Bedingungen extrahiert und untersucht werden. Dies dient der Berücksichtigung von Verunreinigungen durch Lösungsmittel und Chemikalien. Die QC-Messungen dienen als interne Qualitätskontrolle. Dabei wird die Geräteleistung bzw. -messung ständig geprüft. Die

Gruppennr.	Gruppenname	Größe	Gruppennr.	Gruppenname	Gruppengröße
1	Blank	13	2	QC	13
3	WAR	8	4	PAEKF	8
5	PAETG	8	6	CAS	8
7	PAE-CAS-WAR-TG	1	8	ACT	1

Tabelle 3.1.: Gruppengrößen des Datensatzes

PAE-CAS-WAR-TG-Gruppe ist ein Mischprodukt, das eine Ausreißer-Charge darstellt, und wird im Folgenden meist kurz MIX genannt. Die ACT-Gruppe bezeichnet das Konkurrenzprodukt.

Ziel dieser Arbeit ist es,

i) ein Modell für die Aufteilung der 60 Objekte zu finden, sodass die verschiedenen Gruppen möglichst gut wiedergegeben werden, und

ii) festzustellen, welche Features für die Gruppenaufteilung von Bedeutung sind.

3.2. Aufbereitung geeigneter Software-Routinen in R

In diesem Kapitel wird die Implementierung der Heuristik aus Algorithmus 2 zur Dimensionsreduktion für die Statistik-Software R erläutert. Während sich der vollständige Algorithmus in Anhang A befindet, werden in den folgenden Abschnitten im Speziellen die Vorgangsweise und die verwendeten R-Bibliotheken näher beleuchtet. Außerdem werden Möglichkeiten vorgestellt, eine PLS-DA sowie eine PCA in R für Datensätze durchzuführen, bei denen mehr Variablen als Objekte vorhanden sind.

Im Folgenden wird für `Funktionen` eine Schreibmaschinenschrift, für *Bibliotheken* eine kursive und für Parameter eine serifenfreie Schrift verwendet.

3.2.1. Dimensionsreduktion

3.2.1.1. Heuristischer Ansatz zur Dimensionsreduktion

Zur Berechnung einer Gruppierung der Features kann die selbstgeschriebene Funktion `select_features` genutzt werden, die unter anderem die Anzahl jener Features ausgibt, die nach einer solchen Gruppierung übrig bleiben. Die Heuristik verwendet für die Berechnung die Korrelationen zwischen den Features. Die gesamte Funktion `select_features` ist in Anhang A beschrieben.

```
> select_features <- function(datamatrix,treshold=0.9, m_features=0)
```

Als Input werden der Funktion die Datenmatrix datamatrix, der Schwellwert treshold für hoch korrelierte Zufallsvariablen sowie die Anzahl der verwendeten Features m_features übergeben. Die Datenmatrix soll in den Zeilen die Objekte und in den Spalten die Variablen enthalten. Dabei ist die Angabe der zwei letzten Parameter optional. Kennt man die Anzahl der Features explizit, empfiehlt es sich, diese über den Parameter m_features an die Funktion `select_features` zu übergeben. Wenn kein Eintrag als Input erfolgt, ist der Schwellwert für korrelierte Zufallsvariablen standardmäßig auf 0.9 gesetzt. Als Anzahl der Features wird die Spaltenanzahl der Datenmatrix verwendet. Der Schwellwert treshold für hoch korrelierte Zufallsvariablen kann verändert werden, womit die Größe der resultierenden Gruppierungen mitbestimmt werden kann. Setzt man den Schwellwert nahe Eins, so fasst die Funktion `select_features` eher kleinere Gruppen zusammen, während die Funktion bei einem Schwellwert nahe 0.5 eher größere Gruppen ausgibt.

Zu Beginn setzt die Funktion die Anzahl der maximalen Features max_features fest, falls der Parameter m_features der Funktion übergeben wurde. Anschließend werden der Vektor selected_features sowie zwei Vektoren kor_matr und kor_matr2 als Nullvektoren initialisiert. Der Vektor selected_features dient zur Markierung der für die Gruppenbildung selektieren Features. Am Anfang ist jeder Eintrag in selected_features auf Null gesetzt. Im Vektor kor_matr wird für jedes Feature die Anzahl der zu diesem Feature über dem Schwellwert korrelierten Features abgespeichert. Die Initialisierung von max_features, selected_features, kor_matr und kor_matr2 passiert in den Codezeilen 2 bis 10 in Anhang A.

Zunächst wird der Vektor kor_matr gebaut. Die genaue Vorgangsweise ist in den Codezeilen 11 bis 24 in Anhang A nachzulesen. Dafür durchläuft die Funktion zwei Schleifen. Sowohl der Parameter l der äußeren *while*-Schleife als auch der Parameter j der inneren *while*-Schleife werden für jedes Feature genau einmal durchlaufen. Für ein Featurepaar (l, j), $l \neq j$, bei dem sowohl Feature j als auch Feature l noch nicht betrachtet wurden, wird die dazugehörige Korrelation bestimmt. Falls die Korrelation des Featurepaares über dem Schwellwert treshold liegt, wird der Eintrag im Vektor kor_matr an der Stelle l um Eins erhöht. Der Vektor kor_matr speichert damit für alle Features ab, mit wie vielen anderen Features eine über dem Schwellwert treshold liegende Korrelation besteht. Im Vektor l_kor werden die zu dem Feature l hoch korrelierten Features abgespeichert, das heißt, dass diese Features mit Feature l eine Korrelation über dem Schwellwert treshold aufweisen.

Die Einträge des Vektors kor_matr2 sind um einiges rechenaufwändiger zu erhalten (siehe Codezeilen 25 bis 39 in Anhang A) und die Berechnung basiert auf den zu Feature l hoch korrelierten Features in l_kor. Hier berechnet die *while*-Schleife für alle Paare von Features in l_kor die Korrelation (also die Korrelationsmatrix des Vektors l_kor) und speichert jeweils das minimale Element iterativ auf min_kor ab. Die kleinste paarweise Korrelation zwischen den zu l korrelierten Features wird dann dem Vektor kor_matr2 an der Stelle l zugewiesen.

Als Nächstes werden in den Zeilen 40 und 41 die Parameter features_todo und features_count initialisiert. features_todo gibt die Anzahl der Features an, die noch nicht betrachtet wurden und noch zu bearbeiten sind. features_count dient am Schluss der Funktion als Rückgabeparameter und speichert die Anzahl der ausgewählten Features ab.

Im ersten Schritt werden in den Zeilen 42 bis 52 in Anhang A all jene Features betrachtet, die mit keinem anderen Feature eine Korrelation über dem Schwellwert treshold aufweisen, d. h. deren zugehöriger Eintrag in kor_matr gleich Null ist. Für jedes dieser Features wird der entsprechende Eintrag in selected_features auf Eins gesetzt, der dazugehörige Eintrag in kor_matr auf -1 sowie in kor_matr2 auf $+2$. Alle bereits betrachteten Features sind durch einen negativen Eintrag mit -1 des Vektors kor_matr erkennbar, da sie nun mit keinem anderen Feature mehr korrelieren. Die minimale Korrelation des Vektors kor_matr2 wird auf den fiktiven Wert $+2$ gesetzt, da Korrelationen ja nur Werte zwischen -1 und $+1$ annehmen können. Der fiktive Wert $+2$ verhindert, dass dasselbe Feature später nochmals ausgewählt werden kann. Der Vektor features_count wird um die Anzahl der Features, deren zugehöriger Eintrag in kor_matr gleich Null ist, erhöht. Analog dazu wird der Vektor features_todo um dieselbe Anzahl Features verringert.

Im nächsten Schritt wird die *while*-Schleife in den Zeilen 53 bis 99 in Anhang A so lange durchlaufen, bis alle Features bearbeitet wurden. In den Zeilen 54 bis 55 wird jenes Feature min_feature bestimmt, das von allen Features, die noch nicht ausgewählt wurden und den kleinsten Eintrag im Vektor kor_matr besitzen, die größte Korrelation im Vektor kor_matr2 aufweist. Die Idee dahinter ist, dass von den Features mit den wenigsten hoch korrelierten Features jenes Feature ausgewählt wird, bei dem auch die paarweisen Korrelationen zwischen den mit diesem Feature korrelierten Features möglichst groß sind. Mit dieser speziellen Auswahl sollen die Korrelationen innerhalb der später gebildeten Cluster erhöht werden.

In den Zeilen 56 bis 64 werden all jene zu min_feature hoch korrelierten Features j bestimmt, deren zugehöriger Eintrag in selected_features Null ist. Der dazugehörige Eintrag des Vektors selected_features an der Stelle j wird auf $-$min_feature gesetzt. Die Features j werden wegen Feature min_feature eliminiert. Im Vektor selected_features ist damit sofort ersichtlich, welche Features miteinander eine hohe absolute Korrelation aufweisen. Analog zur vorherigen Iteration erfolgt ein Update der Vektoren kor_matr, kor_matr2 sowie features_todo.

Im Anschluss daran werden in den Zeilen 65 bis 68 jene Features i bestimmt, die mit dem bereits eliminierten Feature j eine hohe Korrelation aufweisen. Da nun Feature i mit einem Feature weniger korreliert (j existiert ja nicht mehr), muss der dazugehörige Eintrag von kor_matr an der Stelle i um Eins verringert werden.

Genauso muss ein Update des Vektors kor_matr2 erfolgen, der das Maximum der minimalen Korrelationspaare pro Feature abspeichert. Dies geschieht in den Codezeilen 69 bis 99 in Analogie zu den Zeilen 25 bis 39. Es werden wiederum iterativ der Vektor l_kor

und das Element min_kor berechnet sowie die Einträge in kor_matr2 dementsprechend erneuert. Am Schluss jeder Iteration erfolgt ein Update der Vektoren selected_features, features_todo, features_count, kor_matr und kor_matr2.

Die Funktion `select_features` gibt am Ende die Anzahl der selektierten Features mit features_count sowie den Vektor selected_features zurück. Der Vektor selected_features hat an der Stelle von ausgewählten Features den Eintrag +1, alle übrigen Features haben als Eintrag die negative Nummer jenes Features, das der Grund für ihre Eliminierung war.

3.2.1.2. Dimensionsreduktion mit Zusammenhangskomponenten

Die durch die Zusammenhangskomponenten in einem Graphen[10] festgelegte Anzahl der Cluster stellt gemäß Beobachtung 2 aus Abschnitt 2.3.1.1 eine untere Schranke für die Anzahl der möglichen Cluster dar. Zugleich wird mit den Zusammenhangskomponenten die Forderung 1 b erfüllt, dass alle Korrelationen zwischen den Clustern unter einem bestimmten Schwellwert t liegen.

Eine Funktion zum Finden solcher Zusammenhangskomponenten bezüglich der Korrelationen in einer Matrix, wobei auf die Modellierung als Graphen zurückgegriffen wird, ist in Algorithmus 3 angegeben. Ausgehend von einem Feature i wird iterativ Breitensuche angewandt.

Als Input werden der Funktion `connect_componend` aus Algorithmus 3 die Datenmatrix datamatrix sowie der Schwellwert treshold für hoch korrelierte Zufallsvariablen übergeben.

In Zeile 3 von Algorithmus 3 wird mit max_features die Anzahl der im Datensatz vorkommenden Features bestimmt. korr_matr_round gibt in Zeile 4 die gerundete Korrelationsmatrix der Features an, wobei ein Eintrag der Korrelationsmatrix Eins ist, wenn die zugehörige Korrelation zwischen den Features über dem Schwellwert treshold liegt. In den Zeilen 5 bis 8 werden die Parameter queue, active, clusters und start_clusters initialisiert. Am Anfang werden alle Features mit dem Wert 1 als aktiv bzw. noch nicht bearbeitet markiert. Der Vektor queue folgt einer FIFO-Datenstruktur. In der queue werden beginnend bei einem Feature nacheinander alle dazu hoch korrelierten Features notiert.

Die *for*-Schleife in den Zeilen 9 bis 28 wird für alle Features durchlaufen. Ein noch aktives Features i wird sodann in der Datenstruktur queue hinten angefügt, als current_cluster abgespeichert, beim Vektor start_clusters vorne hinzugefügt und der dazugehörige Eintrag im Vektor active auf Null gesetzt. Der Vektor current_cluster soll alle Elemente des zurzeit betrachteten Clusters enthalten. Der Vektor start_clusters beinhaltet alle Features, bei denen ein neuer Cluster eröffnet wird.

[10] Entsprechend Abschnitt 2.3 auf Seite 12 kann die Korrelationsmatrix der Variablen als Graph modelliert werden.

Algorithmus 3 Algorithmus zum Finden von Zusammenhangskomponenten in einem Graphen

```
connect_componend<-function(datamatrix,treshold)
{
max_features<-length(datamatrix[1,])
korr_matr_round<-round(abs(cor(datamatrix))-treshold+0.5)
queue<-c()
active<-c(rep(1,max_features))
clusters<-c()
start_clusters<-c()

for i in 1:max_features do
  if active[i]==1 then
    queue<-c(queue,i)
    current_cluster<-c(i)
    active[i]<-0
    start_clusters<-c(start_clusters,i)

    while length(queue)>0 do
      consider<-queue[1]
      queue<-queue[-1]
      for j in 1:max_features do
        if (korr_matr_round[consider,j]==1) && (active[j]==1) then
          active[j]=0
          queue<-c(queue,j)
          current_cluster<-c(current_cluster,j)
        end if
      end for
    end while

    clusters<-c(clusters,current_cluster)
  end if
end for

return  list(start_clusters,clusters)
}
```

Solange sich noch Elemente in der queue befinden, wird die *while*-Schleife in den Zeilen 15 bis 25 ausgeführt. Das erste Element des Vektors queue wird auf den Parameter consider abgespeichert. Anschließend wird das erste Element von queue gelöscht. Vom Feature consider ausgehend wird nun *Breitensuche* durchgeführt. Dabei werden jene Elemente j bestimmt, die mit dem Element consider in der gerundeten Korrelationsmatrix korr_matr_round eine Korrelation von Eins aufweisen und bislang noch nicht betrachtet wurden. Die Elemente j werden als nicht mehr aktiv markiert (active $= 0$) und zum aktuellen Cluster current_cluster hinzugefügt. Außerdem werden die Elemente j an die queue hinten angehängt.

Im Vektor clusters werden jeweils die aktuellen Cluster, sofern eine vollständige Zusammenhangskomponente gefunden wurde, abgespeichert. Am Schluss gibt die Funktion `connect_componend` mit start_clusters die Startnummern der Cluster sowie mit clusters den vollständigen Vektor aller gefundenen Cluster zurück.

3.2.2. Partial-Least-Squares-Diskriminanzanalyse

Um eine PLS-DA in R durchzuführen, wird für den Metabolomics-Datensatz auf die bereits vordefinierte Funktion `plsda` aus der Bibliothek ***caret*** zurückgegriffen. Die zugehörige Signatur ist in Algorithmus 4 dargestellt. `plsda` wird verwendet, um ein PLS-Modell für die Klassifikation zu fitten. Dabei gibt ncomp die Anzahl der zu extrahierenden PLS-Hauptkomponenten an. Die Funktion `predict` kann die Klassenzugehörigkeit eines neuen Objektes mit dem vorhandenen PLS-Modell vorhersagen. Vorhersagen der Klassenzugehörigkeit können auch für Modelle mit weniger als ncomp Klassen erfolgen.

Algorithmus 4 Signatur `plsda`

plsda(x, y, ncomp = 8, probMethod = c(“softmax“,“Bayes“), prior = NULL, ...)
predict(object, newdata = NULL, ncomp = NULL, type = c(“class”,“prob“,“raw“))

Um eine PLS-DA durchzuführen, muss zuerst die Klassenzugehörigkeit der Objekte in eine Indikator-Matrix Y umgewandelt werden. Die Matrix Y besteht nur aus den Werten 0 und 1. Gehört das i-te Objekt der Klasse j an, so hat die Matrix Y an der Stelle $Y[i, j]$ den Wert 1. Alle übrigen Werte in Zeile i sind Null. Die Erstellung der Indikator-Matrix Y kann in R mit der Funktion `makeind` aus der Bibliothek ***BayesTree*** erfolgen. Der zugehörige R-Code ist nachfolgend angegeben.

```
> library(BayesTree)
> gruppen_num<-c(rep(1,13),rep(2,13),rep(3,8),rep(4,8),rep(5,8),rep(6,8),7,8)
> x1 <- as.factor(gruppen_num)
> xx1 <- as.data.frame(x1)
> num_matrix <- makeind(xx1)
```

```
> num_matrix
      x1.1 x1.2 x1.3 x1.4 x1.5 x1.6 x1.7 x1.8
 [1,]    1    0    0    0    0    0    0    0
 [2,]    1    0    0    0    0    0    0    0
 ...    ...
[13,]    1    0    0    0    0    0    0    0
[14,]    0    1    0    0    0    0    0    0
 ...         ...
[25,]    0    1    0    0    0    0    0    0
[26,]    0    1    0    0    0    0    0    0
 ...
[51,]    0    0    0    0    0    1    0    0
[52,]    0    0    0    0    0    1    0    0
 ...                             ...
[58,]    0    0    0    0    0    1    0    0
[59,]    0    0    0    0    0    0    1    0
[60,]    0    0    0    0    0    0    0    1
```

Die PLS-DA wird auf die standardisierte Datenmatrix angewandt. Das bedeutet, dass die Daten vor der Analyse in den Spalten bezüglich der Variablen normiert werden. Es wird der Mittelwert von den Daten subtrahiert und durch die Standardabweichung dividiert. Anschließend wird mit der Funktion `plsda` eine PLS-DA auf die standardisierte Datenmatrix datenmatrix_stand sowie die Klassen-Indikator-Matrix num_matrix angewandt. Der verwendete R-Code ist nachstehend angegeben.

```
> library(caret)
> cowda <- plsda(datenmatrix_stand,num_matrix,ncomp=8)
> summary(cowda)

Data:   X dimension: 60 108
        Y dimension: 60 8
Fit method: kernelpls
Number of components considered: 8
```

Die Eingabe

```
> predict(cowda,type="class")

 [1] x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.1 x1.2 x1.2
[16] x1.2 x1.2 x1.2 x1.2 x1.2 x1.2 x1.2 x1.2 x1.2 x1.2 x1.2 x1.3 x1.3 x1.3 x1.3
[31] x1.3 x1.3 x1.3 x1.6 x1.4 x1.4 x1.4 x1.4 x1.4 x1.4 x1.4 x1.4 x1.6 x1.5 x1.5
[46] x1.5 x1.5 x1.5 x1.5 x1.5 x1.6 x1.6 x1.6 x1.6 x1.6 x1.6 x1.6 x1.6 x1.7 x1.8
Levels: x1.1 x1.2 x1.3 x1.4 x1.5 x1.6 x1.7 x1.8
```

liefert eine Vorhersage der Klassenzugehörigkeit für die 60 Objekte zu den acht verschiedenen Klassen. Mit Hilfe der Funktion `predict` der Bibliothek *caret* werden die Vorhersagen der Klassenzugehörigkeit zu Wahrscheinlichkeitsvariablen transformiert, die Werte zwischen $[0, 1]$ annehmen und deren Zeilensummen 1 ergeben. Die Klasse mit der größten Klassenwahrscheinlichkeit wird mit der Funktion `predict` als Klasse vorhergesagt.

Die Modellvorhersagen der PLS-DA werden durch die (60×8)-Matrix der *fitted values* mit dem Befehl

```
> fitvalall_plsda <- cowda$fitted.values
```

aufgerufen. Da die *fitted values* auch für Modelle mit weniger als acht verwendeten Komponenten abgerufen werden können, kommt die (60×8)-Matrix insgesamt für eine bis acht Komponenten vor. Die Umwandlung der *fitted values* in Wahrscheinlichkeitsvariablen erfolgt durch Subtrahieren des Zeilenminimums und Dividieren der Differenz aus Zeilenmaximum und Zeilenminimum. Das Ergebnis wird im Anschluss durch die Zeilensumme dividiert. Der Code

```
> klassenzuordnung<-matrix(rep(0,360),ncol=6)
> for(i in 3:8) {
+ fitval_plsda<-fitvalall_plsda[1:60,1:8,i:i] # fitted values mit 8 HK
+ max_fitval<-apply(fitval_plsda,1,max)
+ min_fitval<-apply(fitval_plsda,1,min)
+ norm1_fitval<-(fitval_plsda-min_fitval)/(max_fitval-min_fitval)
+ sum_norm1<-apply(norm1_fitval,1,sum)
+ norm_fitval<-norm1_fitval/sum_norm1
+ klassenzuordnung[,i-2]<-max.col(norm_fitval)
+ i<-i+1
+ }
```

berechnet die Vorhersage der Klassenzugehörigkeit für die Verwendung von drei bis acht Hauptkomponenten. Häufig sinkt der Prozentsatz der Fehlklassifikationen, je mehr Hauptkomponenten im Modell verwendet werden.

Das PLS-DA Modell wird – wie bereits in Abschnitt 2.4.2 eingeführt – durch die Gleichungen

$$X = TP^T + E,$$
$$Y = UQ^T + F$$

beschrieben, wobei Q und P Ladungsmatrizen und T und U Score-Matrizen bezeichnen. Bei einem Datensatz mit 108 Features und 60 Objekten (dieser Datensatz wird in Abschnitt 3.3 eingeführt) haben die in der PLS-DA vorkommenden Matrizen bei einer Extraktion von acht Hauptkomponenten folgende Dimensionen:

$$X = (60 \times 108), \quad T = (60 \times 8), \quad P = (108 \times 8), \quad E = (60 \times 108),$$
$$Y = (60 \times 8), \quad U = (60 \times 8), \quad Q = (8 \times 8), \quad F = (60 \times 8).$$

In R ist es möglich, durch die Funktionsaufrufe in Tabelle 3.2 auf die verschiedenen Matrizen des PLS-DA-Modells zuzugreifen.

Mit dem Befehl `cowda$scores` ist es möglich, auf die (60×8)-Matrix der X-Scores zuzugreifen. Die Scores stellen die in den acht-dimensionalen Unterraum transformierten HK dar. Mit den Scores ist es möglich, die Unterschiede in den Klassen – sofern welche existieren – grafisch aufzuzeigen. Für die praktische Anwendung einer PLS-DA siehe Abschnitt 3.3.2.

Aufruf	Matrix	Aufruf	Matrix
cowda$scores	T	cowda$loadings	P
cowda$Yscores	U	cowda$Yloadings	Q
cowda$residuals	F		

Tabelle 3.2.: Funktionsaufrufe in R, um auf die verschiedenen Matrizen einer PLS-DA zuzugreifen

3.2.3. Kreuzvalidierung

Der vollständige R-Code der Kreuzvalidierung befindet sich in Anhang B. Bei der Kreuzvalidierung in Anhang B werden von den 60 Objekten per geschichteter Zufallsauswahl 16 Objekte ausgewählt, die als Testdatensatz dienen. Dabei werden von der Blank- und QC-Gruppe jeweils vier Objekte ausgewählt und von den restlichen 8-er Gruppen jeweils zwei Objekte. Die verbleibenden 44 Objekte inkl. den zwei Solo-Klassen ergeben dann den Trainingsdatensatz.

Als Parameter werden der Funktion `kreuzvalidierung_func` die Datenmatrix datamatrix sowie die Anzahl repetition der durchzuführenden Iterationen übergeben.

In den Codezeilen 7 bis 22 von Anhang B wird die Gruppenzugehörigkeit des Testdatensatzes sowie des Trainingsdatensatzes angegeben. Da für den Testdatensatz eine Auswahl der Gruppen mit $(4, 4, 2, 2, 2, 2)$ getroffen wird, bleiben in den Gruppen mit mehreren Objekten für den Trainingsdatensatz neun bzw. sechs Objekte übrig. Die zwei Solo-Klassen werden auch fix den Trainingsdaten zugeordnet.

Als Nächstes wird sowohl für die Testdaten als auch für die Trainingsdaten eine Indikatormatrix der Gruppenzugehörigkeit erzeugt. Dies geschieht in den Zeilen 23 bis 29 mit der bereits in Abschnitt 3.2.2 eingeführten Funktion `makeind` aus der Bibliothek *BayesTree*. In den Zeilen 30 bis 37 werden einige Hilfsvariablen indiziert.

Die *for*-Schleife beginnend in Zeile 38 wird insgesamt repetition-mal durchlaufen. In den Zeilen 39 bis 41 wird der Testdatensatz erzeugt. Dies geschieht mit der in R im Package *sampling* vorimplementierten Funktion `strata`, die eine Zufallsauswahl der unterschiedlichen Klassen in der vorgegebenen Größe trifft. Die Erzeugung des Testdatensatzes erfolgt durch den Befehl

```
> strata_data<-strata(ohne_solo,size=c(4,4,2,2,2,2),stratanames=c("Gruppe"),
                      method="srswor",description=FALSE)
```

Die Matrix ohne_solo wird aus der datamatrix gewonnen, wobei die Solo-Klassen aus dem Datensatz entfernt wurden und eine Spalte mit einem Indikator für die Gruppenzugehörigkeit hinzugefügt wurde. Der Parameter stratanames übergibt exakt diese

Gruppeneinteilungen. Der Parameter size gibt genau die Gruppengrößen für die Zufallsauswahl in der Reihenfolge, in der die Gruppen im Datensatz vorkommen, an. Die Methode srswor indiziert eine Zufallsauswahl ohne Zurücklegen.

Im Anschluss werden in den Zeilen 43 bis 46 mit der Auswahl der Funktion `strata` die Matrix der Trainingsdaten sowie die Matrix der Testdaten erzeugt. In den Zeilen 47 bis 50 sowie 67 bis 70 findet eine Standardisierung dieser beiden Matrizen statt.

In Bezug auf die Trainingsdaten wird beginnend mit Zeile 51 eine PLS-DA sowie eine Vorhersage der Klassenzuordnung durchgeführt. In Zeile 62 wird mit error_train der prozentuelle Anteil der falsch klassifizierten Objekte der Trainingsdaten ausgegeben. Die Zeilen 63 bis 66 geben die falsch klassifizierten Objekte der Trainingsdaten aus. Das erhaltene Modell wird beginnend mit Zeile 71 auf den Testdatensatz angewandt, wodurch eine Klasseneinteilung für die Testdaten vorhergesagt wird. In Zeile 92 wird wiederum mit error_test der Fehlerprozentsatz der falsch klassifizierten Objekte des Testdatensatzes ermittelt. In den Zeilen 93 bis 95 werden die falsch klassifizierten Objekte der Testdaten ausgegeben.

Die Objekte 34, 43, 50 und 58 zeigen sich bei der PLS-DA in Abschnitt 3.3.2 auffällig und werden deswegen in den Zeilen 98 bis 115 von Anhang B näher untersucht. Es wird ermittelt, wie oft der Fall auftritt, dass ein Element dieser vier Objekte in den Testdaten liegt und bei der Klassifikation der richtigen Klasse zugeordnet wird. Wenn ein solcher Fall eintritt, wird das zugehörige Element auf merke abgespeichert. Die Matrix matrix_special_test beinhaltet für jedes dieser Objekte merke und den zugehörigen Vektor der Testdaten.

Am Schluss gibt die Funktion `kreuzvalidierung_func` eine Liste mit mehreren Parametern zurück. Als die ersten beiden Parameter werden die Fehlerprozentsätze der Klassifikation bei den Trainings- sowie Testdaten zurückgegeben. Der dritte Parameter listet alle Objekte auf, die in den Testdaten falsch klassifiziert wurden. Als vierter Parameter werden nur mehr unterschiedliche falsch klassifizierte Objekte der Testdaten angezeigt. Der Parameter suspect_class gibt mit Nullen oder Einsen an, wie oft ein Element der Objekte 34, 43, 50 oder 58 in den Testdaten richtig klassifiziert wurde. Der sechste Parameter gibt die Matrix matrix_special_test und der siebente Parameter den Vektor merke zurück.

Die Kreuzvalidierung kann auch für andere Verhältnisse von Test- zu Trainingsdatensatz durchgeführt werden. Die Implementierung in Anhang B ist für das Verhältnis 16:44 angegeben. Für andere Verhältnisse – wie 8:52 und 32:28, die in Abschnitt 3.3.3 verwendet werden – ist die Funktion `kreuzvalidierung_func` in analoger Weise abzuändern. Die geschichtete Zufallsauswahl der Funktion `strata` wird auf die Größe des Testdatensatzes angepasst. Es kommt zu Änderungen bei den Parametern gruppen_num_train, gruppen_num_test, matrix_special_test, strata_data sowie fitval_plsda.

3.3. Anwendung der Methoden und Modelle auf einen Datensatz mit mehreren Klassen und Vergleich der verschiedenen Modellergebnisse

Bevor auf den mehrklassigen Datensatz multivariate statistische Verfahren angewandt werden, werden die Daten univariat untersucht. Zuerst wird durch Abbildung 3.1 gezeigt, ob beim SN10-Datensatz mit 613 Features durch den Mittelwert und die Standardabweichung der Objekte Unterschiede zwischen den einzelnen Gruppen oder Objekten auffallen.

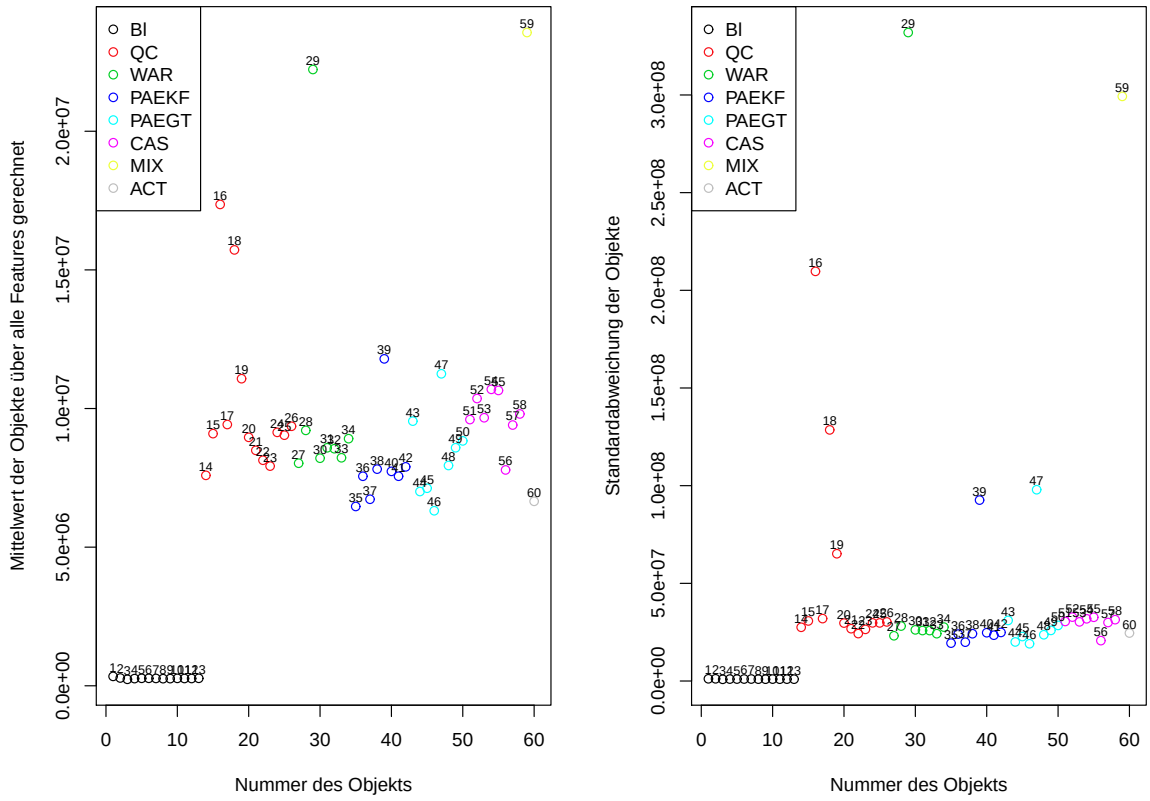

Abbildung 3.1.: Mittelwerte und Standardabweichungen der einzelnen Objekte des mehrklassigen Datensatzes

Die Blank-Objekte (Bl) haben alle einen Mittelwert und eine Standardabweichung nahe Null – was ja auch so beabsichtigt ist. In der Gruppe WAR fällt das 29. Objekt sowohl wegen des hohen Mittelwertes als auch wegen der extrem hohen Standardabweichung auf. Von den zwei Solo-Klassen sticht das MIX-Objekt auch wegen des extrem hohen Mittelwertes und der Standardabweichung ins Auge. Von der QC-Gruppe gibt es drei Objekte, die einen etwas höheren Mittelwert haben, ansonsten sind in

Abbildung 3.1 keine weiteren Auffälligkeiten erkennbar. Wird dieselbe Analyse des Mittelwertes und der Standardabweichung mit dem reduzierten SN200-Datensatz der 243 Features durchgeführt, so ist innerhalb der unterschiedlichen Gruppen sowohl beim Mittelwert als auch bei der Standardabweichung eine geringere Schwankung erkennbar. Dies deutet darauf hin, dass bei der feineren Selektion tatsächlich mehr Rauschen ausselektiert wurde und sich die Objekte derselben Gruppe im Mittelwert ähnlicher sind.

Insgesamt ist durch Abbildung 3.1 leicht erkennbar, dass noch weitere Analysen notwendig sind, um Unterschiede in den verschiedenen Gruppen festzustellen. Durch den Mittelwert und die Standardabweichung sind keine deutlichen Abweichungen zwischen den Gruppen ersichtlich. Univariate statistische Methoden sind nicht ausreichend, um Gruppenunterschiede aufzuzeigen. Aus diesem Grund werden im Folgenden (nach der Reduktion der Dimension) auf den Datensatz multivariate statistische Methoden angewandt, die das Feststellen von Gruppenunterschieden und Ausreißern erst ermöglichen.

3.3.1. Dimensionsreduktion

Durch Bestimmung der Zusammenhangskomponenten – wie in Abschnitt 2.3.1.1 beschrieben – wird gewährleistet, dass zwischen keinem Paar von Clustern eine Korrelation auftritt, die größer als der Schwellwert t ist. Bei einem Schwellwert $t = 0.9$ liefert Algorithmus 3 angewandt auf den SN10-Datensatz mit 613 Features insgesamt 108 verschiedene Cluster, von denen zwei Zusammenhangskomponenten mit 321 bzw. 117 Features groß sind. Alle übrigen Cluster beinhalten lediglich sieben oder weniger Features. In Abbildung 3.2 wird die Anzahl der resultierenden Cluster für Zusammenhangskomponenten in Abhängigkeit vom gewählten Schwellwert t für $t \in [0, 1]$ grafisch veranschaulicht.

Der resultierende Graph in Abbildung 3.2 verhält sich annähernd wie eine Exponentialfunktion. Bis zu einem Schwellwert von $t = 0.8$ steigt die Anzahl der Cluster kaum an, während sie für $t > 0.8$ regelrecht explodiert und exponentiell ansteigt.

Ein ähnliches Bild liefert auch die Heuristik aus Algorithmus 2. Nach Anwendung von Algorithmus 2 bleiben bei einem Schwellwert von $t = 0.9$ von den insgesamt 613 Features nur noch 212 Features[11] übrig. Wendet man Algorithmus 2 für 100 Schwellwerte $t \in [0, 1]$ an, so sieht auch hier der Zusammenhang zwischen dem Schwellwert und der Anzahl der verbleibenden Cluster exponentiell aus. Dieser Zusammenhang für die Heuristik aus Algorithmus 2 ist in Abbildung 3.3 rot eingezeichnet. Die schwarze Kurve gibt den Zusammenhang für die Zusammenhangskomponenten wieder, der auch schon in Abbildung 3.2 dargestellt wurde.

[11] Die Heuristik aus Algorithmus 2 liefert insgesamt 212 Cluster; durch die in Abschnitt 2.3.1.3 beschriebene Auswahl eines Repräsentanten für jeden Cluster lassen sich daher 212 Features der ursprünglichen 613 Features auswählen.

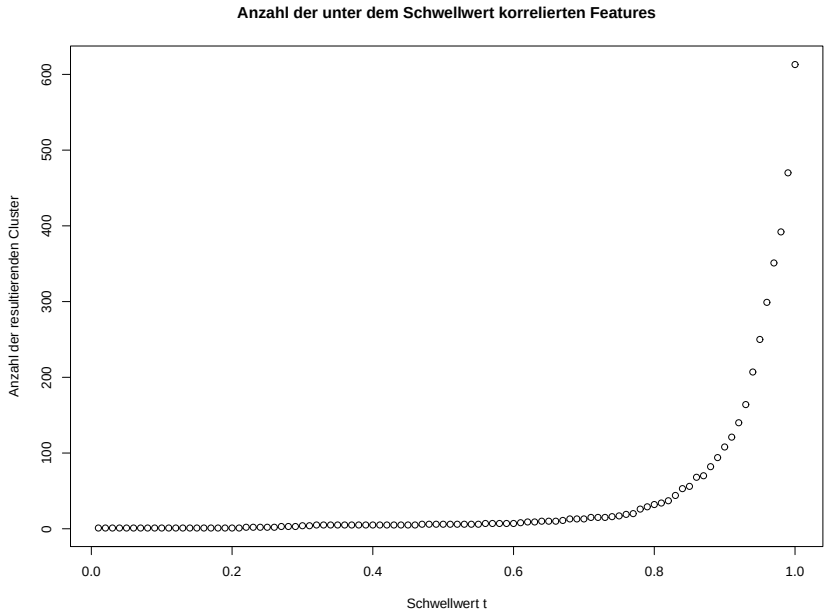

Abbildung 3.2.: Grafische Veranschaulichung der Abhängigkeit der resultierenden Clusteranzahl vom gewählten Schwellwert für den Algorithmus mit den Zusammenhangskomponenten

Die Heuristik aus Algorithmus 2 liefert bei einem Schwellwert von $t = 0.9$ insgesamt 212 verbleibende Cluster. Bei dem Algorithmus mit den Zusammenhangskomponenten sind es für $t = 0.9$ in Summe 108 Cluster. Abbildung 3.3 zeigt, dass der Algorithmus mit den Zusammenhangskomponenten – wie es Beobachtung 2 nahe legt – für alle Schwellwerte $t \in [0, 1]$ weniger Cluster als die Heuristik aus Algorithmus 2 liefert.

Wird der SN200-Datensatz der 243 Features mit Algorithmus 2 reduziert, so bleiben bei einem Schwellwert von $t = 0.9$ insgesamt 108 Features übrig. Es können beim selben Datensatz 60 Zusammenhangskomponenten gefunden werden.

3.3.2. Partial-Least-Squares-Diskriminanzanalyse

In den Abbildungen 3.4, 3.5 sowie 3.6 werden die Scores der ersten acht Hauptkomponenten der Partial-Least-Squares-Diskriminanzanalyse für den SN10-Datensatz mit 613 Features gegeneinander geplottet. Insgesamt liefern die acht Hauptkomponenten $\binom{8}{2} = 28$ unterschiedliche Grafiken, wenn die HK paarweise betrachtet werden.

In den Abbildungen 3.4 und 3.5 ist die gute Trennung der ersten acht Hautkomponenten erkennbar. Die vierte Hauptkomponente scheint am besten zu trennen, während bei der ersten Komponente die Trennung schlechter ersichtlich ist. Bei Betrachtung aller Plots lässt sich erkennen, dass die HK 2, 3, 4 und 5 die unterschiedlichen Gruppen im zweidimensionalen Raum am besten trennen. Auffallend ist oft das Feature 50

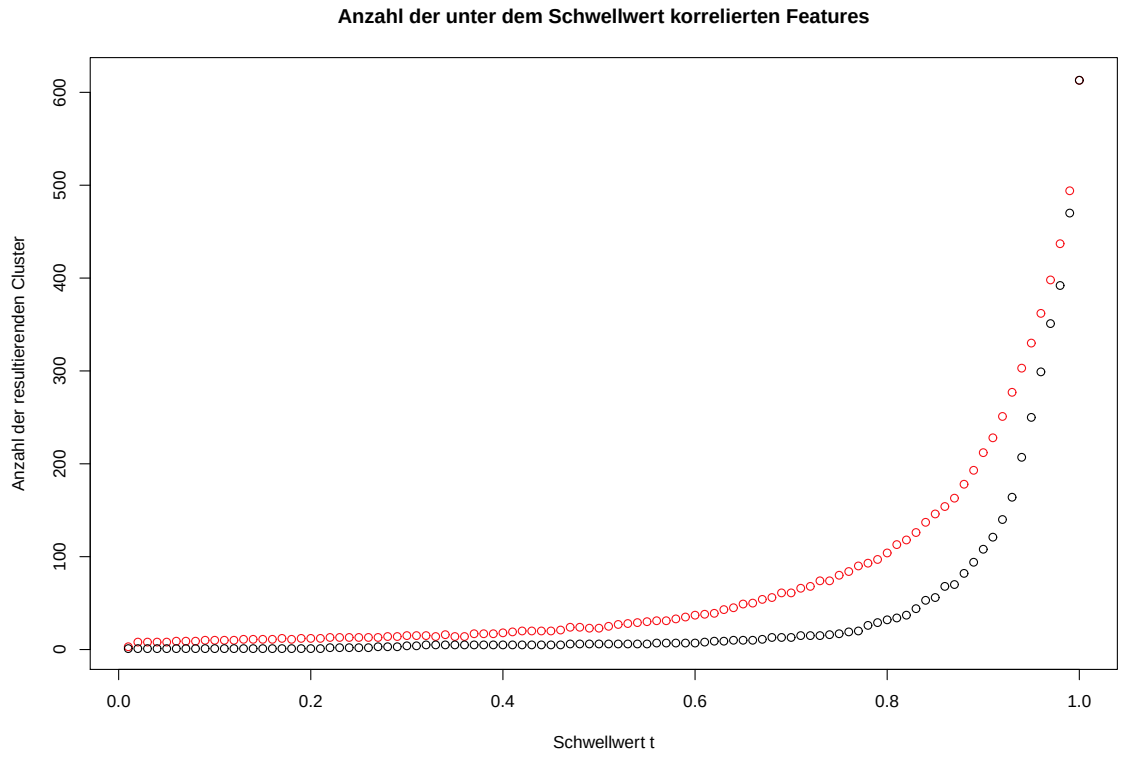

Abbildung 3.3.: Grafische Veranschaulichung der Abhängigkeit der resultierenden Clusteranzahl vom gewählten Schwellwert für den Algorithmus mit den Zusammenhangskomponenten (schwarz) sowie der Heuristik aus Algorithmus 2 (rot)

der Gruppe PAETG, das bei sehr vielen Scatterplots inmitten der WAR-Gruppe liegt. Genauso fällt das Feature 43 der Gruppe PAETG auf: Es liegt – wie die Abbildung 3.5 erkennen lässt – meist weit entfernt von den anderen Objekten der Gruppe PAETG und nahe der Gruppe CAS. Die beiden Solo-Klassen MIX und ACT (in gelb bzw. grau eingezeichnet) sind bei allen Scatterplots gut von den anderen Klassen unterscheidbar. Auch die Blank-Gruppe (in schwarz eingezeichnet) ist in allen Scatterplots eindeutig identifizierbar.

Führt man im nächsten Schritt eine Modellvorhersage und Klassifizierung für den SN10-Datensatz mit 613 Features basierend auf den *fitted values* – wie in Abschnitt 3.2.2 beschrieben – durch, so kann man sich, abhängig von der verwendeten Anzahl von PLS-Hauptkomponenten, den %-CC-Wert oder die Fehlklassifikationsrate ausgeben lassen. Eine Klassifizierung des SN10-Datensatzes mit acht HK ordnet das Feature 50 der WAR-Gruppe zu. Bei Verwendung von allen acht PLS-Hauptkomponenten wird nur ein einziges Objekt falsch klassifiziert, nämlich dieses eine Objekt 50. Die Anzahl der falsch klassifizierten Objekte in Abhängigkeit der zur Berechnung des Modells verwendeten PLS-Komponenten ist in Tabelle 3.3 dargestellt.

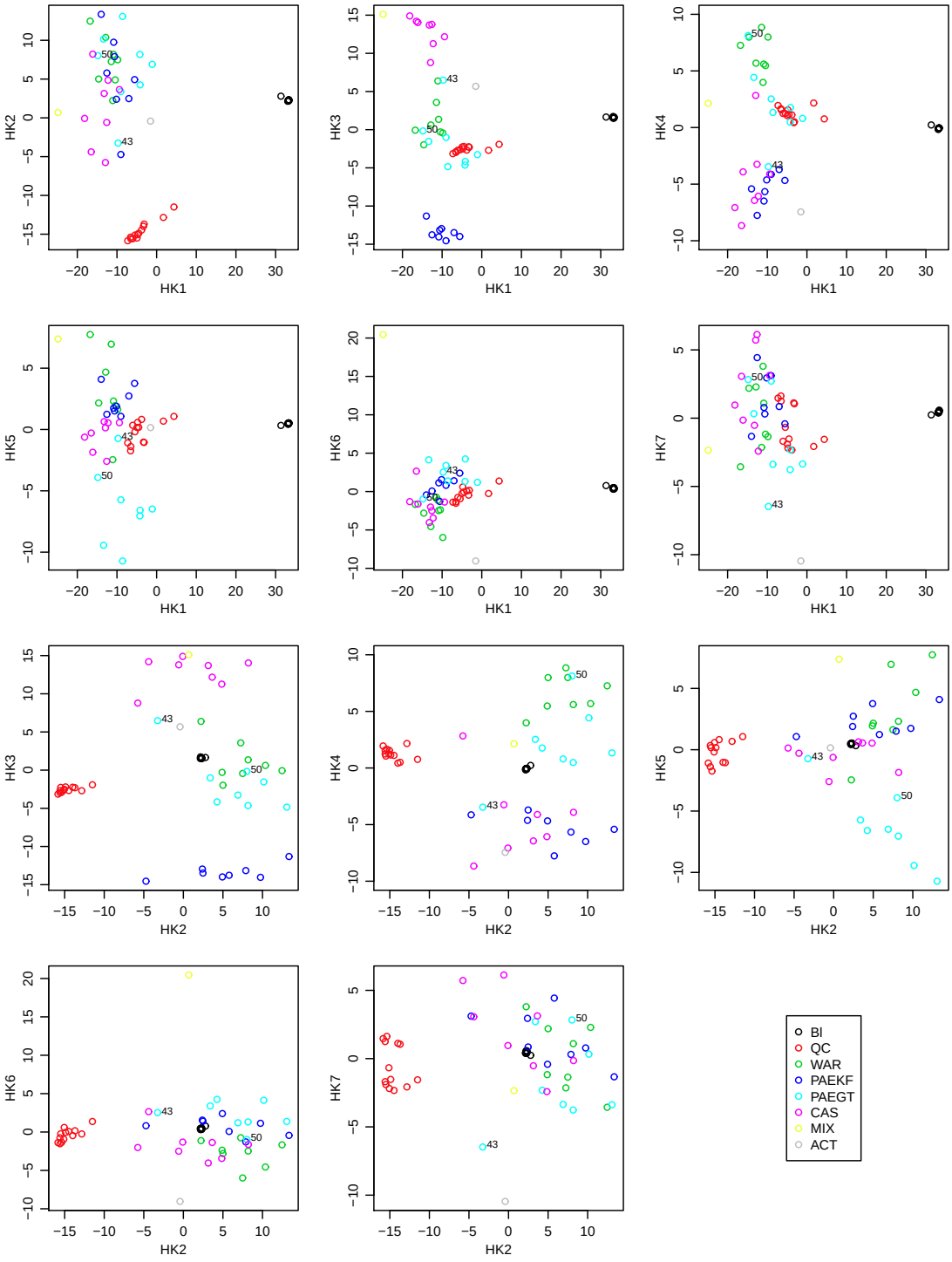

Abbildung 3.4.: Scatterplots der ersten acht Hauptkomponenten einer Partial-Least-Squares-Diskriminanzanalyse, Teil 1

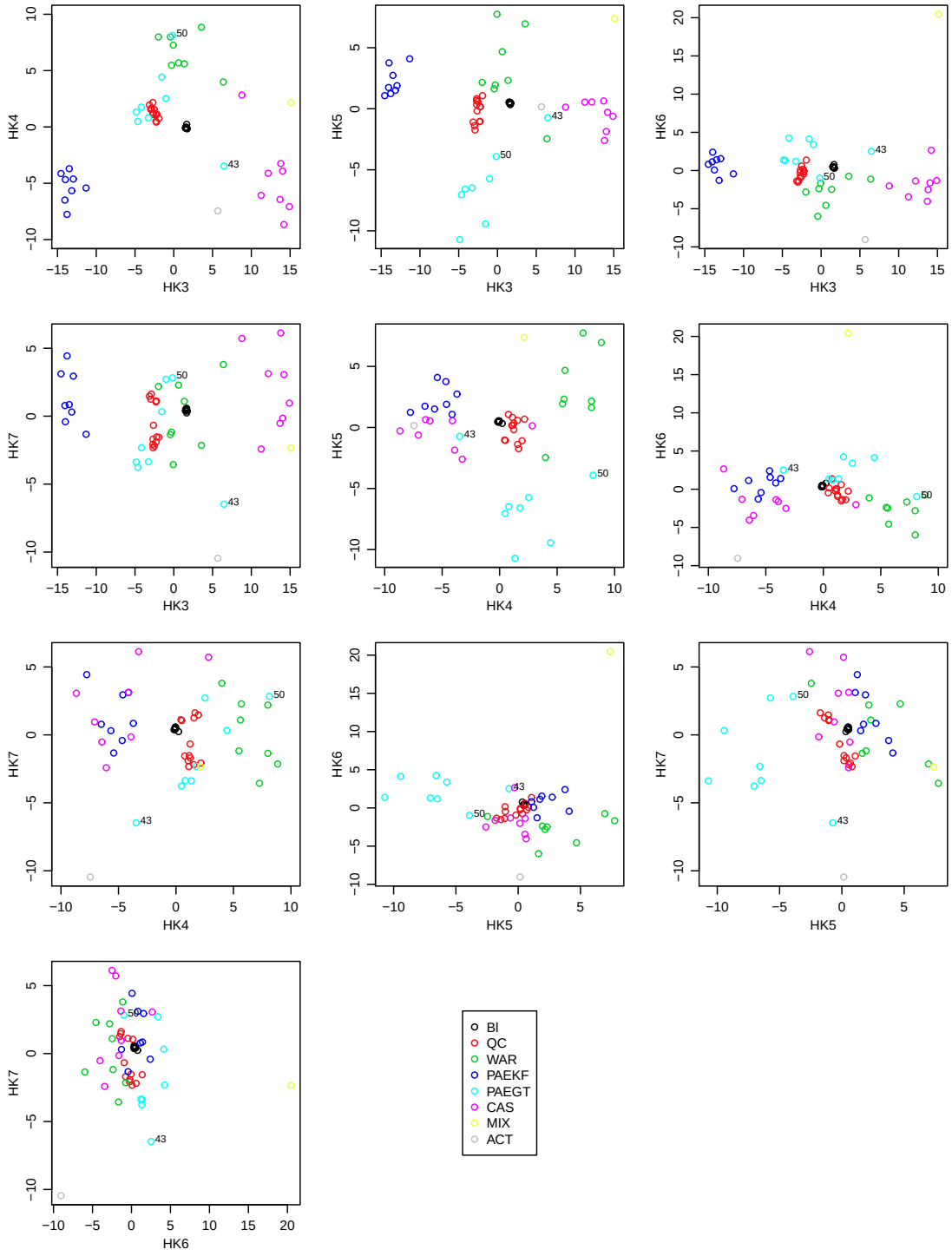

Abbildung 3.5.: Scatterplots der ersten acht Hauptkomponenten einer Partial-Least-Squares-Diskriminanzanalyse, Teil 2

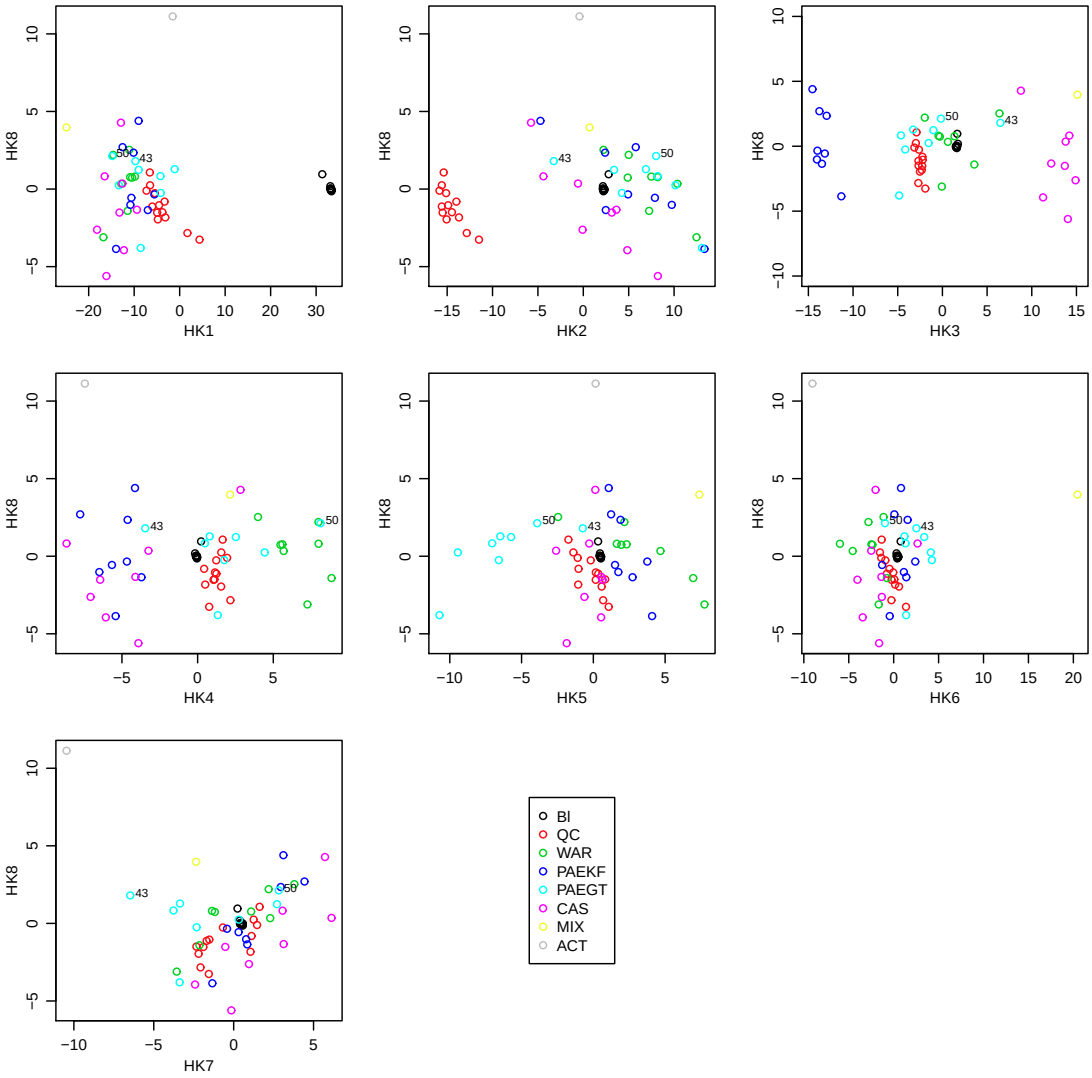

Abbildung 3.6.: Scatterplots der ersten acht Hauptkomponenten einer Partial-Least-Squares-Diskriminanzanalyse, Teil 3

# PLS-HK	# falsch klassifiziert	% falsch klassifiziert	%-CC
3	13	21.67 %	78.33 %
4	11	18.33 %	81.67 %
5	5	8.33 %	91.67 %
6	4	6.67 %	93.33 %
7	2	3.33 %	96.67 %
8	1	1.67 %	98.33 %

Tabelle 3.3.: Vergleich der Fehlerraten bei der Klassenzuordnung einer PLS-DA in Abhängigkeit von der Anzahl der verwendeten Hauptkomponenten

In Tabelle 3.3 ist erkennbar, dass die Fehlklassifikationsrate sinkt, je mehr HK im Modell verwendet werden. Bei einem Datensatz mit acht Klassen ist die Verwendung von zumindest sieben HK sinnvoll. Bei Verwendung von acht HK wird nur das Objekt 50 der Gruppe 5 (PAETG) fälschlicherweise der Klasse WAR (Gruppe 3) zugeordnet. Werden sieben HK für die Klassifikation herangezogen, so werden die Objekte 50 und 34 der falschen Klasse zugeordnet, wobei das 34. Objekt der Gruppe 6 (CAS) zugeordnet wird. Bei Verwendung von sechs HK werden schon vier Objekte – die Objekte 34, 43, 58 und 60 – der falschen Klasse zugeteilt. Eine genaue Aufschlüsselung der Klassifizierung mit PLS-DA ist in Tabelle 3.4 nachzulesen. In dieser Tabelle sind die dazugehörigen Klassifizierungen für verschiedene Anzahlen von im Modell verwendeten PLS-Hauptkomponenten dargestellt. Die falsch klassifizierten Objekte sind rot markiert und es wird jene Gruppe angegeben, in die die Objekte fälschlicherweise zugeordnet werden würden.

Bis jetzt wurde die PLS-DA nur für den großen SN10-Datensatz mit 613 Features durchgeführt. Dieselben Analysen können jedoch auch auf kleinere Datensätze angewandt und die Ergebnisse miteinander verglichen werden. In den weiteren Analysen wird häufig zwischen folgenden fünf Datensätzen unterschieden:

- dem großen SN10-Datensatz mit 613 Features, der mit Hilfe des Signal-Rausch-Verhältnisses größer 10 gefiltert wurde (im Folgenden kurz „613er“-Datensatz genannt),
- dem Datensatz mit 212 Features, der durch Anwendung des Algorithmus 2 auf den SN10-Datensatz entsteht (vgl. Abschnitt 3.3.1),
- dem SN200-Datensatz mit 243 Features, der durch ein Signal-Rausch-Verhältnis größer 200 feiner selektiert wurde,
- dem Datensatz mit 108 Features, der aus dem SN200-Datensatz nach Anwendung des Algorithmus 2 entsteht (vgl. Abschnitt 3.3.1) und
- dem „25er“-Datensatz, der aus 22 sicher identifizier- und zuordenbaren Features besteht.

Nummer	Name	Originalklasse	3 PLS-HK	4 PLS-HK	5 PLS-HK	6 PLS-HK	7 PLS-HK	8 PLS-HK
1	Bl01	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...
26	QC13	2	2	2	2	2	2	2
27	WAR5	3	6	3	3	3	3	3
28	WAR3	3	3	3	3	3	3	3
...	...	...	...	...	...	...	...	...
32	WAR2	3	3	3	3	3	3	3
33	WAR4	3	4	3	3	3	3	3
34	WAR8	3	6	3	5	5	6	3
35	PAEKF4	4	4	4	4	4	4	4
...	...	...	...	...	...	...	...	...
42	PAEKF5	4	4	4	4	4	4	4
43	PAETG2	5	6	6	6	6	5	5
44	PAETG7	5	4	3	5	5	5	5
45	PAETG3	5	4	4	5	5	5	5
46	PAETG4	5	4	4	5	5	5	5
47	PAETG5	5	4	3	5	5	5	5
48	PAETG6	5	4	3	5	5	5	5
49	PAETG8	5	4	3	5	5	5	5
50	PAETG1	5	3	3	5	5	3	3
51	CAS4	6	6	6	6	6	6	6
...	...	...	...	...	...	...	...	...
57	CAS2	6	6	6	6	6	6	6
58	CAS7	6	6	2	2	2	6	6
59	MIX	7	6	6	3	7	7	7
60	ACT	8	6	6	6	6	8	8

Tabelle 3.4.: Vergleich der Klassenzuordnung bei der PLS-DA bei Verwendung von unterschiedlich vielen Hauptkomponenten

In Tabelle 3.5 sind die falsch klassifizierten Objekte in Abhängigkeit vom verwendeten Datensatz sowie von der Anzahl der Hauptkomponenten angegeben. Für den großen Datensatz mit 613 Features und den reduzierten Datensatz mit 108 Features wurde auch eine PLS-DA durchgeführt, bei der die beiden Solo-Klassen MIX und ACT aus dem Modell genommen wurden. Es liegen dann 58 Objekte in sechs unterschiedlichen Klassen vor, weshalb eine Verwendung von zumindest fünf HK Sinn ergibt. Folgende Beobachtungen können aus Tabelle 3.5 gewonnen werden:

- Unabhängig vom verwendeten Datensatz sinkt die Anzahl der Fehlklassifikationen mit der Anzahl der im Modell verwendeten Hauptkomponenten.

- Wenn mehr Features für die Klassifikation verwendet werden, dann findet tendenziell eine bessere Zuordnung zu den unterschiedlichen Gruppen statt.

- Der aus den 243 Features gewonnene Datensatz mit 108 Features scheint die Gruppenklassifikation nicht wirklich zu verschlechtern. Es werden im Vergleich nicht mehr Objekte der falschen Klasse zugeordnet. Anscheinend ist es mit den Korrelationen tatsächlich gelungen, den Datensatz in der Dimension zu reduzieren, ohne die Klassifikation merklich zu verschlechtern.

- Werden beim „613er“- und beim „108er“-Datensatz die Klassifikationen mit und ohne Verwendung der zwei Solo-Klassen miteinander verglichen, so scheinen die beiden Solo-Klassen doch ihren Anteil zur korrekten Klasseneinteilung beizu-

tragen. Bei beiden Datensätzen steigt die Fehlklassifikationsrate, wenn die Vorhersage mit einem Modell ohne die Solo-Klassen getroffen wird.

- Es werden bei allen Datensätzen bei Verwendung von sieben oder acht HK maximal drei Objekte der falschen Klasse zugeordnet. Auffallend hierbei ist die extrem hohe Fehlklassifikation, die beim „25er"-Datensatz auftritt. Es macht den Anschein, dass die 22 sicher identifizierbaren Features deutlich weniger Informationsgehalt für die Klassifikation aufweisen.

Datensatz	# Objekte	# PLS-HK	# falsche Kl.	% falsche Kl.	falsche Objekte
108	60	8	2	3.3	34, 43
108	60	7	3	5.0	34, 43, 50
108	58	6	3	5.2	34, 43, 50
108	58	5	3	5.2	43, 50, 58
613	60	8	1	1.7	50
613	60	7	2	3.3	34, 50
613	58	6	3	5.2	34, 43, 58
613	58	5	3	5.2	34, 43, 58
243	60	8	3	5.0	34, 43, 50
243	60	7	3	5.0	34, 43, 50
25	60	8	6	10.0	25, 29, 34, 43, 50, 58
25	60	7	7	11.2	25, 29, 43, 48, 50, 58, 60

Tabelle 3.5.: Falsch klassifizierte Objekte in Abhängigkeit vom verwendeten Datensatz sowie von der Anzahl der betrachteten PLS-Hauptkomponenten

Bemerkenswert ist bei Tabelle 3.5, dass unabhängig vom verwendeten Datensatz immer wieder dieselben vier Objekte auffallen und falsch klassifiziert werden. Von der Fehlklassifikation sind häufig folgende Objekte betroffen, die im Folgenden auch mit *verdächtige Objekte* bezeichnet werden:

- Objekt 34 der Klasse 3 (WAR),
- Objekt 43 der Klasse 5 (PAETG),
- Objekt 50 der Klasse 5 (PAETG) und
- Objekt 58 der Klasse 6 (CAS).

Bei den falsch klassifizierten Objekten in Tabelle 3.5 fallen die Objekte 43 und 50, die beide der Gruppe PAETG angehören, oft als Ausreißer auf. Das 50. Objekt wird meist fälschlicherweise der 3. Gruppe WAR zugeordnet, während das 43. Objekt zu CAS (6. Gruppe) zugeordnet wird. Um nähere Informationen zu den Objekten und deren Beziehungen zueinander zu erhalten, wird die Korrelationsmatrix betrachtet. Im Speziellen werden die höchsten Korrelationen des 43. Objektes in Tabelle 3.6 angegeben.

Objektnummer	ID	Gruppe	Korrelation	Objektnummer	ID	Gruppe	Korrelation
51	2833	CAS	0.9706	53	2832	CAS	0.9696
32	2837	WAR	0.9682	28	2838	WAR	0.9681
31	2836	WAR	0.9652	52	2830	CAS	0.9639
56	2835	CAS	0.9559	34	2974	WAR	0.9474
44	2979	PAETG	0.9296	57	2831	CAS	0.9288
50	2840	PAETG	0.9274	55	2834	CAS	0.9258

Tabelle 3.6.: Größte Korrelationen des auffallenden 43. Objektes

Bei den Korrelationen des 43. Objektes in Tabelle 3.6 ist erkennbar, dass dieses PAETG-Objekt die höchste Korrelation zu einem CAS-Objekt der Gruppe 6 aufweist. Generell hat das 43. Objekt seine höchsten Korrelationen mit Objekten der Gruppen WAR und CAS. Das kann mitunter ein Grund sein, warum das 43. Objekt bei der PLS-DA fälschlicherweise der Gruppe CAS zugeordnet wurde. Bemerkenswert ist auch, dass das 43. Objekt hoch mit den anderen oft auffälligen Objekten 34 und 50 korreliert, nämlich $\mathrm{Kor}(43, 34) = 0.9474$ und $\mathrm{Kor}(43, 50) = 0.9274$.

3.3.3. Ergebnisse der Klassifikation mit Kreuzvalidierung

Für den Datensatz mit 60 Objekten ist die Vorgangsweise bei der Kreuzvalidierung folgendermaßen:

1. Wähle von den 60 Objekten per Zufallsauswahl 16 Objekte aus, die als Testdatensatz dienen. Dabei werden von der Blank- und QC-Gruppe jeweils vier Objekte und von den restlichen 8-er Gruppen jeweils zwei Objekte ausgewählt.
2. Die verbleibenden 44 Objekte (inkl. der zwei Solo-Klassen) stellen den Trainingsdatensatz dar.
3. Auf die Trainingsdaten wird eine PLS-DA mit acht HK sowie eine Vorhersage der Klassenzuordnung durchgeführt. Es wird der prozentuelle Anteil der falsch klassifizierten Objekte ausgegeben.
4. Das erhaltene Modell wird auf den Testdatensatz angewandt und damit eine Klasseneinteilung für die Testdaten vorhergesagt. Es wird wiederum der Fehlerprozentsatz der falsch klassifizierten Objekte des Testdatensatzes ermittelt.
5. Am Schluss wird sowohl die mittlere Anzahl der falsch klassifizierten Objekte bei den Trainingsdaten als auch bei den Testdaten ermittelt.

Diese Vorgangsweise wird insgesamt 2000 Mal wiederholt, um eine gute Schätzung für die durchschnittlichen Fehlerquoten der Klassifikation zu bekommen. Es werden für die Analyse verschiedene Größen der Testdatensätze betrachtet, nämlich 8 Testdaten,

16 Testdaten und 32 Testdaten. Die Verhältnisse Testdaten zu Trainingsdaten sind mit 8:52; 16:44 und 32:28 gegeben. Im Allgemeinen ist es für das Modell aussagekräftiger, die Testdatenmenge kleiner als die Trainingsdatenmenge zu wählen. Bei 32 Testdaten (also bei ca. gleich vielen Testdaten wie Trainingsdaten) ist mit keiner so guten Klassifikation mehr zu rechnen. Die zwei Solo-Klassen werden immer standardmäßig den Trainingsdaten zugeordnet. In Tabelle 3.7 werden die Fehler der Trainings- sowie Testdaten nach Anwendung der Kreuzvalidierung für unterschiedliche Datensätze miteinander verglichen. Zum einen wird der *große* Datensatz SN10 mit den 613 Variablen verwendet, zum anderen der *kleine* Datensatz SN200 mit den 243 Features. Weiters wird der mit Hilfe von Korrelationen reduzierte Datensatz aus Algorithmus 2 benutzt, wobei bei der Reduktion des großen Datensatzes 212 Features und bei der Reduktion des kleinen Datensatzes 108 Features übrig bleiben. Zuletzt wird der „25er"-Datensatz mit den 22 Substanzen, deren Struktur bekannt ist, verwendet.

Unabhängig vom Verhältnis zwischen Testdaten und Trainingsdaten werden bei den Testdaten meistens die verdächtigen Objekte $34, 43, 50$ oder 58 falsch klassifiziert, sofern sie den Testdaten angehören.

# Testdaten	Datensatz	% Testf.	% Trainingsf.	# Testf.	# Trainf.
8	613	5.49	0.57	0.44	0.30
8	243	4.49	3.72	0.36	1.93
8	212	5.18	1.14	0.41	0.59
8	108	3.91	2.91	0.31	1.51
8	25	17.71	8.17	1.42	4.25
16	613	6.25	0.30	1.00	0.13
16	243	5.98	2.13	0.96	0.93
16	212	6.13	0.75	0.98	0.33
16	108	5.62	1.75	0.89	0.77
16	25	19.50	6.33	3.12	2.78
32	613	6.08	0.05	1.94	0.01
32	243	6.29	0.45	2.01	0.13
32	212	5.87	0.11	1.88	0.03
32	108	6.49	0.43	2.07	0.12
32	25	22.98	3.12	7.35	0.87

Tabelle 3.7.: Vergleich der Fehlerraten bei der Kreuzvalidierung bei 2000 Wiederholungen für verschiedene Datensätze

In den letzten beiden Spalten von Tabelle 3.7 ist die mittlere Anzahl der falsch klassifizierten Features für den Testdatensatz sowie für den Trainingsdatensatz in Abhängigkeit des gewählten Verhältnisses angegeben. Hier liefert die Kreuzvalidierung mit den 16 Testdaten und den 44 Trainingsdaten für beide Datensätze im Mittel eine ungefähr gleich hohe Anzahl von falsch klassifizierten Objekten. Beim reduzierten Datensatz mit 108 Features wird im Mittel sowohl beim Test- als auch Trainingsdatensatz weniger als ein Feature falsch klassifiziert. Bei den 22 bekannten Substanzen ist die Rate

der Falschklassifikation schon wesentlich höher. Dort werden im Mittel drei Features einer Gruppe falsch zugeordnet. In Summe ist das gleiche Bild wie in Tabelle 3.5 erkennbar. Der Datensatz mit 108 Features weist beim Testdatensatz (bei Verwendung von 8 bzw. 16 Testdaten) die geringste mittlere Fehlklassifikation auf. Der Datensatz mit den 22 identifizierbaren Substanzen klassifiziert die Objekte deutlich schlechter und weist ca. die dreifachen Fehlklassifikationsraten auf. Zusammengefasst lässt sich mit Tabelle 3.7 feststellen:

- Bei der Kreuzvalidierung erweist sich bei den 60 Objekten eine Wahl von 8 bzw. 16 Testdaten als sinnvoll. Bei der Verwendung von mehr als der Hälfte der Objekte als Testdaten steigt die mittlere Anzahl von falsch klassifizierten Objekten bei den Testdaten um mehr als das Doppelte an.
- Der Datensatz mit 108 Features, der eine gute Reduktion der Features darstellt, scheint auch bei der Kreuzvalidierung eine zufriedenstellende Klassifikation zu liefern und weist im Mittel bei Test- und Trainingsdatensatz weniger als ein Objekt der falschen Klasse zu.

3.3.4. Ergebnisse der Klassifikation mit dem abgeschnittenen Datensatz

In Tabelle 3.5 sind häufig dieselben Objekte auffällig, die falsch klassifiziert werden. Meist sind die Objekte 34, 43, 50 und 58 von der Fehlklassifikation betroffen. In Tabelle 3.8 werden die Objekte des Datensatzes nochmals genau betrachtet. In der ersten Spalte sind die Objekte nach der Reihenfolge der Messung geordnet und in der zweiten Spalte findet sich die zugehörige Nummer des Objektes.

Objekt	Nr.	Objekt	Nr.	Objekt	Nr.	Objekt	Nr.
01_Bl	1	16_Bl	4	31_Bl	7	46_Bl	10
02_pool_QC	14	17_pool_QC	17	32_pool_QC	20	47_pool_QC	23
03_2971_WAR	27	18_2973_WAR	29	33_2847_PAEKF	40	48_2839_WAR	33
04_2845_PAEKF	35	19_2981_MIX	59	34_2848_PAEKF	41	49_2978_PAETG	48
05_2841_PAETG	43	20_2975_PAETG	45	35_2836_WAR	31	50_2831_CAS	57
06_Bl	2	21_Bl	5	36_Bl	8	51_Bl	11
07_pool_QC	15	22_pool_QC	18	37_pool_QC	21	52_pool_QC	24
08_2844_PAEKF	36	23_2830_CAS	52	38_2970_CAS	54	53_2969_CAS	58
09_2843_PAEKF	37	24_2850_ACT	60	39_2977_PAETG	47	54_2980_PAETG	49
10_2838_WAR	28	25_2972_WAR	30	40_2834_CAS	55	55_2840_PAETG	50
11_Bl	3	26_Bl	6	41_Bl	9	56_Bl	12
12_pool_QC	16	27_pool_QC	19	42_pool_QC	22	57_pool_QC	25
13_2833_CAS	51	28_2842_PAEKF	39	43_2846_PAEKF	42	58_2974_WAR	34
14_2849_PAEKF	38	29_2832_CAS	53	44_2835_CAS	56	59_Bl	13
15_2979_PAETG	44	30_2976_PAETG	46	45_2837_WAR	32	60_pool_QC	26

Tabelle 3.8.: Sortierung der Objekte gemäß der Reihenfolge der Messung

Bei Betrachtung der Objekte in der Reihenfolge der Messung kommt eine interessante Tatsache zum Vorschein: Offenbar spielt bei diesem Datensatz die Reihenfolge der

Messung eine wichtige Rolle. Die Objekte 34, 50 und 58 sind Objekte, die erst gegen Ende des Experiments gemessen wurden. Anscheinend wurde bei den letzten Durchläufen des Experiments nicht mehr zuverlässig gemessen. Solche Ungenauigkeiten in der Messung können aufgrund von Verunreinigungen des Massenspektrometers auftreten. Aus diesem Grund fallen die Objekte 34, 50 und 58 meist als Ausreißer auf. Das 43. Objekt hingegen kommt zu Beginn der Mess-Serie vor; die Reihenfolge kann beim 43. Objekt nicht als ausschlaggebender Grund für das Ausreißerverhalten angegeben werden.

Im Folgenden wird ein gestutzter Datensatz mit 52 Objekten betrachtet, bei dem der Datensatz ab der 53. Messung abgeschnitten wird. Dadurch werden die in der Reihenfolge der Messung hinteren Objekte $12, 13, 25, 26, 34, 49, 50$ und 58 vom Datensatz entfernt, um den Verfälschungen der Messfolge entgegenzuwirken. Dieser Datensatz wird in Folge mit *abgeschnitten* bezeichnet. In Tabelle 3.9 sind jeweils für den abgeschnittenen Datensatz, bei dem die zeitlich später gemessenen Objekte fehlen, die falsch klassifizierten Objekte in Abhängigkeit vom verwendeten Datensatz sowie von der Anzahl der Hauptkomponenten angegeben. Im Vergleich zum vollständigen Datensatz werden hier weniger Objekte der falschen Klasse zugeteilt. Über Tabelle 3.9 sind im Vergleich zu Tabelle 3.5 folgende Aussagen denkbar:

- Im Vergleich zum vollständigen Datensatz werden beim abgeschnittenen Datensatz deutlich weniger Objekte der falschen Klasse zugeordnet.
- Für die größeren Datensätze gibt es bei Verwendung von acht HK keine einzige Fehlklassifikation.
- Das bereits im vollständigen Datensatz auffallende Objekt 43 ist auch beim abgeschnittenen Datensatz auffällig, d. h. dass genauer zu hinterfragen ist, warum dieses Objekt Auffälligkeiten zeigt bzw. dass auch zu überlegen ist, dieses Objekt vom Datensatz zu entfernen.
- Zusätzlich zum Objekt 43 wird häufig das Objekt 60 der Solo-Klasse ACT einer falschen Klasse zugeordnet.
- Auch bei den abgeschnittenen Daten schafft der Datensatz mit den 22 bekannten Substanzen eine deutlich schlechtere Klassifikation der Objekte.

In Tabelle 3.10 sind die Fehlerraten bei Anwendung einer Kreuzvalidierung auf den abgeschnittenen Datensatz mit 2000 Wiederholungen angegeben. Hier ist wiederum erkennbar, dass die Fehlerraten der falsch klassifizierten Objekte im Vergleich zu Tabelle 3.7 unabhängig vom verwendeten Datensatz sinken. Die Ergebnisse aus Tabelle 3.10 lassen sich folgendermaßen zusammenfassen:

- Beim reduzierten Datensatz mit 108 Features wird beim Trainingsdatensatz kein einziges Mal ein Objekt falsch klassifiziert, während beim Testdatensatz pro Iteration maximal ein Objekt der falschen Klasse zugeordnet wird. In 85 % der Falschklassifikationen ist das 43. Objekt von der Fehlklassifikation betroffen.

Datensatz	# Objekte	# PLS-HK	# falsche Klasse	% falsche Klasse	falsche Objekte
108	52	8	0	0.0	-
108	52	7	1	1.9	43
108	52	6	2	3.8	43, 60
613	52	8	0	0.0	-
613	52	7	0	0.0	-
613	52	6	2	3.8	43, 60
243	52	8	0	0.0	-
243	52	7	0	0.0	-
243	52	6	2	3.8	43. 60
25	52	8	2	3.8	48, 52
25	52	7	2	3.8	48, 60
25	52	6	3	5.8	43, 48, 60

Tabelle 3.9.: Falsch klassifizierte Objekte in Abhängigkeit vom verwendeten Datensatz sowie von der Anzahl der betrachteten PLS-Hauptkomponenten für den abgeschnittenen Datensatz mit nur 52 Objekten

- Generell wird bei der Verwendung von acht HK und acht oder 16 Testdaten beim Testdatensatz im Mittel weniger als ein Objekt falsch klassifiziert – außer beim Datensatz mit den 22 bekannten Substanzen.
- Wird die Kreuzvalidierung beim „108er“-Datensatz mit 32 Testdaten durchgeführt, werden bei den Testdaten pro Durchlauf im Mittel ein Objekt und im Maximum drei Objekte falsch klassifiziert. Das 43. Objekt wird in 62 % der Durchläufe der falschen Klasse zugeordnet.
- Erst durch Eliminieren der Ausreißer in den Daten gelingt es, ein zufriedenstellendes Modell mit niedrigeren Fehlklassifikationsraten für die Gruppenvorhersage anzugeben.

Generell bleibt das 43. Objekt auch beim abgeschnittenen Datensatz auffallend. Da die zeitlich später gemessenen Objekte vom Datensatz entfernt wurden, sinkt die Fehlklassifikationsrate sowohl für den Test- als auch den Trainingsdatensatz im Vergleich zu Tabelle 3.7 enorm. Trotzdem ist das 43. Objekt in den meisten Durchläufen für die Fehlklassifikation verantwortlich.

Nach Rücksprache mit dem Institut für Bioanalytik und Metabolomics am Joanneum Research wurde bestätigt, dass Objekt 43 tatsächlich einen Ausreißer in den Daten darstellt. Bei diesem Objekt wurde ein Chargenwechsel von CAS auf PAETG durchgeführt, weshalb in dieser Probe noch ein großer Anteil an CAS vorhanden ist.

Aus diesem Grund wird nun das Objekt 43 für die restlichen Analysen von PAETG auf CAS umcodiert. Ab sofort gibt es daher nur noch sieben Objekte in der Gruppe

# Testdaten	Datensatz	# HK	% Testf.	% Trainingsf.	# Testf.	# Trainf.
8	613	8	2.14	0.00	0.17	0.00
8	243	8	1.89	0.00	0.15	0.00
8	108	8	2.08	0.00	0.17	0.00
8	25	8	15.31	3.38	1.22	1.49
16	613	8	2.19	0.00	0.35	0.00
16	243	8	2.25	0.00	0.36	0.00
16	108	8	2.12	0.00	0.34	0.00
16	108	7	1.95	0.33	0.31	0.12
16	25	8	17.47	2.38	2.80	0.86
32	613	8	3.30	0.00	1.06	0.00
32	243	8	3.30	0.00	1.06	0.00
32	108	8	3.32	0.00	1.06	0.00
32	25	8	21.58	0.52	6.90	0.10

Tabelle 3.10.: Vergleich der Fehlerraten des abgeschnittenen Datensatzes bei der Kreuzvalidierung bei 2000 Wiederholungen für verschiedene Datensätze

PAETG und neun Objekte in der Gruppe CAS. In Folge wird dieser Datensatz als *umcodiert* bezeichnet.

Wird nun auf den abgeschnittenen umcodierten Datensatz mit 108 Features eine PLS-DA angewandt, so werden sowohl bei der Verwendung von acht HK als auch bei sieben HK alle Objekte richtig klassifiziert. Bei sechs HK wird nur das einzige Element der Solo-Klasse ACT falsch zugeordnet. Um dieses gute Klassifikationsergebnis des abgeschnittenen umcodierten Datensatzes zu bestätigen, wird noch eine Kreuzvalidierung durchgeführt. Die dazugehörigen Fehlklassifkationsraten finden sich in Tabelle 3.11.

# Testdaten	Datensatz	# HK	% Testf.	% Trainingsf.	# Testf.	# Trainf.
8	108	8	4.45	0.00	0.36	0.00
8	108	7	4.31	0.00	0.35	0.00
8	25	8	13.60	2.00	1.09	0.88
8	25	7	10.40	4.28	0.83	1.88
16	108	8	4.46	0.00	0.71	0.00
16	108	7	4.40	0.00	0.70	0.00
16	25	8	14.76	1.63	2.36	0.59
16	25	7	11.83	3.51	1.89	1.26

Tabelle 3.11.: Fehlerraten bei der Kreuzvalidierung für den abgeschnittenen Datensatz, wobei das 43. Objekt richtig zu CAS codiert wird

Im Vergleich zu Tabelle 3.10 zeigt sich beim umcodierten Datensatz für die 22 bekannten Substanzen nochmals eine Verbesserung in der Klassifikation. Für den „25er“-

Datensatz sinkt sowohl beim Test- als auch beim Trainingsdatensatz die Fehlklassifikationsrate. Beim „108er“-Datensatz steigt hingegen die Fehlklassifikation bei den Testdaten um ungefähr das Doppelte an; es werden im Schnitt bei acht Testdaten 0.36 Objekte und bei 16 Testdaten 0.71 Objekte falsch klassifiziert. In Tabelle 3.11 wird beim Datensatz mit 108 Features sowohl bei sieben als auch bei acht HK kein einziges Objekt der Trainingsdaten falsch klassifiziert. Auch die mittlere Anzahl falsch klassifizierter Objekte liegt bei den Testdaten unter Eins. Beim Übergang von acht auf sieben HK macht sich eine Reduzierung des Testfehlers sowie beim „25er“-Datensatz eine Erhöhung des Trainingsfehlers erkennbar.

Insgesamt lässt sich sagen, dass der abgeschnittene Datensatz, bei dem das 43. Objekt zu CAS umcodiert wird, für den Datensatz mit den 22 bekannten Substanzen das beste Klassifikationsergebnis liefert. Beim abgeschnittenen Datensatz und Verwendung der 108 Features aus Algorithmus 2 hingegen steigt durch Umcodieren des 43. Objektes der Testfehler (im Vergleich zu Tabelle 3.10) wieder minimal. Dies lässt sich darauf zurückführen, dass durch das Umcodieren der Trainingsdatensatz anders trainiert wird und nun offenbar andere Objekte falsch klassifiziert werden, die vorher nicht auffällig waren.

3.3.5. Fehlklassifikationsraten der verdächtigen vier Objekte mit Kreuzvalidierung

Da die verdächtigen Objekte 34, 43, 50 und 58 oft – unabhängig von der Anzahl der verwendeten Features – der falschen Gruppe zugeordnet werden, stellt sich die Frage, welcher Gruppe sie fälschlicherweise zugeordnet werden und wie die Verteilung der Fehlklassifikation aussieht. Bei Durchführung von nur einer PLS-DA (ohne Kreuzvalidierung) werden das 34. Objekt und das 43. Objekt der Gruppe CAS zugeordnet, das 50. Objekt der Gruppe WAR sowie das 58. Objekt der Gruppe QC.

Um nun eine aussagekräftige Fehlklassifikationswahrscheinlichkeit dieser vier verdächtigen Objekte zu bekommen, wird eine Kreuzvalidierung mit 16 Testdaten, 44 Trainingsdaten sowie 8000 Wiederholungen durchgeführt. Es wird der Datensatz mit allen 60 Objekten verwendet, wobei das Objekt 43 zur Klasse CAS umcodiert wird. Bei jedem Durchlauf wird für jedes der vier verdächtigen Objekte folgende Information abgespeichert:

- Kommt das verdächtigte Objekt im Trainings- oder Testdatensatz vor?
- In welche Klasse wird das Objekt eingeordnet?

Insgesamt erhält man drei Fehlklassifikationswahrscheinlichkeiten dieser vier verdächtigen Objekte: eine für den gesamten Datensatz sowie jeweils eine für den Trainings- und Testdatensatz. Es lässt sich genau feststellen, wie häufig ein Objekt der jeweiligen Klasse zugeordnet werden würde. Die Ergebnisse der Fehlklassifikationsraten für die

vier verdächtigen Objekte sind in Tabelle 3.12 zusammengefasst. Die Klasse, der die Objekte tatsächlich angehören, ist gelb hinterlegt.

Objekt		gesamter Datensatz			Trainingsdatensatz			Testdatensatz		
34	Gruppe	3	5	6	3	6		3	5	6
	# Zuteilung	2667	56	5277	2564	3420		103	56	1857
	% Zuteilung	33.33	0.70	65.97	42.85	57.15		5.11	2.78	92.11
43	Gruppe	6			6			6		
	# Zuteilung	8000			6198			1802		
	% Zuteilung	100.00			100.00			100.00		
50	Gruppe	3	5		3	5		3	5	
	# Zuteilung	2543	5457		277	5456		2266	1	
	% Zuteilung	31.79	68.21		4.83	95.17		99.96	0.04	
58	Gruppe	2	3	6	2	3	6	3	6	
	# Zuteilung	8	1555	6437	8	1161	5052	394	1385	
	% Zuteilung	0.10	19.44	80.46	0.13	18.66	81.21	22.15	77.85	

Tabelle 3.12.: Fehlklassifikationsraten bei der Kreuzvalidierung bei 8000 Wiederholungen für die vier verdächtigen Objekte

In Tabelle 3.12 ist sofort ersichtlich, dass das 43. Objekt, das zur Gruppe 6 (CAS) umcodiert wurde, nun keine einzige Fehlklassifikation mehr aufweist.

Das Objekt 34, das eigentlich der 3. Gruppe WAR angehört, wird zu 66 % der Gruppe 6 (CAS) zugeordnet und nur zu ca. einem Drittel der tatsächlichen Gruppe 3. Auffallend ist, dass es sowohl in den Testdaten als auch in den Trainingsdaten mit einer Wahrscheinlichkeit von über 57 % der Gruppe 6 zugeordnet werden würde. Wenn es im Testdatensatz liegt, kommt es auch zu ca. 3 % zu einer Fehlklassifikation zur Gruppe 5 (PAETG). Bei den Trainingsdaten wird es in ca. 43 % der Fälle der richtigen 3. Gruppe zugeordnet werden, während dies bei den Testdaten nur zu 5 % der Fall ist.

Das 50. Objekt, das in Wirklichkeit der Gruppe 5 (PAETG) angehört, wird, wenn es in den Trainingsdaten liegt, zu über 95 % der richtigen Gruppe zugeordnet. Falls es im Testdatensatz vorkommt, wird es fast gänzlich falsch klassifiziert und der Gruppe 3 (WAR) zugeordnet. Beim Objekt 50 hängt die Fehlklassifikationsrate stark davon ab, ob das Objekt zu den Testdaten zählt oder nicht. Dies deutet darauf hin, dass das 50. Objekt auf die Klassenbildung großen Einfluss ausübt.

Das Objekt 58, das eigentlich der 6. Gruppe CAS angehört, wird auch zu ca. 80 % dieser Gruppe zugeordnet. Sowohl als Teil des Trainingsdatensatzes als auch des Testdatensatzes wird es zu ca. 80 % der Gruppe 6 (CAS) zugeteilt. Zu ca. 20 % wird es fälschlicherweise der Gruppe 3 (WAR) zugeordnet. Wenn es im Trainingsdatensatz liegt, kommt in ca. einem Promille der Fälle vor, dass es der Gruppe 2 (QC) zugeteilt wird. Da das 58. Objekt immer zu einem hohen Prozentsatz der richtigen Klasse zugeordnet wird, ist dieses Objekt bei der Klassenbildung sicher nicht so einflussreich wie das 50. oder 34. Objekt.

3.3.6. Analyse der Ladungen bei der PLS-DA

Zuerst werden für den Datensatz mit den 22 bekannten Features die Ladungen auf die ersten acht HK betrachtet, da die Ladungen in dieser Dimension noch tabellarisch veranschaulichbar sind. Im Anschluss werden auf die Ladungen des Datensatzes mit den 108 Features ähnliche und weiterführende Analysen angewandt.

In Tabelle 3.14 werden für den Datensatz mit den 22 bekannten Substanzen die betragsmäßig größten Ladungen auf die acht Hauptkomponenten angegeben. Die erste Spalte jeder HK beinhaltet die Featurenummer, von der die Ladung stammt. Die zweite Spalte jeder HK beinhaltet den betragsmäßigen Wert der Ladung und in der dritten Spalte ist angegeben, ob die Ladung positiv (1) oder negativ (0) zur HK beiträgt. In hellgrün ist für jedes der 22 Features die betragsmäßig maximale Ladung farbig markiert. Außerdem sind die maximalen Ladungen jeder HK eingefärbt (in blau, orange, dunkelgrün, rosa, gelb, grau und rot). Die Ladungen derselben Features in den anderen HK sind mit der gleichen Farbe markiert.

In Tabelle 3.13 sind für jede der acht HK aus Tabelle 3.14 der Mittelwert, die Standardabweichung und der Variationskoeffizient (CoV) der Ladungen angegeben.

	HK1	HK2	HK3	HK4	HK5	HK6	HK7	HK8
Mw	4.5064	3.4706	3.8054	3.2530	3.8055	4.3590	4.1851	4.1611
Stabw	0.0634	0.1474	0.1320	0.1591	0.1461	0.1658	0.1351	0.1659
CoV	0.0141	0.0425	0.0347	0.0489	0.0384	0.0380	0.0323	0.0399

Tabelle 3.13.: Mittelwert und Standardabweichung der Ladungen für jede der acht HK beim Datensatz mit den 22 identifizierbaren Features

Bei den Ladungen für den Datensatz mit den 22 bekannten Substanzen ist in den Tabellen 3.13 und 3.14 auffallend:

(a) Die erste HK hat betragsmäßig sehr kleine Ladungen, von denen zahlreiche Ladungen sehr nahe beisammen liegen. Insgesamt ist zwar gemäß Tabelle 3.13 die Summe der Ladungen auf die erste HK mit 4.5 am größten, sie weist jedoch von allen HK mit 0.06 die geringste Standardabweichung der Ladungen auf. Die Standardabweichung ist nicht einmal halb so groß wie die Standardabweichungen der anderen Komponenten. Auch der Variationskoeffizient ist mit 1.41 % im Vergleich zu den übrigen Komponenten, deren CoV-Werte zwischen 3 und 5 % liegen, sehr klein.

    Dies ist weiters ein Hinweis dafür, dass die erste HK keine so gute Trennung der Gruppen wie die anderen HK liefert, was bereits in den Abbildungen 3.4 und 3.5 erkennbar wurde. Generell können HK die unterschiedlichen Gruppen besser trennen, wenn sie unterschiedlich große Ladungen beinhalten und dadurch die Standardabweichung bzw. der Variationskoeffizient größer ist.

(b) Bei der zweiten HK springt sofort ins Auge, dass die Features der ersten fünf größten Ladungen auch bei dieser HK betragsmäßig ihre maximale Ladung aufweisen. Die zweite HK hat mit 3.47 den zweitkleinsten Mittelwert, mit 0.147 die drittgrößte Standardabweichung und mit 4.25 % den zweithöchsten CoV-Wert. Auffallend ist auch, dass die ersten fünf größten Ladungen über 0.3 liegen und somit größer als alle Ladungen der ersten HK sind. Diese Ergebnisse sind Faktoren dafür, dass die zweite HK eine gute Trennung der verschiedenen Gruppen garantiert.

(c) Die sechste HK besitzt, wie in Tabelle 3.13 angegeben, von allen HK mit 0.1658 die größte Standardabweichung. Der Mittelwert ist hinter dem der ersten HK am zweitgrößten. Die hohe Standardabweichung deutet auch darauf hin, dass die sechste HK eine gute Trennung der Gruppen gewährleistet, was sich auch schon in den Abbildungen 3.4 sowie 3.5 widerspiegelte.

(d) Die HK 4, 6, 7 und 8 besitzen ein Feature, das mit einer betragsmäßigen Ladung von über 0.6 extrem hoch auf die HK lädt. Die zweithöchsten Ladungen sind mit einem Wert unter 0.41 im Gegensatz dazu schon deutlich niedriger.

(e) Auf den ersten Blick fällt Feature Nummer 6 auf, weil es sowohl bei der sechsten als auch bei der achten HK an erster Position steht und bei beiden HK die betragsmäßig größte Ladung aufweist, wobei die maximale Ladung des sechsten Features in der achten HK mit 0.707 auftritt. Dieses sechste Feature (mit gelb markiert) hat im Vergleich auch bei den anderen HK überall eine relativ hohe Ladung.

(f) Die Bedeutung der Features für die HK kann angegeben werden, indem pro Feature die Ladungen auf die HK betrachtet werden. In Abschnitt 2.4.1.7 wurden verschiedene Kriterien besprochen, um die bedeutendsten Features für die HK sowie die Klassifikation zu bestimmen. Gemäß der ersten Methode werden die Ladungen aller Features betragsmäßig (ohne Wiederholung) geordnet. Die betragsmäßig größten Ladungen für jedes der 22 Features sind in Tabelle 3.14 in hellgrün eingefärbt. Es ist sofort ersichtlich, dass die zu den betragsmäßig größten Ladungen über 0.38 gehörenden Features folgendermaßen geordnet sind:

$$22 \geq 13 \geq 6 \geq 17 \geq 2 \geq 10 \geq 18 \geq 19 \geq 1 \geq 16 \geq 9 \geq 15.$$

Diese zwölf Features liefern sicherlich einen bedeutenden Betrag zu den HK und in Folge zum Klassifikationsmodell. Es lässt sich damit die Anzahl der relevanten Features von 22 auf 12 verkleinern. Die Features, deren betragsmäßig größten Ladungen im Vergleich am kleinsten sind und unter 0.3 liegen, lassen sich nach den Beträgen ihrer Ladungen in dieser Reihenfolge ordnen:

$$20 \leq 12 \leq 21 \leq 11 \leq 4.$$

Diese fünf Features mit einer maximalen betragsmäßigen Ladung von unter 0.3 können vom Datensatz entfernt werden, da sie kaum Einfluss auf die HK und in weiterer Folge auf die Klassifikation ausüben.

HK1 Feature	HK1 Ladung	HK1	HK2 Feature	HK2 Ladung	HK2	HK3 Feature	HK3 Ladung	HK3	HK4 Feature	HK4 Ladung	HK4
4	0.261	0	10	0.506	0	2	0.521	0	22	0.723	0
3	0.258	0	1	0.400	0	9	0.380	0	15	0.385	0
11	0.258	0	9	0.386	0	15	0.370	1	2	0.245	1
12	0.257	0	5	0.328	1	6	0.346	0	1	0.218	0
20	0.254	0	7	0.328	1	22	0.250	0	10	0.192	0
18	0.253	0	6	0.235	1	21	0.214	1	7	0.172	0
14	0.253	0	15	0.211	1	8	0.197	1	6	0.169	1
21	0.249	0	17	0.165	0	19	0.195	1	19	0.154	1
8	0.248	0	20	0.150	0	16	0.188	1	14	0.149	1
16	0.247	0	8	0.136	1	18	0.155	1	5	0.145	0
19	0.230	0	22	0.134	1	13	0.151	1	3	0.144	1
13	0.229	0	19	0.127	1	4	0.133	0	9	0.138	0
17	0.227	0	2	0.108	1	5	0.124	0	20	0.128	1
5	0.211	0	13	0.067	0	14	0.110	0	17	0.087	1
7	0.207	0	12	0.046	0	7	0.108	0	16	0.063	1
2	0.164	0	4	0.032	0	17	0.103	1	11	0.056	1
15	0.157	0	11	0.032	0	20	0.099	0	8	0.032	1
6	0.152	0	18	0.027	1	3	0.059	1	12	0.021	1
1	0.140	0	14	0.025	1	12	0.054	1	21	0.021	1
9	0.131	0	21	0.020	1	1	0.043	0	18	0.005	0
22	0.061	0	16	0.008	1	11	0.005	0	4	0.004	1
10	0.059	0	3	0.001	0	10	0.003	0	13	0.004	0

HK5 Feature	HK5 Ladung	HK5	HK6 Feature	HK6 Ladung	HK6	HK7 Feature	HK7 Ladung	HK7	HK8 Feature	HK8 Ladung	HK8
6	0.540	0	13	0.720	1	17	0.649	0	6	0.707	1
18	0.419	0	16	0.390	1	19	0.407	1	2	0.381	0
1	0.366	0	6	0.372	1	1	0.318	1	1	0.336	0
3	0.359	1	8	0.358	0	22	0.270	1	13	0.324	1
4	0.276	0	14	0.307	0	21	0.258	1	17	0.322	1
11	0.263	1	19	0.299	1	13	0.244	1	10	0.318	1
13	0.229	1	4	0.270	0	15	0.207	0	18	0.302	0
16	0.227	1	17	0.229	0	2	0.174	0	16	0.218	0
22	0.205	1	3	0.227	0	7	0.171	0	15	0.185	0
15	0.161	0	21	0.201	0	9	0.160	0	8	0.170	1
14	0.102	1	1	0.193	1	16	0.151	0	11	0.139	0
17	0.097	0	18	0.113	0	14	0.144	1	14	0.121	0
19	0.092	1	9	0.109	0	8	0.142	0	3	0.120	1
9	0.087	1	20	0.104	1	5	0.140	0	9	0.104	1
12	0.084	1	22	0.087	1	6	0.130	1	4	0.101	0
2	0.083	1	12	0.077	1	10	0.121	0	20	0.090	0
7	0.060	1	10	0.074	0	20	0.116	1	5	0.060	1
8	0.058	1	15	0.074	0	12	0.105	1	22	0.059	1
21	0.032	1	7	0.060	0	4	0.104	0	12	0.040	1
20	0.029	0	11	0.045	0	18	0.097	1	21	0.026	1
5	0.024	0	2	0.035	1	11	0.065	0	19	0.022	0
10	0.013	0	5	0.016	1	3	0.012	0	7	0.016	1

Tabelle 3.14.: Vergleich der betragsmäßig größten Ladungen auf die acht PLS-Hauptkomponenten beim Datensatz mit den 22 identifizierbaren Features: in hellgrün markiert: betragsmäßig maximale Ladung pro Feature andersfarbig markiert: pro HK Feature mit maximaler Ladung sowie dasselbe Feature bei den anderen HK vorkommend

(g) Es werden nun jene Features ausgewählt, deren betragsmäßiger Ladungswert über einem bestimmten Schwellwert liegt. Bei einem Schwellwert von 0.38 wurden in Unterpunkt (f) bereits zwölf Features selektiert. Für eine feinere Selektion gibt es noch weitere Möglichkeiten für die Wahl des Schwellwertes. Es gibt drei Features, deren betragsmäßig maximale Ladung über 0.7 liegt, nämlich die Features mit den Nummern $22, 13$ und 6, die auf die HK $4, 6$ und 8 am meisten laden. Möchte man alle Features auswählen, die mit einer maximalen Ladung von über 0.5 auf die HK wirken, so bleiben insgesamt folgende sechs Features – betragsmäßig geordnet nach ihren Ladungen – übrig:

$$22 \geq 13 \geq 6 \geq 17 \geq 2 \geq 10.$$

Je nach gewählten Schwellwert können mehr oder weniger Features aus dem Datensatz selektiert werden.

Dieselben Analysen der Ladungen können auch für den Datensatz mit 108 Features angewandt werden, nur lassen sich in diesem Fall die Ladungen aufgrund der hohen Dimension (Ladungsmatrix der Dimension 108×8) nicht mehr so anschaulich tabellarisch ordnen. Es wird jener Datensatz betrachtet, bei dem die hinteren acht Objekte abgeschnitten werden und das verdächtige 43. Objekt zur Gruppe CAS umcodiert wird.

Bei diesem Datensatz liegen nun elf Objekte in den Gruppen Blank und QC, sieben Objekte in der Gruppe WAR, acht Objekte in CAS (das 43. Objekt zählt nun anstatt zu PAETG zu CAS) und PAEKF sowie fünf Objekte in PAETG. In Summe sind 52 Objekte und 108 Features gegeben.

	HK1	HK1	HK1	HK2	HK2	HK2	HK3	HK3	HK3	HK4	HK4	HK4
Max	64	0.1323	0	82	0.2198	0	77	0.2689	0	6	0.2986	0
	51	0.1320	0	74	0.2084	0	91	0.2216	1	83	0.2916	0
	21	0.1312	0	204	0.2020	0	146	0.2206	1	221	0.2435	0
	...	...	...	...	...	...	...	...	...	...	...	...
Mw		0.0924			0.0773			0.0749			0.0716	
Stabw		0.0279			0.0598			0.0636			0.0661	
CoV		0.3019			0.7736			0.8491			0.9232	
	HK5	HK5	HK5	HK6	HK6	HK6	HK7	HK7	HK7	HK8	HK8	HK8
Max	217	0.4386	0	228	0.2430	1	187	0.4580	0	126	0.3931	0
	227	0.4039	0	106	0.2359	0	227	0.3534	0	144	0.3241	0
	126	0.2880	0	216	0.1998	0	69	0.3207	0	239	0.2545	1
	...	...	...	...	...	...	...	...	...	...	...	...
Mw		0.0691			0.0814			0.0667			0.0785	
Stabw		0.0694			0.0557			0.0713			0.0689	
CoV		1.0043			0.6843			1.0690			0.8777	

Tabelle 3.15.: Betragsmäßig größte Ladungen auf die acht PLS-Hauptkomponenten beim abgeschnittenen umcodierten Datensatz mit 108 Features

Die stärksten drei Ladungen der PLS-DA, angewandt auf den abgeschnittenen umcodierten Datensatz mit 108 Features, werden in Tabelle 3.15 angegeben. In dieser Tabelle ist auch für jede der ersten acht HK der Mittelwert, die Standardabweichung sowie der Variationskoeffizient der Ladungen angeführt.

In Tabelle 3.15 ist bei den Ladungen der ersten HK das gleiche Bild wie bei den Ladungen auf den Datensatz mit den 22 Features in Tabelle 3.14 erkennbar. Die erste HK weist eine deutlich geringere Standardabweichung von 0.03 auf; die höchsten Ladungen sind mit 0.13 nur halb so groß wie die maximalen Ladungen auf die anderen HK. Weiters ist zu bemerken, dass die Features auf die erste HK nur negativ laden. Die erste HK scheint auch bei diesem Datensatz die Gruppenaufteilung nicht so gut wiederzugeben. Die übrigen HK haben eine Standardabweichung zwischen 0.06 und 0.07. Auch der Variationskoeffizient ist bei der ersten HK mit 30 % deutlich kleiner als bei den restlichen HK, wo der CoV-Wert überall über 77 % liegt. Bei den HK 5 und 7 nimmt der Variationskoeffizient sogar einen Wert von über 100 % an. Das bedeutet, dass bei diesen beiden HK die Standardabweichung größer als der Mittelwert ist. Auffallend ist auch, dass die HK 1 bis 4 sowie die HK 6 nur kleinere Ladungen als 0.3 aufweisen, während bei den HK 5, 7 und 8 die maximalen Ladungen über 0.3 liegen.

3.3.7. Beitrag der Features zur Klassifikation

Um festzustellen, welche Features beim abgeschnittenen umcodierten Datensatz für die HK von Bedeutung sind, werden die betragsmäßig maximalen Ladungen von jedem Feature bestimmt und diese 108 Werte laut der Variablenselektion aus Abschnitt 2.4.1.7 absteigend geordnet. Jene 30 Features, die betragsmäßig am meisten auf die HK laden, sind in Tabelle 3.16 angeführt. Dabei wird angegeben, ob es sich um eine positive Ladung (1) oder eine negative Ladung (0) handelt und von welcher HK diese Ladung stammt.

Nr.	Ladung	pos./neg.	HK	Nr.	Ladung	pos./neg.	HK	Nr.	Ladung	pos./neg.	HK
187	0.4580	0	7	239	0.2545	1	8	82	0.2198	0	2
217	0.4386	0	5	221	0.2435	0	4	42	0.2178	0	4
227	0.4039	0	5	228	0.2430	1	6	92	0.2153	1	3
126	0.3931	0	8	106	0.2359	0	6	129	0.2143	1	3
144	0.3241	0	8	199	0.2310	0	4	235	0.2135	1	3
69	0.3207	0	7	19	0.2302	0	4	138	0.2104	1	3
6	0.2986	0	4	131	0.2248	0	8	230	0.2098	0	4
83	0.2916	0	4	24	0.2217	1	8	74	0.2084	0	2
77	0.2689	0	3	91	0.2216	1	3	243	0.2073	1	3
229	0.2603	0	7	146	0.2206	1	3	14	0.2059	0	5

Tabelle 3.16.: Liste jener 30 Features, die betragsmäßig am meisten auf die acht PLS-Hauptkomponenten beim abgeschnittenen umcodierten Datensatz laden

In Tabelle 3.16 ist erkennbar, dass die Features auf die HK des abgeschnittenen umcodierten Datensatzes generell nicht sehr hoch laden. Die höchste Ladung beträgt 0.4580

und wird vom Feature mit der Nummer 187 zur siebenten HK beigetragen. Es gibt insgesamt drei Features, deren maximale Ladung über 0.40 liegt, sechs Features, deren maximale Ladung über 0.30 liegt, und 36 Features, die mit einem maximalen Wert von über 0.20 zu den HK beitragen. Außerdem ist auffallend, dass bei den 30 Features, die am meisten auf die HK laden, kein Feature dabei ist, dessen Ladung von der ersten HK stammt. Das bedeutet, dass hier die erste HK zur Klassifikation keinen nennenswerten Beitrag liefert.

Mit dem maximalen Beitrag der Features auf die HK ergibt sich nun eine sehr interessante Fragestellung:

Auf wie viele Features lässt sich die Dimension des abgeschnittenen umcodierten Datensatzes ausgehend von den 108 Features noch weiter reduzieren, sodass das Ergebnis der Klassifikation nicht verschlechtert wird?

Zur Beantwortung dieser Frage werden die ersten k Features herangezogen, die auf die HK betragsmäßig am meisten laden. Mit diesen k Features werden neue Datensätze der Dimension $(52, k)$ generiert, wobei das 43. Objekt wiederum als zur Gruppe CAS gehörig codiert ist. Insgesamt werden 17 verschiedene Datensätze für $k \in [10, 100]$ betrachtet. Auf diese Datensätze wird eine PLS-DA angewandt und anschließend bestimmt, wie hoch die Fehlklassifikationsrate in Abhängigkeit von der verwendeten Anzahl von Features ist. Wünschenswert wäre natürlich, dass die Fehlklassifikationsrate sinkt, je mehr betragsmäßig hoch ladende Features im Modell verwendet werden. Es gilt eine Anzahl von Features zu bestimmen, auf die die Dimension des Datensatzes reduziert werden kann, ohne eine Verschlechterung der Klassifikation zu erreichen. In Abbildung 3.7 werden die Ergebnisse dieser Analyse grafisch dargestellt. In Abhängigkeit von der Anzahl der betragsmäßig hoch ladenden Features im Modell und der Anzahl der verwendeten HK wird angegeben, wie hoch die Fehlklassifikationsrate der PLS-DA ist.

In Abbildung 3.7 ist erkennbar, dass die Fehlklassifikationsrate, unabhängig davon, ob sieben oder acht HK im Modell verwendet werden, sinkt, je größer die Anzahl der verwendeten Features im Modell wird. Bei acht HK schafft man es bereits, dass bei Verwendung der 18 am meisten ladenden Features kein einziges Objekt falsch klassifiziert wird. Bei sieben HK ist dies möglich, wenn die Features mit den 24 größten Ladungen im Modell verwendet werden.

Mit dieser Vorgangsweise wurde eine weitere Methode gefunden, um die Dimension der Daten anhand der Ladungen zu reduzieren und dennoch keinen Verlust beim Ergebnis der Klassifikation hinzunehmen. Es ist dadurch möglich, die 108 Features auf nur noch 30 Features zu reduzieren, womit alle Objekte des Datensatzes richtig klassifiziert werden können.

Um diese Behauptung zu bestätigen, wird nun sowohl auf den Datensatz mit den 20 am meisten ladenden Features als auch auf den Datensatz mit den 30 am meisten ladenden Features eine Kreuzvalidierung mit 2000 Wiederholungen angewandt. Die Ergebnisse sind in Tabelle 3.17 zusammengefasst.

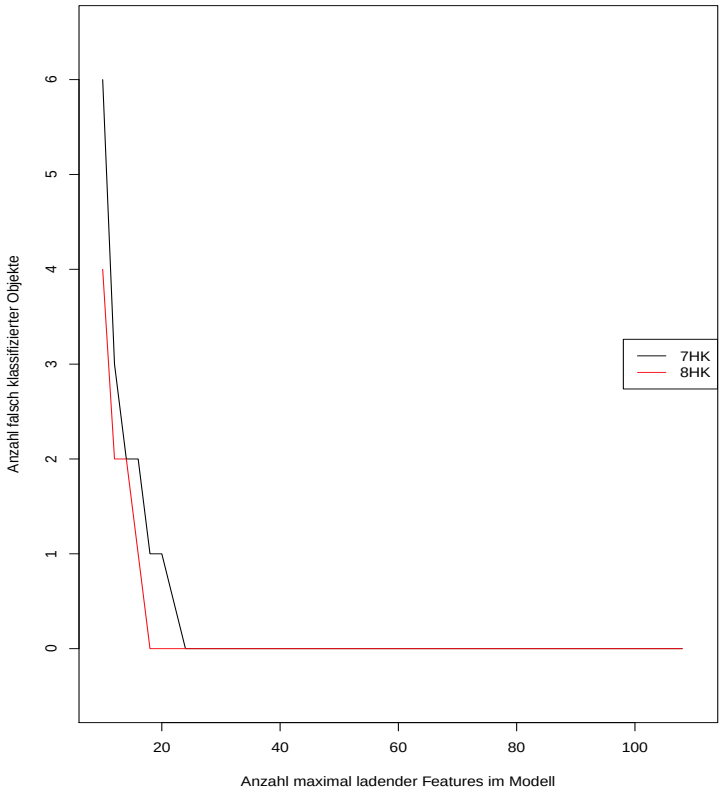

Abbildung 3.7.: Fehlklassifikationsrate in Abhängigkeit von der Anzahl der betragsmäßig hoch ladenden Features und von der Anzahl der verwendeten HK im Modell

In Tabelle 3.17 wird die Kreuzvalidierung für acht und für 16 Testdaten sowie mit acht und sieben HK durchgeführt. Es ist sofort ersichtlich, dass bei allen Varianten die mittlere Anzahl von falsch klassifizierten Objekten bei Test- und Trainingsdatensatz unter 1.2 liegt. Im Mittel wird bei Verwendung der 30 betragsmäßig am meisten ladenden Features beim Test- und Trainingsdatensatz nicht einmal ein Objekt falsch klassifiziert. Zusammenfassend lässt sich mit Tabelle 3.17 folgern:

- Werden bei 16 Testdaten anstelle von acht nur sieben HK im Modell verwendet, so steigt sowohl die Fehlklassifikationsrate beim Test- als auch beim Trainingsdatensatz.
- Werden für das Modell statt der 30 am meisten ladenden Features die 20 Features mit dem höchsten Beitrag zu den HK verwendet, so ergibt sich ein analoges Bild. Bei einem Übergang von den 30 auf die 20 am meisten ladenden Features im Modell steigt die Fehlklassifikationsrate. Im Speziellen ist beim Fehlerprozentsatz der Testdaten ein enormer Unterschied sichtbar. Bei Verwendung der 30 maximal ladenden Features im Modell werden im Mittel nur ca. halb so viele Features

# Testdaten	Datensatz	# HK	% Testf.	% Trainingsf.	# Testf.	# Trainf.
8	20	8	6.29	0.85	0.50	0.37
8	20	7	6.74	1.74	0.54	0.76
8	30	8	2.73	0.00	0.22	0.00
8	30	7	2.49	0.00	0.20	0.00
16	20	8	6.52	0.90	1.04	0.32
16	20	7	7.05	1.34	1.13	0.48
16	30	8	2.37	0.00	0.38	0.00
16	30	7	3.58	0.00	0.57	0.00

Tabelle 3.17.: Fehlerraten bei der Kreuzvalidierung der Datensätze mit den 20 bzw. 30 am meisten auf die HK ladenden Features für den abgeschnittenen umcodierten Datensatz

falsch klassifiziert wie bei der Analyse mit den 20 maximal ladenden Features (bei acht Testdaten nur ca. 0.2 Features statt 0.5 Features und bei 16 Testdaten unter 0.6 Features statt über 1.04 Features).

- Insgesamt lässt sich sagen, dass es durchaus sinnvoll ist, den Datensatz mit 108 Features noch weiter zu reduzieren, da bei Verwendung von den 30 zu den höchsten Ladungen gehörenden Features kaum Informationen für die Klassifikation verloren gehen. Im Vergleich zum „108er“-Datensatz in Tabelle 3.11 sinkt die Fehlklassifikation bei der Verwendung von 16 Testdaten und den 30 am meisten ladenden Features deutlich (um ca. die Hälfte). Bei 16 Testdaten wurden beim „108er“-Datensatz mit acht HK in den Testdaten im Mittel 0.71 Objekte falsch klassifiziert, bei Verwendung der 30 maximal ladenden Features liegt die Fehlklassifikation durchschnittlich bei 0.38 Objekten.

- Vergleicht man dieses Ergebnis mit der Fehlklassifikationsrate bei der Kreuzvalidierung des Datensatzes mit den 22 bekannten Substanzen in Tabelle 3.11, so liegt auch hier eine klar ersichtliche Verbesserung vor. Während bei der Verwendung von acht HK beim „25er“-Datensatz bei den Test- bzw. Trainingsdaten 15 % bzw. 3 % falsch klassifiziert wurden, erreicht man mit den 30 maximal ladenden Features eine Reduktion der Fehlklassifikation bei den Testdaten um ca. vier Fünftel auf 3 %, während bei den Trainingsdaten kein einziges Objekt falsch klassifiziert wird. Ein näherer Vergleich dieser beiden Datensätze wird in Abschnitt 3.3.9 gegeben.

Hiermit wurde eine statistische Methode gefunden, die Dimension der Daten von 108 auf 30 zu reduzieren, die einen deutlich höheren Prozentsatz an richtig klassifizierten Objekten liefert als die Klassifikation durch die 22 bekannten Features des „25er“-Datensatzes. Im Vergleich zu den größeren Datensätzen tritt kaum ein Informationsverlust ein; es wird im Schnitt sowohl bei den Test- als auch bei den Trainingsdaten weniger als ein Objekt der falschen Klasse zugeordnet.

3.3.8. Resultate bei Durchführung einer PCA

Am Ende des Abschnittes 2.4.1 wurden die Varianten *R-mode* und *Q-mode* der PCA eingeführt, abhängig davon, ob beim Datensatz mehr Variablen als Objekte vorliegen oder umgekehrt. Für beide Varianten gibt es auch in R unterschiedliche Anwendungsmöglichkeiten.

Die Funktion `princomp` aus dem Package *stats* kann eine *R-mode*-PCA durchführen, d. h. die Funktion `princomp` ist für Datensätze geeignet, bei denen weniger Variablen als Objekte vorhanden sind.

```
> princomp(X, cor = TRUE)
```

Der Funktion `princomp` wird die Datenmatrix X übergeben. Weiters wird mit TRUE und FALSE angegeben, ob für die Berechnung die Korrelationsmatrix verwendet werden soll. `princomp` verwendet die Eigenwerte und Eigenvektoren der Kovarianz- bzw. Korrelationsmatrix X. Liegen bei einem Datensatz (wie es meist bei den Metabolomics-Daten der Fall ist) mehr Variablen als Objekte vor, so muss für die Durchführung einer PCA auf die Funktion `prcomp` (aus demselben Package *stats*) zurückgegriffen werden, die die Berechnung einer *Q-mode*-PCA ermöglicht. Die Berechnung erfolgt bei `prcomp` mit einer Singulärwertzerlegung der (zentrierten und skalierten) Datenmatrix.

Die Klassifikation wurde in dieser Arbeit ausschließlich anhand der PLS-DA durchgeführt, da die Klassifikation in Abhängigkeit der Gruppenzugehörigkeit bei der PCA mehr Schritte benötigt und somit aufwändiger ist. Man müsste nach der PCA beispielsweise noch eine Diskriminanzanalyse anwenden, um eine Klassifizierung der Objekte bzgl. der Gruppenzugehörigkeit zu erreichen. Für die folgenden Analysen wird der Datensatz mit den 22 bekannten Substanzen betrachtet; d. h. in diesem Fall liegen weniger Variablen (Features) als Objekte vor und die PCA wird mit der Funktion `princomp` durchgeführt.

Um den Anteil der erklärten Variabilität pro Feature zu bestimmen, wird auf die Ergebnisse der PCA zurückgegriffen. In Anlehnung an Abschnitt 2.4.1.5 wird für jede Variable im Modell der erklärte Varianzanteil berechnet. Dies geschieht mit der im Package *chemometrics* implementierten Funktion `pcaVarexpl`. Der Funktion `pcaVarexpl` wird nur die Datenmatrix und die Anzahl der zu extrahierenden HK übergeben und liefert – für jede Variable in einem Barplot dargestellt – die erklärte Variabilität. Für den Datensatz mit den 22 bekannten Substanzen ist die erklärte Variabilität für jedes der 22 Features in Abhängigkeit von der Anzahl der verwendeten HK im Modell in Abbildung 3.8 dargestellt.

In Abbildung 3.8 ist – wie bereits in Abschnitt 2.4.1.5 beschrieben – erkennbar, dass der Anteil erklärter Variabilität pro Feature steigt, je mehr HK in das Modell eingehen. Bei Verwendung von nur einer HK ist ersichtlich, dass die Features 1, 2, 6, 9, 10, 15 und 22 einen sehr geringen Anteil von unter 50 % erklären. Werden weitere HK ins Modell aufgenommen, so steigt der Erklärungsgrad. Bei vier HK weisen alle Variablen

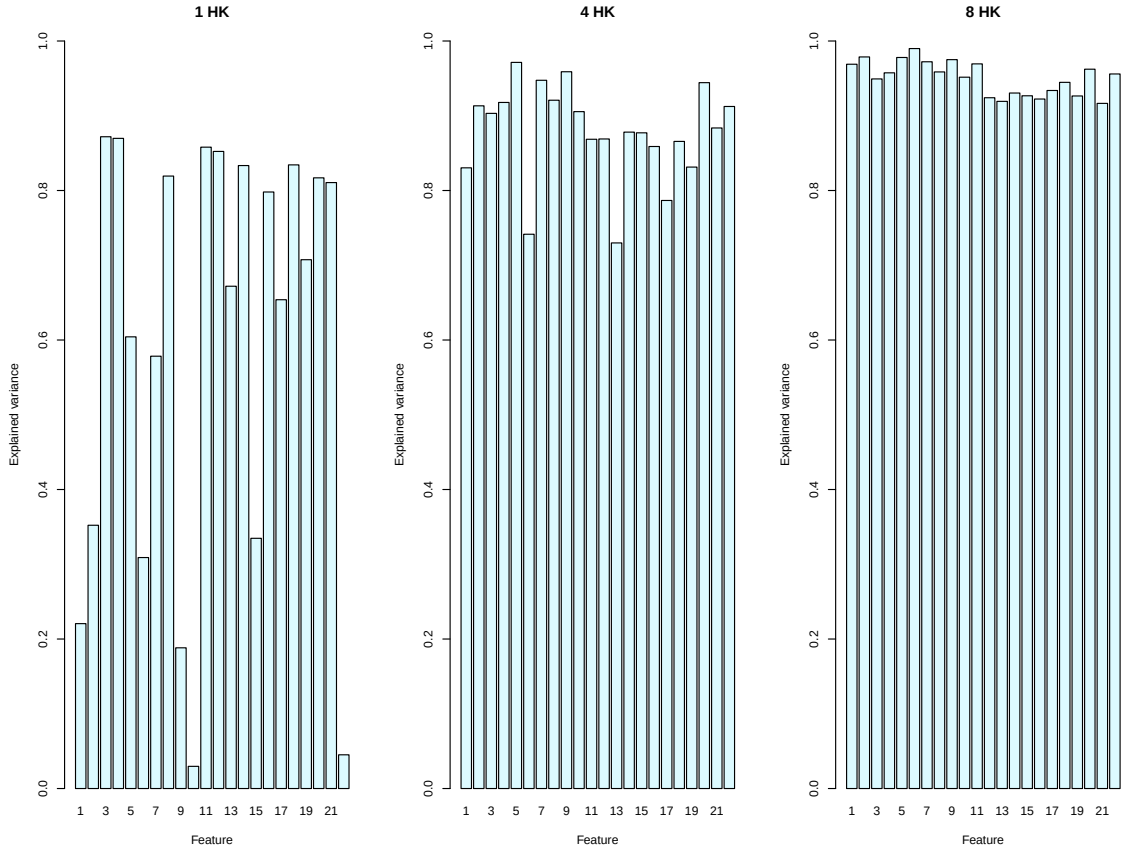

Abbildung 3.8.: Erklärte Variabilität pro Feature für den Datensatz mit den 22 bekannten Substanzen

einen Erklärungsgrad von über 70 % auf, wobei die Variablen 6 und 13 auffallen, da sie im Vergleich zu den anderen Variablen etwas weniger an Varianz erklären. Bei der Verwendung von acht HK im Modell ist zwischen den einzelnen Variablen kaum noch ein Unterschied im Erklärungsgrad erkennbar. Es liegen alle Werte über 90 %. Um also niedrigen Erklärungsgraden an Variabilität entgegenzuwirken, ist es zielführend, weitere HK in das Modell aufzunehmen.

Bei der PCA gibt es auch die Möglichkeit, die Objekte nach Ausreißern mit einer hohen Score- bzw. orthogonalen Distanz zu untersuchen. Im Folgenden wird dies für den Datensatz mit den 22 bekannten Substanzen durchgeführt, bei dem noch alle 60 Objekte im Modell vorkommen. Es wird auf den Datensatz eine PCA mit acht HK durchgeführt. Die Analyse erfolgt in Anlehnung an Abschnitt 2.4.1.6 unter Verwendung der Funktion `pcaDiagplot` des Packages *chemometrics*. Der kritische Wert für die Score-Distanz beträgt

$$\sqrt{\chi^2_{8,0.975}} = \sqrt{17.53} \approx 4.19.$$

Der Cutoff-Wert für die orthogonale Distanz ergibt sich durch

$$\left(\text{med}(\text{OD}^{\frac{2}{3}}) + \text{MAD}(\text{OD}^{\frac{2}{3}}) \cdot z_{0.975}\right)^{\frac{3}{2}} = (239886.7 + 95344.61 \cdot 1.96)^{\frac{3}{2}} = 2.7879 \cdot 10^8,$$

wobei $z_{0.975}$ das 97.5-%-Quantil der Standardnormalverteilung bezeichnet und mit MAD im Falle von Standardnormalverteilung der konsistente Schätzer der Standardabweichung gemeint ist. Für den Datensatz mit den 22 bekannten Substanzen lassen sich die Score-Distanzen bzw. orthogonalen Distanzen in Abbildung 3.9 ablesen.

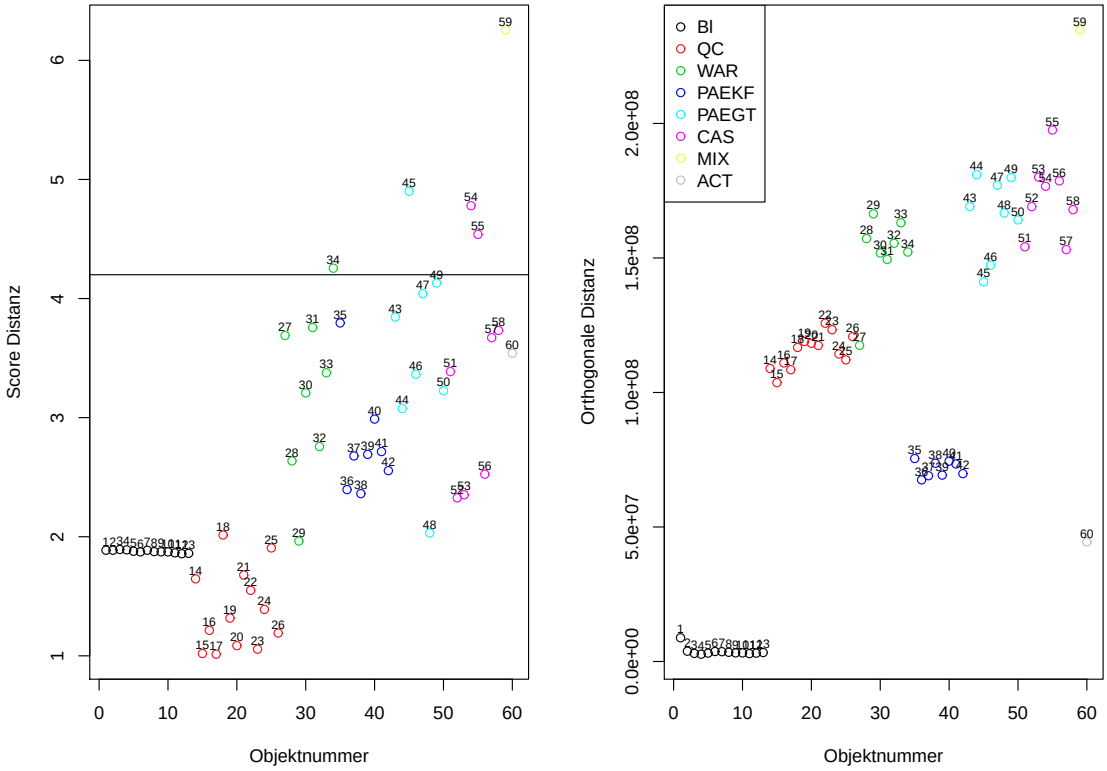

Abbildung 3.9.: Score-Distanzen und orthogonale Distanzen der Objekte für den Datensatz mit den 22 bekannten Substanzen

In Abbildung 3.9 sind nur bei den Score-Distanzen Ausreißer erkennbar, bei den orthogonalen Distanzen liegen dagegen alle Werte unter der kritischen Grenze von $2.7879 \cdot 10^8$. Es sind fünf Objekte auffällig, die eine Score-Distanz über dem kritischen Wert von 4.19 aufweisen: Bei den Objekten 34, 45, 54, 55 und 59 weist die Projektion in den PCA-Raum eine große Distanz zum Zentrum auf. Von diesen Objekten war bei der PLS-DA und Kreuzvalidierung in Abschnitt 3.3.3 das Objekt 34 immer wieder auffällig. Die anderen Objekte zeigten bei der PLS-DA keine Auffälligkeiten und wurden richtig klassifiziert.

Der große Unterschied besteht darin, dass mit der Score-Distanz bzw. orthogonalen Distanz Ausreißer in den Daten in Bezug auf den gesamten Datensatz gefunden werden, bei der PLS-DA hingegen können auch Ausreißer innerhalb der unterschiedlichen Gruppen festgestellt werden. Für die Frage der Klassifizierung ist in diesem Fall die PLS-DA zielführender.

3.3.9. Vergleich des Datensatzes der 22 bekannten Features mit dem Datensatz der 30 selektierten Features

In Tabelle 3.18 wird der „25er"-Datensatz der 22 identifizierbaren Features dem Datensatz der 30 in Abschnitt 3.3.7 (Tabelle 3.16) selektierten Features gegenübergestellt.

22 bekannte Features			30 selektierte Features		
Anzahl	Featurenummer	ID	Anzahl	Featurenummer	ID
1	5	271979	1	187	273359
2	6	271984	2	217	273525
3	13	271993	3	227	273687
4	21	272003	4	126	272390
5	35	272029	5	144	272735
6	42	272045	6	69	272100
7	46	272060	7	6	271984
8	59	272084	8	83	272136
9	66	272094	9	77	272128
10	74	272106	10	229	273703
11	100	272162	11	239	273807
12	103	272205	12	221	273562
13	105	272263	13	228	273698
14	108	272267	14	106	272265
15	111	272280	15	199	273406
16	118	272339	16	19	272001
17	157	273243	17	131	272448
18	159	273248	18	24	272006
19	172	273295	19	91	272147
20	178	273319	20	146	272776
21	182	273333	21	82	272135
22	217	273525	22	42	272045
			23	92	272148
			24	129	272412
			25	235	273787
			26	138	272628
			27	230	273718
			28	74	272106
			29	243	273836
			30	14	271995

Tabelle 3.18.: Gegenüberstellung vom Datensatz der 22 identifizierbaren Features und dem Datensatz der 30 selektierten Features

In Tabelle 3.18 ist ersichtlich, dass von den 30 in Abschnitt 3.3.7 selektierten Features nur vier mit den 22 identifizierbaren Features übereinstimmen. Davon befinden sich zwei Features – nämlich die Features mit den Nummern 217 und 6 – bereits unter den

20 am meisten ladenden Features. Das Feature 217 hat auf die HK der PLS-DA sogar den zweitgrößten Einfluss.

Die 30 am meisten ladenden Features liefern ein wesentlich besseres Klassifikationsergebnis als der Datensatz mit den 22 bekannten Features – wie in Tabelle 3.17 angegeben, ist die Fehlklassifikationsrate der Objekte weitaus niedriger. Die 22 bekannten Features beinhalten offenbar trotz der starken Dimensionsreduktion noch genügend Rauschen, das bei den 30 maximal ladenden Features durch für die Klassifikation relevante Information ersetzt wird.

Beim „25er“-Datensatz sind zwar aus chemischer Sicht alle Features sicher identifizier- und zuordenbar, jedoch lässt sich mit den 30 durch rein statistische Methoden der Dimensionsreduktion selektierten Features die Gruppenaufteilung der Objekte besser reproduzieren. Die Interpretation der 30 maximal ladenden Features hingegen gestaltet sich aus chemischer Sicht nicht so einfach. Es wird an der Bedeutung der Features geforscht; Ergebnisse werden vermutlich in wenigen Monaten vorliegen.

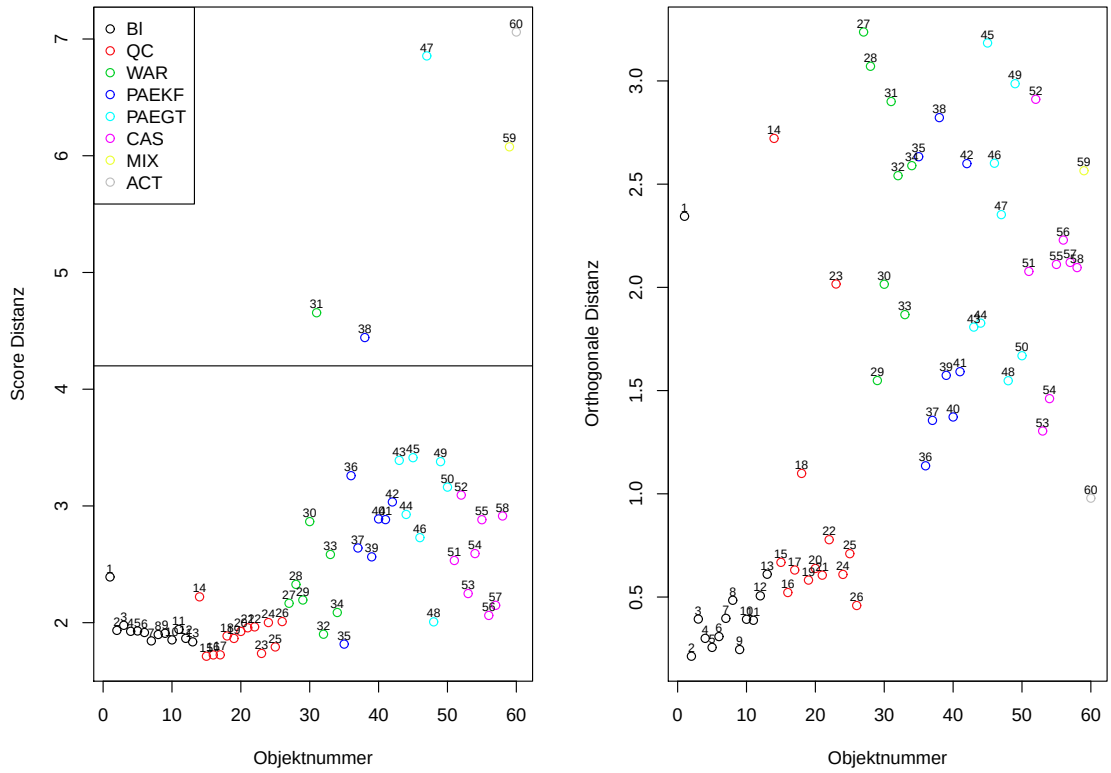

Abbildung 3.10.: Score-Distanzen und orthogonale Distanzen der Objekte für den Datensatz mit den 30 selektierten Features

Auch für die 30 selektierten Features wurden analog zu Abbildung 3.9 die Score- bzw. orthogonalen Distanzen bestimmt. Die Distanzen sind in Abbildung 3.10 dargestellt.

Wie schon bei den 22 identifizierbaren Features sind auch in Abbildung 3.10 nur in Hinblick auf die Score-Distanz Ausreißer erkennbar. Jeweils ein Objekt der Klassen WAR, PAEKF und PAETG weist eine hohe Score-Distanz auf. Außerdem fallen die beiden Objekte 59 und 60 der Solo-Klassen auf.

Insgesamt liefern die 30 maximal ladenden Features, die nur mit Hilfe rein statistischer Methoden gewonnen wurden, die geringste Fehlklassifikationsrate der Objekte. Somit gelingt mit weniger als 5 % der Features (Variablen) das beste Klassifikationsergebnis.

4. Resümee

4.1. Vor- und Nachteile der Methodik

Ein Nachteil der statistischen Analysemethoden bei Metabolomics-Datensätzen ist die starke Abhängigkeit der statistischen Verfahren von der im Vorfeld stattfindenden chemischen Voranalyse. Je mehr Hintergrundinformation der Statistiker über die Vorgangsweise der Erhebung und Messung der Daten bekommt, desto leichter fällt es ihm, den Datensatz sowie die Ergebnisse zu interpretieren.

Im Metabolomics-Journal wird die Wichtigkeit der Parameterwahl und der Voranalysen am Anfang des Versuchs folgendermaßen betont:

> „[...] Hiring a statistician after the data have been collected is like hiring a physician when the patient is in the morgue.“ (Goodacre, 2005)

Der Statistiker ist somit auf eine korrekte Voranalyse und Erhebung der Daten angewiesen, was sich unter Umständen als schwierig erweist und oft nicht gewährleistet ist. Wie in Abschnitt 3.3.4 erkennbar wurde, zählt bei dem in dieser Arbeit behandelten mehrklassigen Metabolomics-Datensatz die letzte Messreihe zu den Ausreißern. Durch die vielen Proben sind bei den letzten Durchläufen anscheinend Verschmutzungen im Massenspektrometer aufgetreten, die gerade durch die statistischen Analysen sichtbar wurden. Erst durch Abschneiden des Datensatzes und Weglassen der letzten acht Objekte liefert die PLS-DA mit der Kreuzvalidierung eine gute Vorhersage der Testdaten.

In dieser Arbeit wurden verschieden große Metabolomics-Datensätze behandelt. Je mehr Variablen vorliegen, desto schwieriger gestalten sich die Analysen sowie die Interpretationen der Ergebnisse. Die Modelle des großen Datensatzes mit 613 Variablen erweisen sich als sehr instabil; es ist somit keine zufriedenstellende Klassifikation der Daten möglich. Im Vergleich zu den reduzierten Datensätzen zeigt sich bei der Kreuzvalidierung eine deutlich höhere Fehlklassifikationsrate. Aus diesem Grund ist es immer zu empfehlen, vor Beginn der Analyse die Dimension in den Daten zu reduzieren. Die Dimensionsreduktion von multivariaten Daten hat zum Ziel, solche riesige Metabolomics-Datensätze zunächst leichter interpretierbar zu machen und in Folge Biomarker bzw. Ausreißer zu bestimmen.

In dieser Arbeit wurde Algorithmus 2 entwickelt, der die Dimension der Daten mit Hilfe der Korrelationen wesentlich reduziert. Durch Betrachtung der maximal ladenden Features bei einer PLS-DA gelingt noch eine weitere Reduktion der Dimension, sodass der %-CC-Wert der Klassifikation (im Vergleich zu den großen Datensätzen) nicht verschlechtert wird. Somit wurde eine Methode gefunden, die Dimension sogar mit deutlicher Senkung der Fehlklassifikationsrate von 613 Features auf nur mehr 30 Features zu reduzieren. Durch die starke Dimensionsreduktion wird der Datensatz überschaubar, wodurch die Ergebnisse leichter interpretierbar werden. Es gelingt, wenige Features zu bestimmen, die einen hohen Beitrag zur Klassifikation der Daten liefern.

Der Vorteil der PLS-DA liegt im Vergleich zur PCA darin, dass bei der PLS-DA in den Hauptkomponenten auch die Information der Matrix Y miteinbezogen wird. Bei der PLS wird die Kovarianz zwischen den Scores der Matrix X und der Matrix Y maximiert. Damit wird sowohl eine hohe Varianz von X sowie eine hohe Korrelation mit Y gewährleistet. Die PCA verwendet hingegen zur Bestimmung der Hauptkomponenten nur die Matrix X. Außerdem muss nach der PCA als zweiter Schritt beispielsweise eine Diskriminanzanalyse erfolgen, um eine Klassifikation in Abhängigkeit von den unterschiedlichen Gruppen zu erhalten. Bei der PLS-DA ist es möglich, durch die Analyse ohne einen weiteren Schritt die Klassifikation zu bekommen. Speziell für riesige Metabolomics-Datensätze, bei denen meistens weit mehr Variablen als Objekte vorliegen, erweist sich die PLS-DA als Klassifikationsmethode vorteilhaft.

Bei dem in dieser Arbeit behandelten Datensatz waren häufig die Objekte 34, 43, 50 und 58 als Ausreißer auffällig. Es ist bemerkenswert, dass alle Abweichungen statistisch erklärt werden konnten. Das Objekt 43 wird aufgrund eines Chargenwechsels fälschlicherweise einer anderen Gruppe zugeordnet. Die Auffälligkeiten der Objekte 34, 50 und 58 sind auf Messungenauigkeiten des Massenspektrometers zurückzuführen. Diese drei Objekte gehören zu den letzten acht Objekten des Datensatzes, welche nur mehr Rauschen darstellen. Erst durch Eliminieren dieser letzten acht Objekte vom Datensatz und Umcodieren des Objektes 43 lässt sich ein zufriedenstellendes Klassifikationsergebnis erreichen.

Mit rein statistischen Methoden wurde es in dieser Arbeit möglich, die ursprüngliche Dimension der Daten von 613 Features auf weniger als 5 % (30 Features) zu reduzieren. Die 30 selektierten Features beinhalten die größten Ladungen auf die HK der PLS-DA. Sie ermöglichen ein deutlich besseres Ergebnis in Hinblick auf die Fehlklassifikationsrate der Objekte als die 22 aus chemischer Sicht sicher identifizierbaren Objekte des „25er“-Datensatzes. Mit den 30 maximal ladenden Features lässt sich im Vergleich zu den 22 bekannten Features beim Testdatensatz eine Reduktion der Fehlklassifikationsrate um ca. vier Fünftel von 15 % auf 3 % erreichen. Ein Vergleich der beiden Datensätze wird in Abschnitt 3.3.9 gegeben. Die Tatsache, dass nur vier Objekte dieser beiden Datensätze identisch sind, deutet darauf hin, dass mit statistischen Verfahren für die Klassifikation wesentlich relevantere Features festgestellt werden können.

4.2. Offene Fragen und Ausblick

Die Möglichkeiten, eine optimale Dimensionsreduktion von Metabolomics-Datensätzen mit Hilfe von Korrelationen durchzuführen, wurden aufgezeigt, aber nicht weiter ausgeführt. Eine mögliche Weiterführung dieser Arbeit wäre also beispielsweise, die Heuristik in Algorithmus 1 in R zu implementieren und die unterschiedlichen Ergebnisse miteinander zu vergleichen.

Auch die Thematik, neue Gruppierungen mit Hilfe der Features zu finden, wurde nicht ausreichend behandelt. Weiterführende Analysen zum Entdecken neuer Gruppen können mit der Clusteranalyse abgehandelt werden. Hier könnte man einen Vergleich zwischen den Ergebnissen der in R vorimplementierten Clusteranalyse und den aus den Korrelationen der Features entstehenden Gruppen anstellen.

Eine sehr interessante Fragestellung, die sich während der Analyse der PLS-DA aufgetan hat, betrifft die Wahl der Zuordnungsvorschrift bei der Klassifikation in Abschnitt 3.2.2. Hier wurde als Klassifikationskriterium der maximale Wert der Wahrscheinlichkeit verwendet. Weiterführend könnten andere Vorschriften mit unterschiedlichen Wahrscheinlichkeiten für die Wahl der Zuordnungsvorschrift verwendet werden.

In Abschnitt 2.4.1.5 wurde ein Verfahren vorgestellt, das bei der PCA nicht nur die Variabilität der verschiedenen HK angibt, sondern auch die Variabilität pro Feature bestimmt. Dieser Ansatz ermöglicht auch eine Untersuchung der Variabilität pro Feature für eine PLS-DA.

Als Weiterführung dieser Arbeit könnten auch die Verteilungen der Fehlklassifikationen für die verschiedenen Datensätze bei der Kreuzvalidierung untersucht werden. Die Matrizen der Fehlklassifikation können in Abschnitt 3.3.3 bei der Kreuzvalidierung leicht bestimmt und wie untenstehend dargestellt werden.

	0_te	1_te	2_te	3_te	4_te	5_te	6_te	7_te	8_te	9_te
0_tr	0	2	15	30	49	31	14	0	1	0
1_tr	1	16	64	133	100	78	33	13	2	1
2_tr	2	49	128	190	175	92	50	17	8	1
3_tr	6	45	107	129	98	71	27	7	0	0
4_tr	1	16	42	44	26	25	15	1	0	0
5_tr	0	4	9	8	17	3	2	1	0	0
6_tr	0	0	0	0	1	0	0	0	0	0
7_tr	0	0	0	0	0	0	0	0	0	0
8_tr	0	0	0	0	0	0	0	0	0	0
9_tr	0	0	0	0	0	0	0	0	0	0

Die Verteilung von Test- und Trainingsfehlern könnte für unterschiedliche Größen des Trainingsdatensatzes mit Hilfe solcher Matrizen untersucht werden.

A. Heuristik zur Dimensionreduktion mit Hilfe der Korrelationen

Algorithmus 5 Heuristik zur Dimensionreduktion mit Hilfe der Korrelationen

```
1.  select_features<-function(datamatrix,treshold=0.9, m_features=0)
2.  if m_features==0 then
3.     max_features<-length(datamatrix[1,])
4.  else
5.     max_features<-m_features
6.  end if
7.  selected_features=c(rep(0,max_features))
8.  l<-1
9.  kor_matr<-c(rep(0,max_features))
10. kor_matr2<-c(rep(0,max_features))
11. while l<1+max_features do
12.    j<-1
13.    l_kor<-c()
14.    while j < 1+max_features do
15.       if (l!=j)&&(selected_features[l]==0)&&(selected_features[j]==0) then
16.          if abs(cor(datamatrix[,l],datamatrix[,j]))>treshold then
17.             kor_matr[l]<-kor_matr[l]+1
18.             l_kor<-c(l_kor,j)
19.          end if
20.          j<-j+1
21.       else
22.          j<-j+1
23.       end if
24.    end while
25.    j1<-1
26.    min_kor<-1
27.    while j1<length(l_kor) do
28.       j2<-j1+1
29.       while j2<=length(l_kor)) do
30.          if abs(cor(datamatrix[,l_kor[j1]],datamatrix[,l_kor[j2]]))<min_kor then
```

```
31.            min_kor<-abs(cor(datamatrix[,l_kor[j1]],datamatrix[,l_kor[j2]]))
32.          end if
33.          j2<-j2+1
34.        end while
35.        j1<-j1+1
36.      end while
37.      kor_matr2[l]<-min_kor
38.      l<-l+1
39.    end while

40.    features_todo<-max_features
41.    features_count<-0
42.    j<-1

43.    while j < 1+max_features do
44.      if kor_matr[j]==0 then
45.        features_todo<-features_todo-1
46.        selected_features[j]<-1
47.        features_count<-features_count+1
48.        kor_matr[j]<- -1
49.        kor_matr2[j]<- 2
50.      end if
51.      j<-j+1
52.    end while

53.    while features_todo>0 do
54.      min_feature<-which(kor_matr2==max(kor_matr2[which(kor_matr==
55.                                  min(kor_matr[which(selected_features==0)]))]))[1]
56.      j<-1
57.      while j<1+max_features do
58.        if (abs(cor(datamatrix[,min_feature],datamatrix[,j]))>treshold)&&
59.                                    (selected_features[j]==0)&&j!=min_feature then
60.          selected_features[j]<- -min_feature
61.          kor_matr[j]<- -1
62.          kor_matr2[j]<- 2
63.          features_todo<-features_todo-1
64.          i<-1
65.          while i<1+max_features do
66.            if (abs(cor(datamatrix[,j],datamatrix[,i]))>treshold)&&
67.                                                  (selected_features[i]==0) then
68.              kor_matr[i]<-kor_matr[i]-1
69.              k<-1
70.              l_kor<-c()
71.              while k < 1+max_features do
72.                if (i!=k)&&(selected_features[i]==0)&&
```

```
73.                                                    (selected_features[k]==0) then
74.                     if abs(cor(datamatrix[,i],datamatrix[,k]))>treshold then
75.                       l_kor<-c(l_kor,k)
76.                     end if
77.                   end if
78.                   k<-k+1
79.                 end while
80.                 j1<-1
81.                 min_kor<-1
82.                 while j1<length(l_kor) do
83.                   j2<-j1+1
84.                   while j2<=length(l_kor) do
85.                     if abs(cor(datamatrix[,l_kor[j1]],datamatrix[,l_kor[j2]]))
86.                                                                    <min_kor then
87.                       min_kor<-abs(cor(datamatrix[,l_kor[j1]],datamatrix[,l_kor[j2]]))
88.                     end if
89.                     j2<-j2+1
90.                   end while
91.                   j1<-j1+1
92.                 end while
93.                 kor_matr2[i]<-min_kor
94.               end if
95.               i<-i+1
96.             end while
97.           end if
98.           j<-j+1
99.         end while
100.        selected_features[min_feature]=1
101.        features_todo<-features_todo-1
102.        features_count<-features_count+1
103.        kor_matr[min_feature]<- -1
104.        kor_matr2[min_feature]<- 2
105.        print features_todo
106.    end while
107.    print „Anzahl selektierter Features“
108.    print features_count
109.    return list(selected_features,features_count)
```

B. Iterative Durchführung der Kreuzvalidierung in R

Algorithmus 6 Iterative Durchführung der Kreuzvalidierung

```
1. kreuzvalidierung_func<-function(datamatrix,repetition)
2. {
3. library(sampling)
4. neuematrix<-cbind(Daten_sort[,2],datamatrix)
5. names(neuematrix)[1]<-"Gruppe"
6. ohne_solo<-neuematrix[1:58,]

7. a_train<-rep(1,9)
8. b_train<-rep(2,9)
9. c_train<-rep(3,6)
10. d_train<-rep(4,6)
11. e_train<-rep(5,6)
12. f_train<-rep(6,6)
13. g<-7
14. h<-8
15. gruppen_num_train<-c(a_train,b_train,c_train,d_train,e_train,f_train,g,h)

16. a_test<-rep(1,4)
17. b_test<-rep(2,4)
18. c_test<-rep(3,2)
19. d_test<-rep(4,2)
20. e_test<-rep(5,2)
21. f_test<-rep(6,2)
22. gruppen_num_test<-c(a_test,b_test,c_test,d_test,e_test,f_test)

23. library(BayesTree)
24. x1tr <- as.factor(gruppen_num_train)
25. xx1tr <- as.data.frame(x1tr)
26. num_matrixtr <- makeind(xx1tr)

27. x1te <- as.factor(gruppen_num_test)
28. xx1te <- as.data.frame(x1te)
29. num_matrixte <- makeind(xx1te)
```

```
error_train<-c()
error_test<-c()
suspect_objects2<-c()
suspect_class<-c()
summeh<-0
matrix_special_test<-t(matrix(c(rep(0,17)),17))
colnames(matrix_special_test)<-c("","","","","","","","","","","","","","","","","","","")
merke<-c()
for i in 1:repetition do
  strata_data<-strata(ohne_solo,size=c(4,4,2,2,2,2),stratanames=c(„Gruppe"),
                                                  method=„srswor",description=FALSE)
  testdaten_sort<-sort(strata_data$ID_unit)
  print testdaten_sort
  all_daten<-c(1:60)
  trainingsdaten<-all_daten[-testdaten_sort]
  trainmatr<-datamatrix[trainingsdaten,]
  testmatr<-datamatrix[testdaten_sort,]
  mean_var<-apply(trainmatr,2,mean)
  sd_var<-apply(trainmatr,2,sd)
  stand_var<-((t(trainmatr)-mean_var)/(sd_var))
  trainmatr_stand<-t(stand_var)
  library(caret)
  cowda<-plsda(trainmatr_stand,num_matrixtr,ncomp=8)
  fitvalall_plsda<-cowda$fitted.values
  fitval_plsda<-fitvalall_plsda[1:44,1:8,8:8]
  max_fitval<-apply(fitval_plsda,1,max)
  min_fitval<-apply(fitval_plsda,1,min)
  norm1_fitval<-(fitval_plsda-min_fitval)/(max_fitval-min_fitval)
  sum_norm1<-apply(norm1_fitval,1,sum)
  norm_fitval<-norm1_fitval/sum_norm1
  klassenzuordnung<-max.col(norm_fitval)
  confmatrix<-confusionMatrix(klassenzuordnung,gruppen_num_train)
  error_train[i]<-1-confmatrix$overall[[1]]
  false_class_train<-trainingsdaten
                                  [-which(klassenzuordnung==gruppen_num_train)]
  print „Falsch klassifizierte Objekte der Trainingsdaten"
  print false_class_train
  mean_vart<-apply(testmatr,2,mean)
  sd_vart<-apply(testmatr,2,sd)
  stand_vart<-((t(testmatr)-mean_var)/(sd_var))
```

```
70.  testmatr_stand<-t(stand_vart)
71.  pred_test_long<-predict(cowda,newdata=testmatr_stand,type="class")
72.  pred_test<-c()
73.  lev1<-which(pred_test_long==„x1tr.1“)
74.  lev2<-which(pred_test_long==„x1tr.2“)
75.  lev3<-which(pred_test_long==„x1tr.3“)
76.  lev4<-which(pred_test_long==„x1tr.4“)
77.  lev5<-which(pred_test_long==„x1tr.5“)
78.  lev6<-which(pred_test_long==„x1tr.6“)
79.  lev7<-which(pred_test_long==„x1tr.7“)
80.  lev8<-which(pred_test_long==„x1tr.8“)
81.  pred_test[lev1]<-1
82.  pred_test[lev2]<-2
83.  pred_test[lev3]<-3
84.  pred_test[lev4]<-4
85.  pred_test[lev5]<-5
86.  pred_test[lev6]<-6
87.  pred_test[lev7]<-7
88.  pred_test[lev8]<-8
89.  pred_test2<-factor(pred_test,levels=1:8)
90.  gruppen_num_test2<-factor(gruppen_num_test,levels=1:8)
91.  confmatrix_test<-confusionMatrix(pred_test2,gruppen_num_test2)
92.  error_test[i]<-1-confmatrix_test$overall[[1]]
93.  false_class_test<-testdaten_sort[-which(pred_test2==gruppen_num_test2)]
94.  print „Falsch klassifizierte Objekte der Testdaten“
95.  print false_class_test
96.  suspect_objects2<-c(false_class_test,suspect_objects2)
97.  suspect_objects<-unique(suspect_objects2)
98.  suspect<-c(34,43,50,58)
99.  j<-1
100.  while j<5 do
101.    t1<-which(testdaten_sort==suspect[j])
102.    t2<-which(suspect[j]==false_class_test)
103.    if length(t1)==1 then
104.      if length(t2)==1 then
105.        suspect_class<-c(0,suspect_class)
106.      end if
107.      if length(t2)==0 then
108.        suspect_class<-c(1,suspect_class)
109.        merke<-c(suspect[j],merke)
110.      end if
111.    end if
```

```
112.       j<-j+1
113.     end while
114.     print „Objekt in Testdaten und trotzdem richtige Klasse“
115.     print suspect_class
116.     if sum(suspect_class)==summeh+1 then
117.       matrix_special_test<-rbind(matrix_special_test,c(merke[1],testdaten_sort))
118.       summeh<-sum(suspect_class)
119.     end if
120.   end for
121.   i<-i+1
122.   return list(error_train,error_test,suspect_objects2,suspect_objects,
123.                                       suspect_class,matrix_special_test[-1,],merke)
124.   }
```

C. Aufschlüsselung der im Datensatz verwendeten 60 Objekte

Objekt	Abkürzung	Objektnummer	Gruppennummer	Gruppenname
01_Bl	Bl01	1	1	Blank
06_Bl	Bl02	2	1	Blank
11_Bl	Bl03	3	1	Blank
16_Bl	Bl04	4	1	Blank
21_Bl	Bl05	5	1	Blank
26_Bl	Bl06	6	1	Blank
31_Bl	Bl07	7	1	Blank
36_Bl	Bl08	8	1	Blank
41_Bl	Bl09	9	1	Blank
46_Bl	Bl10	10	1	Blank
51_Bl	Bl11	11	1	Blank
56_Bl	Bl12	12	1	Blank
59_Bl	Bl13	13	1	Blank
02_pool_QC	QC01	14	2	QC
07_pool_QC	QC02	15	2	QC
12_pool_QC	QC03	16	2	QC
17_pool_QC	QC04	17	2	QC
22_pool_QC	QC05	18	2	QC
27_pool_QC	QC06	19	2	QC
32_pool_QC	QC07	20	2	QC
37_pool_QC	QC08	21	2	QC
42_pool_QC	QC09	22	2	QC
47_pool_QC	QC10	23	2	QC
52_pool_QC	QC11	24	2	QC
57_pool_QC	QC12	25	2	QC
60_pool_QC	QC13	26	2	QC

Tabelle C.1.: Liste der im Datensatz verwendeten 60 Objekte, Teil 1

Objekt	Abkürzung	Objektnummer	Gruppennummer	Gruppenname
03_2971_WAR	WAR5	27	3	WAR
10_2838_WAR	WAR3	28	3	WAR
18_2973_WAR	WAR7	29	3	WAR
25_2972_WAR	WAR6	30	3	WAR
35_2836_WAR	WAR1	31	3	WAR
45_2837_WAR	WAR2	32	3	WAR
48_2839_WAR	WAR4	33	3	WAR
58_2974_WAR	WAR8	34	3	WAR
04_2845_PAEKF	PAEKF4	35	4	PAEKF
08_2844_PAEKF	PAEKF3	36	4	PAEKF
09_2843_PAEKF	PAEKF4	37	4	PAEKF
14_2849_PAEKF	PAEKF8	38	4	PAEKF
28_2842_PAEKF	PAEKF1	39	4	PAEKF
33_2847_PAEKF	PAEKF6	40	4	PAEKF
34_2848_PAEKF	PAEKF7	41	4	PAEKF
43_2846_PAEKF	PAEKF5	42	4	PAEKF
05_2841_PAETG	PAETG2	43	5	PAETG
15_2979_PAETG	PAETG7	44	5	PAETG
20_2975_PAETG	PAETG3	45	5	PAETG
30_2976_PAETG	PAETG4	46	5	PAETG
39_2977_PAETG	PAETG5	47	5	PAETG
49_2978_PAETG	PAETG6	48	5	PAETG
54_2980_PAETG	PAETG8	49	5	PAETG
55_2840_PAETG	PAETG1	50	5	PAETG
13_2833_CAS	CAS4	51	6	CAS
23_2830_CAS	CAS1	52	6	CAS
29_2832_CAS	CAS3	53	6	CAS
38_2970_CAS	CAS8	54	6	CAS
40_2834_CAS	CAS5	55	6	CAS
44_2835_CAS	CAS6	56	6	CAS
50_2831_CAS	CAS2	57	6	CAS
53_2969_CAS	CAS7	58	6	CAS
19_2981_MIX	MIX	59	7	PAE-CAS-WAR-TG
24_2850_ACT	ACT	60	8	ACT

Tabelle C.2.: Liste der im Datensatz verwendeten 60 Objekte, Teil 2

Literaturverzeichnis

[Al-Kandari und Jolliffe 2005] Al-Kandari, N. M. ; Jolliffe, I. T.: Variable selection and interpretation in correlation principal components. In: *Environmetrics* 16 (2005), S. 659–672. – Zitiert auf Seite 33

[Antoniewicz u. a. 2006] Antoniewicz, M. R. ; Stephanopoulos, G. ; Kelleher, J. K.: Evaluation of regression models in metabolic physiology: predicting fluxes from isotopic data without knowledge of the pathway. In: *Metabolomics : Official journal of the Metabolomic Society* 2 (2006), März, Nr. 1, S. 41–52. – Zitiert auf den Seiten 20, 37

[Backhaus u. a. 2008] Backhaus, K. ; Erichson, B. ; Plinke, W. ; Weiber, R.: *Multivariate Analysemethoden.* Springer, 2008. – Zitiert auf den Seiten 5, 28

[Brereton 2010] Brereton, R. G.: *Chemometrics for Pattern Recognition.* John Wiley & Sons, 2010. – Zitiert auf den Seiten 11, 34, 36, 37, 38, 39, 43, 44

[Brown u. a. 2005] Brown, M. ; Dunn, W. B. ; Ellis, D. I. ; Goodacre, R. ; Handl, J. ; Knowles, J. D. ; O'Hagan, S. ; Spasić, I. ; Kell, D. B.: A metabolome pipeline: from concept to data to knowledge. In: *Metabolomics* 1 (2005), März, Nr. 1, S. 39–51. – Zitiert auf den Seiten 3, 7

[Cattell 1966] Cattell, R. B.: The scree test for the number of factors. In: *Mulitvariate Behavioral Research* 1 (1966), April, Nr. 2, S. 245–276. – Zitiert auf Seite 28

[Croux u. a. 2007] Croux, C. ; Filzmoser, P. ; Oliveira, M. R.: Algorithms for projection-pursuit robust principal component analysis. In: *Chemometrics and Intelligent Laboratory Systems* 87 (2007), S. 218–225. – Zitiert auf Seite 31

[Dillon und Goldstein 1984] Dillon, W. R. ; Goldstein, M.: *Multivariate Analysis: Methods and Applications.* John Wiley & Sons, 1984. – Zitiert auf den Seiten 3, 23

[Downard 2004] Downard, K.: *Mass Spectrometry: A Foundation Course.* Royal Soc of Chemistry, 2004. – Zitiert auf Seite 8

[Dunn u. a. 2005] Dunn, W. B. ; Bailey, N. J. C. ; Johnson, H. E.: Measuring the Metabolome: Current Analytical Technologies. In: *The Analyst* 130 (2005), Mai, Nr. 5, S. 606–25. – Zitiert auf Seite 3

[Fahrmeier u. a. 1996] Fahrmeier, L. ; Hamerle, A. ; Tutz, G.: *Multivariate Statistische Verfahren*. Walter de Gryter & Co., 1996. – Zitiert auf Seite 20

[Flury und Riedwyl 1983] Flury, B. ; Riedwyl, H.: *Angewandte multivariate Statistik. Computergestützte Analyse mehrdimensionaler Daten.* Gustav Fischer Verlag, 1983. – Zitiert auf Seite 23

[Goodacre 2005] Goodacre, R.: Metabolomics – the way forward. In: *Metabolomics* 1 (2005), März, Nr. 1, S. 1–2. – Zitiert auf den Seiten 1, 89

[Goodacre u. a. 2007] Goodacre, R. ; Broadhurst, D. ; Smilde, A. K. ; Kristal, B. S. ; Baker, J. D. ; Beger, R. ; Bessant, C. ; Connor, S. ; Capuani, G. ; Craig, A. ; Ebbels, T. ; Kell, D. B. ; Manetti, C. ; Newton, J. ; Paternostro, G. ; Somorjai, R. ; Sjöström, M. ; Trygg, J. ; Wulfert, F.: Proposed minimum reporting standards for data analysis in metabolomics. In: *Metabolomics* 3 (2007), August, Nr. 3, S. 231–241. – Zitiert auf den Seiten 3, 7, 45

[Goodacre u. a. 2004] Goodacre, R. ; Vaidyanathan, S. ; Dunn, W. B. ; Harrigan, G. ; Kell, D. B.: Metabolomics by numbers: acquiring and understanding global metabolite data. In: *Trends in biotechnology* 22 (2004), Mai, Nr. 5, S. 245–52. – Zitiert auf Seite 8

[Griffiths 2008] Griffiths, W. J.: *Metabolomics, Metabonomics and Metabolite Profiling*. RSC Publishing, 2008. – Zitiert auf Seite 2

[Henrion und Henrion 1995] Henrion, R. ; Henrion, G.: *Multivariate Datenanalyse. Methodik und Anwendung in der Chemie und Verwandten Gebieten*. Springer, 1995. – Zitiert auf Seite 37

[Hunter 2009] Hunter, P.: Reading the metabolic fine print. The application of metabolomics to diagnostics, drug research and nutrition might be integral to improved health and personalized medicine. In: *EMBO reports* 10 (2009), Januar, Nr. 1, S. 20–3. – Zitiert auf den Seiten 2, 3, 8

[Kaiser 1960] Kaiser, H. F.: The application of electronic computer to factor analysis. In: *Educational and Psychological Measurement* 20 (1960), S. 141–151. – Zitiert auf Seite 28

[Lin u. a. 2006] Lin, C. Y. ; Viant, M. R. ; Tjeerdema, R. S.: Metabolomics: Methodologies and applications in the environmental sciences. In: *Journal of Pesticide Science* 31 (2006), Nr. 3, S. 245–251. – Zitiert auf Seite 2

[Nielsen und Jewett 2007] Nielsen, J. ; Jewett, M. C.: *Metabolomics: A Powerful Tool in Systems Biology*. Springer, 2007. – Zitiert auf Seite 1

[Pearson 1901] Pearson, K.: On lines and planes of closest fit to systems of points in space. In: *Philosophical Magazine* 6 (1901), Nr. 2, S. 559–572. – Zitiert auf Seite 20

[Pruscha 2006] Pruscha, H.: *Statistisches Methodenbuch*. Springer, 2006. – Zitiert auf den Seiten 20, 22, 23, 26

[Raykov und Marcoulides 2008] Raykov, T. ; Marcoulides, G. A.: *An Introduction to Applied Multivariate Analysis*. Routledge, 2008. – Zitiert auf Seite 27

[Tabachnick und Fidell 1983] Tabachnick, B. G. ; Fidell, L. S.: *Using Multivariate Statistics*. Harper & Row, 1983. – Zitiert auf Seite 9

[Varmuza und Filzmoser 2009] Varmuza, K. ; Filzmoser, P.: *Introduction to Multivariate Statistical Analysis in Chemometrics*. Crc Pr Inc., 2009. – Zitiert auf den Seiten 9, 10, 22, 29, 30, 31, 33, 34

[Villas-Boas u. a. 2007] Villas-Boas, S. G. ; Roessner, U. ; Hansen, M. A. E. ; Smedsgaard, J. ; Nielsen, J.: *Metabolome Analysis: An Introduction*. John Wiley & Sons, 2007. – Zitiert auf Seite 1

[Vinzi u. a. 2010] Vinzi, V. E. ; Chin, W. W. ; Henseler, J. ; Wang, H.: *Handbook of Partial Least Squares: Concepts, Methods and Applications*. Springer, 2010. – Zitiert auf Seite 34

[Warner 2008] Warner, R. M.: *Applied Statistics: From Bivariate Through Multivariate Techniques*. SAGE Publications, 2008. – Zitiert auf Seite 9

[Westerhuis u. a. 2008] Westerhuis, J. A. ; Hoefsloot, H. C. J. ; Smit, S. ; Vis, D. J. ; Smilde, A. K. ; Velzen, E. J. J. ; Duijnhoven, J. P. M. ; Dorsten, F.: Assessment of PLSDA cross validation. In: *Metabolomics* 4 (2008), Januar, Nr. 1, S. 81–89. – Zitiert auf Seite 43