Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 315

Glyphbasierte Modellanalyse von Klassifikatoren

Robert Gänzle

Studiengang: Informatik

Prüfer: Prof. Dr. Daniel Weiskopf

Betreuer: Dipl.-Inf. Rudolf Netzel

Beginn am: 1. April 2016

Beendet am: 1. Oktober 2016

CR-Nummer: I.3.3

Kurzfassung

In den letzten Jahren gewannen maschinelle Lernverfahren immer mehr an Bedeutung, um große Datenmengen automatisiert zu analysieren. Diese Lernverfahren basieren auf Algorithmen, die aus Daten Muster erkennen, anhand derer die Daten in Klassen eingeteilt werden. Diese Algorithmen bilden aus den Daten Lernmodelle, die möglichst genau die Eigenschaften der Klassen abbilden. Es ist nicht möglich für alle Fälle eine exakte Abbildung zu finden, weshalb inzwischen viele Algorithmen entwickelt wurden, die unterschiedliche Lernfunktionen implementieren. Daher ist es wichtig eine Möglichkeit zu schaffen, die verschiedenen Varianten analysieren und vergleichen zu können. Ein wichtiger Aspekt dabei ist es, zwischen den Modellen unterschiedlich klassifizierte Dateninstanzen genauer vergleichen zu können, um die Ursache des Unterschieds zu erkennen. Um die Analyse zu erleichtern, ist es wichtig, die Daten in einer Form darzustellen, die es dem Menschen erleichtert, wichtigen Informationen schnell aufnehmen zu können. Das Ziel dieser Arbeit ist, ein Konzept für die Visualisierung der klassifizierten Daten zu erarbeiten und in einem prototypischen Programm umzusetzen.

Abstract

In the last years machine learning approaches became more important in order to analyse a big amount of data automatized. Thhese learning approaches are based on algorithms which detect features, by which the data are categorized in classes. THe algorithms generate learning models by the data, which map the properties of the classes as accurately as possible to the model. It is not possible to get an exact mapping for all cases, which is why there have already been many algorithms designed to implement different learning methods. That's why it's important to create a possibility which allows to analyse and compare the different alternatives. An impotant issue as well is to compare the differnt classified data instances between the models to identify the reason for the differnce. To simplify the analysis, it is important to express the data in a manner which allows humans to recognize important inforamtion easier. The goal of this work is to design a concept to visualize the classified data and implement it as a prototypical program.

Inhaltsverzeichnis

1 Einleitung **9**
1.1 Motivation . . . 9
1.2 Ziel der Arbeit . . . 9

2 Grundlagen **11**
2.1 Visualisierung . . . 11
2.1.1 Scientific Visualization . . . 11
2.1.2 Informationsvisualisierung . . . 12
2.1.3 Visualisierung multivariater Daten . . . 13
2.1.4 Visual Analytics . . . 15
2.2 Machinelles Lernen . . . 16
2.2.1 Supervised Learning . . . 16
2.2.2 Unsupervised Learning . . . 18

3 Existierende Arbeiten **21**

4 Anforderungsanalyse **25**
4.1 Anforderungen an die Benutzeroberfläche . . . 25
4.1.1 Übersicht . . . 25
4.1.2 Filter- und Zoomfunktion . . . 26
4.1.3 Details . . . 28
4.2 Anforderungen an das Glyphdesign . . . 29
4.2.1 Sternglyphe . . . 29
4.2.2 Durchschnittsglyphe . . . 29
4.2.3 Differenzglyphe . . . 30
4.2.4 Normalisierung . . . 31
4.3 Anforderungen an das maschinelle Lernen . . . 32

5 Implementierung **35**
5.1 Benutzeroberfläche und Interaktion . . . 36
5.1.1 Menü- und Werkzeugleiste . . . 36
5.1.2 Glyphübersicht . . . 37
5.1.3 Detailansicht mit Differenzglyphe . . . 39
5.1.4 Modelvergleichsübersicht . . . 40
5.1.5 Datensatzübersicht . . . 42

6 Mögliche Arbeitsabläufe **45**
6.1 Analyse klassifizierter Daten 45
6.2 Vergleich von Klassifikatoren 46

7 Zusammenfassung und Ausblick **49**

Literaturverzeichnis **53**

Abbildungsverzeichnis

2.1 Streudiagramm Matrix Beispiel 13
2.2 Parallele Koordinaten Beispiel 14
2.3 Chernoff Faces 15
2.4 Grundprizip einer Support Vector Machine 17
2.5 Grudprinzip des Clustering mit k-Mean 19

3.1 Darstellungsschema von Daten außerhalb des Bildbereichs [JFK16] 22
3.2 Darstellungsschema einer Glyphe mit Wortwolke [JFK16] 22

4.1 Beispielglyphe mit 8 Normalisierungen 28
4.2 Einfache Normalisierungsfunktionen 32
4.3 Normalisierungsfunktionen mit Null in der Mitte 32

5.1 Gesamtansicht des Programms 35
5.2 Auswahl verschiedener Einblendungen von Glyphkomponenten 36
5.3 Auswahlmenü der Normalisierung 37
5.4 Durchschnittsglyphe 38
5.5 Glyphübersicht mit geringer Zoomstufen 39
5.6 Glyphübersicht mit hoher Zoomstufen 39
5.7 Detailansicht mit Differenzglyphe 41
5.8 Modellvergleichsansicht 42
5.9 Eingabemaske für Lernalgorithmus 42
5.10 Detailansicht mit Differenzglyphe 43

1 Einleitung

Auf Grund der großen Menge an Daten, die in der heutigen Zeit produziert, verarbeitet und analysiert werden, haben maschinelle Lernverfahren immer mehr an Bedeutung gewonnen. Diese Verfahren werden als Algorithmen für eine automatisierte Erkennung von Eigenschaften in den Daten implementiert. Die großen Internetkonzerne, wie Google und Facebook, sammeln beispielsweise sehr viele Daten über ihre Nutzer, mit denen sie durch maschinelle Lernverfahren, versuchen Suchanfragen an das Nutzerverhalten anzugleichen oder passende Werbeeinblendungen zu schalten. In der Bilderkennung, beispielsweise zum erkennen von Gesichtern oder anderen Mustern, werden ebenfalls Lernalgorithmen verwendet, um diese effizient in großen Datenmengenen zu finden.

Zusätzlich erfreuen sich die Gebiete der Informationsvisualisierung und Visual Analytics immer größerer Aufmerksamkeit. Denn auch hier geht es darum, abstrakte Daten zu analysieren. Der Schwerpunkt liegt in diesem Fall allerdings darauf, die Daten so zu visualisieren, dass sie von einem Menschen möglichst gut ausgewertet werden können.

1.1 Motivation

Es gibt inzwischen sehr viele verschiedene maschinelle Lernverfahren, die jedoch alle nur für bestimmte Datenarten gut funktionieren. Um einem Menschen die Möglichkeit zu geben, diese Datenmengen zu sichten und zu analysieren, bieten sich grafische Darstellungen aus der Informationsvisualisierung an. Das ist wichtig, um eine Überprüfung der Genauigkeit zu ermöglichen und mögliche fehlerhafte Auswirkungen eines Algorithmuses erkennbar zu machen.

1.2 Ziel der Arbeit

Um die Klassiefizierung von Daten beim maschinellen Lernen besser analysieren zu können, soll bei dieser Arbeit ein Framework entstehen. Bei diesem soll durch eine glyphbasierte Visualisierung der Vergleich einzelner Glyphen erleichtert werden. Dies soll es ermöglichen, leichter nachzuvollziehen aus welchen Grüden einzelne Dateninstanzen unterschiedlich klassifiziert

wurden. Damit soll ein Vergleich verschiedener Klassifikatormodellen ermöglicht werden, indem die Dateninstanzen, die sich bei verschiedenen Klassifikatoren unterscheiden, miteinander verglichen werden.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Grundlagen: Hier werden die Grundlagen dieser Arbeit beschrieben. Diese beinhalten eine kurze Einführung in die Visualisierung und Interaktionskonzepte, sowie das maschinelles Lernen.

Kapitel 3 – Existierende Arbeiten: In diesem Kapitel wird auf Arbeiten eingegegangen, deren Thematik mit dieser Arbeit korrelieren.

Kapitel 4 – Anforderungsanalyse: Dieser Teil behandelt die konzeptionellen Anforderungen an die Visualisierung und Interaktionsmethoden, sowie im Speziellen das Glyphdesign, das verwendet wird.

Kapitel 5 – Implementierung: Dieses Kapitel enthält die Beschreibung, wie die Anforderungen aus dem voreherigen Kapitel, in der Implementierung, umgesetzt wurden.

Kapitel 6 – Mögliche Arbeitsabläufe: In diesem Kapitel werden mögliche Arbeisabläufe, zur Analyse der Daten, beschrieben.

Kapitel 7 – Zusammenfassung und Ausblick fasst die erarbeiteten Konzepte zusammen und stellt einige Anknüpfungspunkte zur Erweiterung dieser vor.

2 Grundlagen

In diesem Kapitel werden die Grundlagen der Visualisierung und des maschinellen Lernes beschrieben.

2.1 Visualisierung

In diesem Abschnitt wird ein kurze Einführung in die Grundlagen der Visualisierung gegeben. Grundsätzlich geht es darum, die Daten für den Menschen besser lesbar zu machen, denn der Mensch kann in Bildern und Grafiken schneller Muster und Eigenschaften erkennen als in einer Tabelle voller Datenwerte.

Zu den drei Hauptgebieten der Visualisierung zählen die wissenschaftliche Visualisierung (engl. Scientific Visualization, Abschnitt 2.1.1), die Informationsvisualisierung (engl. Information Visualization, Abschnitt 2.1.2) und Visual Analytics (Abschnitt 2.1.4). In den nächsten drei Abschnitten werden die drei Bereiche voneinander abgegrenzt, wobei die Arbeit sich hauptsächlich der Informationsvisualisierung bedient und einige Grundzüge der Visual Analytics verwendet.

2.1.1 Scientific Visualization

Dieser Bereich beschäftigt sich hauptsächlich mit der Darstellung von physikalischen Modellen, die aus physikalischen Messungen gewonnen oder durch Simulationen erzeugt werden. Dabei geht es darum Oberflächen, Volumen, Strömungen, Verformungen, Beleuchtungen, Temperatureffekte oder andere physikalische Eigenschaften realistisch in einer Grafik darzustellen. Wichtig ist, dass direkt die Wirklichkeit nachgebildet wird und kein Spielraum für andere Interpretationen der Daten ermöglicht wird. Aus Messungen gewonnene Daten können in der Medizin beispielsweise aus einem Computertomographen sein. Diese werden als ein dreidimensionales Modell dargestellt, um den gescannten Körper nachzubilden. Häufige Beispiele für die Visualisierung von Simulationsdaten sind Strömungen, die im Fahrzeugbau und der Luft- und Raumfahrttechnik eine wichtige Rolle spielen. Dabei werden die Strömungen durch Strömungslinien sichtbar gemacht, um ungünstige Turbulenzen zu erkennen. In diesem Zusammenhang werden meistens auch die Kräfte, die auf die Bauteile wirken, dargestellt.

Die Anzeige von Kartendaten wird auch oft mit der wisseschaftlichen Visualisierung verbunden, da diese geographische Messdaten abbildt. In der Regel wird jedoch die geographische Höhe nicht direkt in eine dritte Dimension übertragen, sonder auf sogenannte Höhenlinien reduziert. Heutzutage ist es zwar Standard, in Karten mit Satellitenbildern die Realität genau abzubilden. Dennoch ist es häufig sinnvoller verschiedene Biome stark zu abstrahieren, um einen besseren Überblick zu erhalten. Desweitern werden vielen Karten zusätzliche Informationen wie Straßen- und Städtenamen, Hinweise zu Tankstellen, Banken und viele weitere Orte von Interesse (engl. Points of Interests) hinzugfügt.

Diese Abstraktionsvorgänge, die gerne in Karten verwendet werden, stammen eigentlich aus dem Gebiet der Informationsvisualisierung, welche wir im nächsten Abschnitt 2.1.2 genauer betrachten werden.

2.1.2 Informationsvisualisierung

Die Informationsvisualisierung beschäftigt sich, gegenüber der wissenschaftlichen Visualisierung, mit der grafischen Repräsentation abstrakter Daten. Dabei liegt der Schwerpunkt nicht auf der möglichst realistischen Darstellung der Daten. Stattdessen werden alle relevanten Informationen, die nicht ohne Weiteres ersichtlich sind, in einer Grafik so dargestellt, dass sie von einem Menschen schnell und einfach aufgenommen werden können. Dies ist trotz vieler existierender algorithmischer Lösungen zum Finden von Mustern innerhalb der Daten in vielen Fällen immernoch sinnvoll. Das liegt daran, dass das menschliche Auge, durch eine gute Visualisierung, in vielen Fällen besser Muster erkennen kann als ein Computer.

Zur Unterstützung der Visualisierung gibt es einige Konzepte zur Darstellung von Daten. Die heute oft als Gestalltgesetze bekannten Regeln über die Wahrnehmung, gehen auf die Arbeit von Wertheimer [Wer12] zurück. Diese sind wichtig zu kennen, um durch sie verschiedene Zusammenhänge und Korrelationen zwischen den grafischen Bestandteilen in einer Visualisierung herstellen zu können. Dazu zählen die folgenden Punkte, wodurch die Warnehmung beeinflusst wird.

- Anhand der Positionierung von Elementen kann eine visuelle Abtrennung verschiedener Gruppen erreicht werden.
- Aufgrund von Ähnlichkeiten können Elemente als zusammengehörend wahrgenommen werden. Das kann sowohl über die Form, als auch über die Farbe erreicht werden.
- Durch die Umschließung einzelner Bestandteile können diese als Gruppe erkennbar gemacht werden.
- Verbindet man grafische Elemente, beispielsweise durch Kanten wie bei Graphen miteinander, so wird deren Zusammengehörigkeit ersichtlich.
- Kontinuierliche Verläufe und bestimmte Formen können ebenfalls Zusammenhänge ersichtlich machen.

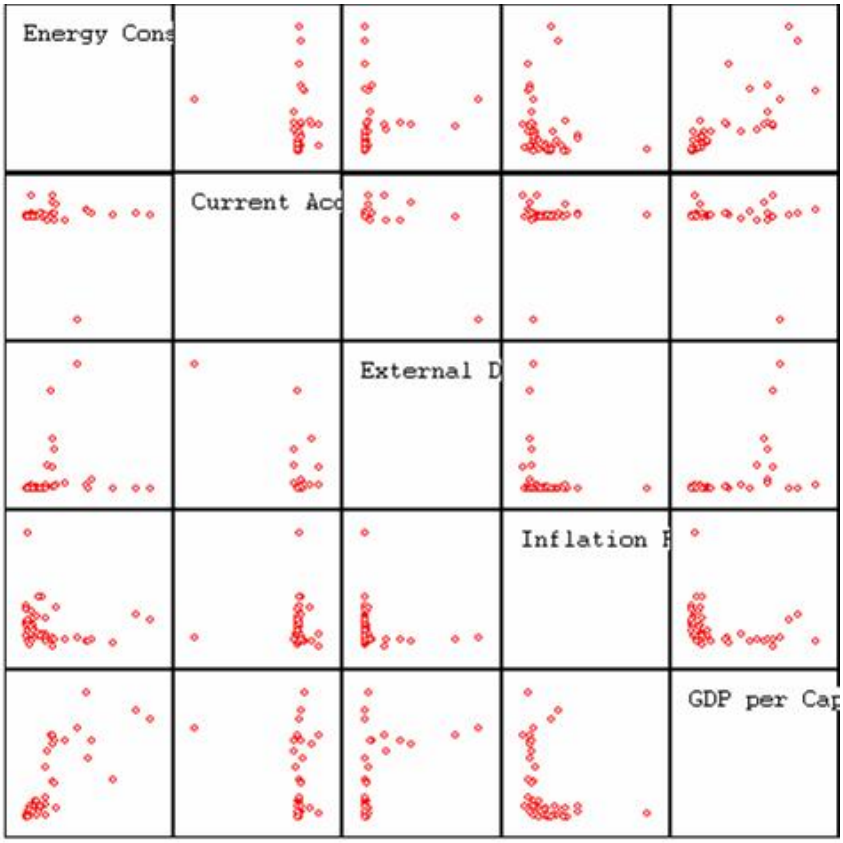

Abbildung 2.1: Hier ist ein Beispiel einer Streudiagramm Matrix zu sehen. [xpl06]

In dieser Arbeit geht es, aufgrund der vielen mehrdimensionalen Dateninstanzen, die beim maschinellen Lernen auftreten, um die Visualisierung von multivariaten Daten. In Abschnitt 2.1.3 werden verschiedene Möglichkeiten zur Visualisierung dieser Daten genauer beschrieben.

2.1.3 Visualisierung multivariater Daten

Es gibt verschiede Möglichkeiten multivariate Daten visuell darzustellen. Im Folgenden werden kurz einige davon vorgestellt.

Tabellen

Es ist möglich die Daten direkt in einer Tabelle anzuordnen, ohne weitere grafische Elemente hinzuzufügen. Damit können zwar gut die exakten Werte gelesen werden, es ist jedoch schwer irgendeine Form von Zusammenhängen zu erkennen.

Streudiagramm Matrix

Eine Streudiagramm Matrix (engl. Scatterplot Matrix) wird gebildet, indem alle Variablen als Zeilen, sowie als Spalten aufgetragen werden. In jeder Zelle $a_{i,j}$ der Matrix wird ein Streudiagramm mit den Variablen i und j als Achsen dargestellt. Da es sich um eine Matrix handelt, werden in beide Richtungen alle Dimensionen aufgetragen, wie es in dem Beispiel auf der Abbildung 2.1 zu sehen ist. Dadurch existiert zu jeder Zweierkombination der Achsen ein Streudiagramm, in welchem die Korrelation dieser beiden Dimensionen zueinander analysiert werden kann.

Parallele Koordinaten

Bei der Visualisierung mit parallelen Koordinaten werden für alle Variablen parallele Achsen nebeneinander in äquidistanten Abständen angeordnet. Die Daten der jeweiligen Instanzen werden mit den Werten der jeweiligen Variablen auf jeder Achsen miteinander verbunden. Im Gegensatz zu der Streudiagramm Matrix, ist nicht jede Achse mit jeder anderen direkt verbunden, wie in Abbildung 2.2 zu sehen ist. Es ist also nur möglich, die Korrelationen zwischen zwei nebeneinander liegenden Achsen genauer zu sehen. Um solche Probleme zu lösen, beschreiben Heinrich und Weiskopf [HW13] in ihrer Arbeit verschiedene Verfahren und Modelle, um parallele Koordinaten darzustellen, zu analysieren und mit ihnen zu interagieren.

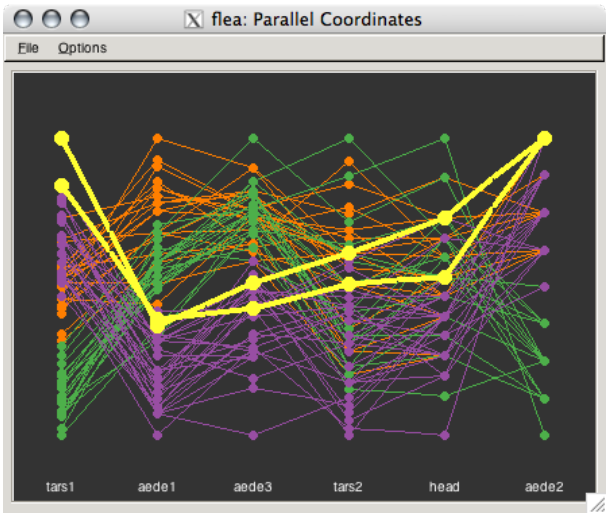

Abbildung 2.2: Ein Beispiel, wie parallele Koordinaten grundlegend aussehen. [Coo08]

Glyphen

Bei Glyphen handelt es sich um eine Repräsentation, bei der alle Variablen einer Instanz in einer Grafik dargestellt werden und die einzelnen Werte durch Form, Farbe, Größe und

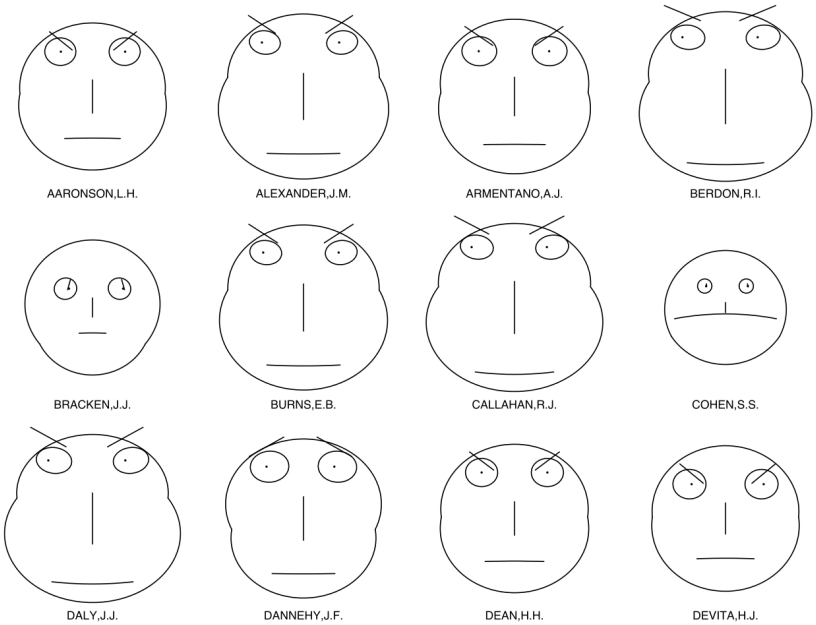

Abbildung 2.3: Ein Beispiel, wie die Chernoff Faces aussehen können. Der Autor [Ave10] verwendete dabei Daten über die Bewertung von Richtern in den USA.

Position unterschieden werden. Das Ziel ist es, verschiedene Glyphen durch die verschiedenen Anordnungen der Werte innerhalb der Bilder, anhand der genannten grafischen Eigenschaften, vergleichen zu können. Ward [War02] stellt in seiner Arbeit einige verschieden Glyphtypen und deren Eigenschaften vor.

Es gibt dabei viele verschiedene Varianten, von denen einige eher abstrakt sind. Bekannte Vertreter sind Profilglyphen, bei denen es sich um kleine Diagramme handelt. Anhand ihrer Konturen können sie schnell verglichen werden. Eine weitere Variante wäre die Sternglyphe, bei welcher die Variablenwerte kreisförmig aufgetragen werden, wodurch eine sternförmige Form entsteht.

Unter den weniger abstrakten Vertretern gibt es die so genannten Chernoff Faces [Che73], welche verschiedene Gesichter darstellen. Dabei werden die Variablen auf verschiedene Gesichtszüge und -formen übertragen, um die Daten auf die dazu passenden Emotionen abzubilden. In Abbildungen 2.3 ist ein Besipiel zu sehen, wie solche Gesichter aussehen können.

2.1.4 Visual Analytics

Im Gebiet der Visual Analytics geht es darum, Visualisierungsmethoden und Interaktionsmöglichkeiten so zu verknüpfen, dass Menschen in der Lage sind, möglichst einfach Informationen aus den Daten zu lesen. Dabei ist es auch möglich, die Daten durch maschinelle Lernverfahren

vorverarbeiten zu lassen. Dadurch werden die Daten, schon bevor sie dem Anwender präsentiert werden, in einer Form zusammengestellt, die weniger Anpassungen durch Interaktionen, mit der Visualisierung benötigen.

Ein Grundlegendes Prinzip wurde von Shneiderman [Shn96] als das „Visual Information Seeking Mantra“ bezeichnet, was soviel heißt, wie „Mantra zur visuellen Informationssuche“. Das Mantra lautet auf deutsch etwa „Zuerst die übersicht, dann Zoomen und Filtern, zum Schluss Details auf Abruf.“ (engl. „Overview first, zoom and filter, then details-on-demand.“, original von Shneiderman [Shn96]). Das beschreibt ganz gut, worum es bei der visuellen Analyse geht. Es kann daraus gefolgert werden, wie eine Visualisierung der Informationen aufgebaut werden sollte.

Die Aussage des Mantras ist, dass die Visualisierung zuerst eine Übersicht der Informationen liefern soll. Desweiteren soll es möglich sein, Ausschnitte in der Übersicht heranzoomen zu können und mit Filtern so anzupassen, dass die gewünschten Informationen bestmöglich präsentiert werden. Durch weiter interaktive Methoden werden Details durch interaktives abrufen angeboten. Ein großer Teil dieser Aufgaben wird schon durch die Informationsvisualisierung abgedeckt. Der Schwerpunkt der Visual Analytics liegt dabei auf den Zoom- und Filtermöglichkeiten. Es geht also darum durch Interaktivität die Darstellung anzupassen und die Daten so zu Filtern, dass nur die benötigten Informationen gut ablesbar sind. Diese Aufgabe ist ein iterativer Prozess, bei welchem nach Änderungen an den Filtern die Visualisierung aktualisiert wird. Danach wird die Anzeige analysiert und gegebenenfalls immer wieder angepasst, bis die gewünschten Informationen offengelegt sind.

2.2 Machinelles Lernen

Beim maschinellen Lernen geht es grundsätzlich darum, durch Algorithmen automatisiert Eigenschaften aus Daten zu erlernen und auf neue Daten anzuwenden. Die folgenden zwei Kathegorien, das überwachte Lernen (engl. supervised Learning) und das nicht überwachte Lernen (engl. unsupervised Learning) aus dem Gebiet des maschinellen Lernens, sind für diese Arbeit relevant und werden hier behandelt. Anhand von Beispielen werden in den nächsten beiden Abschnitten diese beiden Bereiche beschrieben.

2.2.1 Supervised Learning

Beim überwachten Lernen bekommt der Algorithmus einen Datensatz, der bereits durch einen Spezialisten den passenden Klassen zugeordnet wurde. Anhand dieser vorklassifizierten Daten berechnet der Algoritmus ein Modell, in welchem die Klassen so aufgeteilt sind, wie es diese Daten ermöglicht haben. Anhand des erzeugten Modells können jetzt neue unklassifizierte Daten automatisiert den Klassen zugeteilt werden.

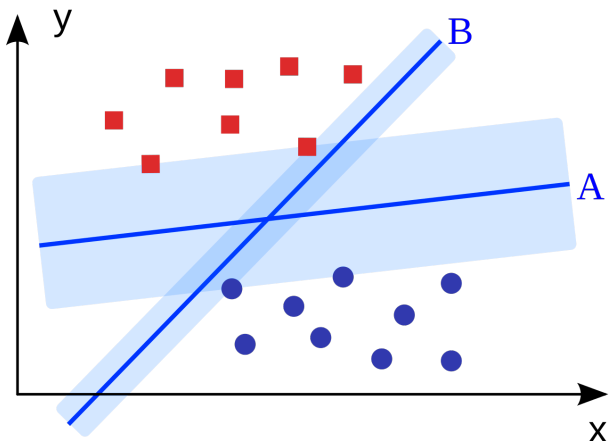

Abbildung 2.4: Grundprizip einer Support Vector Machine reduziert auf einen zweidimensionalen Vektorraum. Dabei wurde der Abstand zwischen A und den Vektoren entlang der Grenze maximiert. Ist dies nicht der Fall, wie bei B, so wird die freie Fläche zwischen den Klassen kleiner und die Klassen werden nicht so gut von einander getrennt. [Enn10]

Suport Vector Machine

Bei einer Support Vector Machine (SVM) werden alle Datenelemente jeweils von einem Vektor repräsentiert. Zwischen den Vektoren werden Ebenen so durch den Vektorraum gelegt, dass diese, die Vektoren einer Klasse von denen einer anderen Klasse, abgegrenzt werden. Dabei werden die Abstände zwischen den Ebenen und den Vektoren im Grenzbereich maximiert. Dies wird in Abbildung 2.4 durch die Begrenzung auf einen zweidimensionalen Vektorraum vereinfacht dargestellt. Die Grenzlinie A in der Abbildung entspricht dabei der Ebene die auf den maximalen Abstand optimiert wurde und somit gegenüber B die größtmögliche Abgrenzungsfläche zwischen den Klassen bildet. In [Pla98] wird eine Variante beschrieben, wie sie im Weka Framework [HFH+09] implementiert zu finden ist.

Entscheidungsbaum

Ein Entscheidungsbaum ist ein binärer Baum, der für jedes Attribut eine Knotenebene enthält. Dabei kann sich die Anzahl der Knoten in jeder Ebene zu einem Vollständigen Binärbaum verdoppeln. Es muss allerdings kein vollständiger Baum gebildet werden, wenn es Attribute gibt, die für manche Entscheidungen nicht benötigt werden. Dann kann der Pfad auch schon früher in einem Blattknoten enden, in welchem die Klasse steht. An jedem Knoten wird entschieden, ab welchem Wert des Attributs zu welchem der beiden Kindknoten weitergegangen wird. Der Entscheidungsbaum wird dabei mit den Attributwerten vorklassifizierter Daten aufgebaut. Im Weka Framework [HFH+09] findet sich ein Algorithmus der auf einem Entscheidungsbaum basiert und „J48“ genannt wird. Dieser basiert wiederum auf der Arbeit von Quinlan [Qui93].

Neuronalem Netz mit Backpropagation

Rumelhart, Hinton und Williams [RHW88] beschreiben in ihrer Arbeit, wie neuronale Netze mit Backpropagation, durch Anpassung der Gewichtung maschinell gelernt werden können. Ein neuronales Netz ist so aufgebaut, dass jedes Attribut eines Datenelements einen eigenen Eingabeknoten des Netzes hat. Jede Klasse, in welche die Daten eingeteilt werden sollen, erhält einen Ausgabeknoten. Zu Beginn werden die Eingaben in das neuronale Netz mit einer Gewichtung versehen. Mit dieser Gewichtung wird jeweils eine Übertragungsfunktion bestimmt, die die Eingabe in das Netz ergibt. Die Aktivierungsfunktion des Netzes bestimmt anschließend, ob die Ausgabe aktiveirt wird oder nicht. Wird ein Element in das Netz eingegeben, kann die Aktivierung der entsprechenden Ausgabe mit der Klasse in den vorklasifizierten Daten verglichen werden. Stimmt das Ergebnis nicht mit der Klasse überein, so werden die Gewichte durch die Backpropagation neu festgelegt. Dazu wird eine Fehlerfunktion verwendet, anhand derer das neue Gewicht berechnet wird.

2.2.2 Unsupervised Learning

Beim nicht überwachten Lernen ordnet ein Algorithmus die Daten, aufgrund von deren Ähnlichlkeiten, in Gruppen ein. Dabei wird keine vorherige Klassifikation einer Teilmenge der Daten durch einen Experten benötigt. Bekannte Verfahren, die dafür verwendet werden, sind die sogenannten Clusteringverfahren. Dabei werden die Daten in sogenannte Cluster eingeteilt, welche Gruppen von ähnlichen Datenelementen entsprechen. Eine der häufig verwendeten Clusteringmethoden verwendet den k-Meams Algoritmus, der im folgenden beschrieben wird.

Clustering mit k-Means

Der k-Means-Algrithmus basiert auf der Arbeit von [Llo82] und wurde von vielen Wissenschaftlern wieder aufgegriffen. Im Weka Framework [HFH+09] ist unter anderem der von Arthur und Vassilvitskii [AV07] vorgestellte Algorithmus, names k-Means++ implementiert, der eine Erweiterung des k-Means ist.

Für die Beschreibung des k-Means Clustering werden folgende Annahmen festgelegt.

- X sei die Menge der Punkte im Vektorraum.
- $x_i \in X$ sei der i-te Datenpunkt, der zu clusterisierenden Vektordaten.
- k sei die Anzahl der Cluster.
- C sei die Menge der Clusterzentren.
- $c_j \in C$ sei das j-te Clusterzentrum.

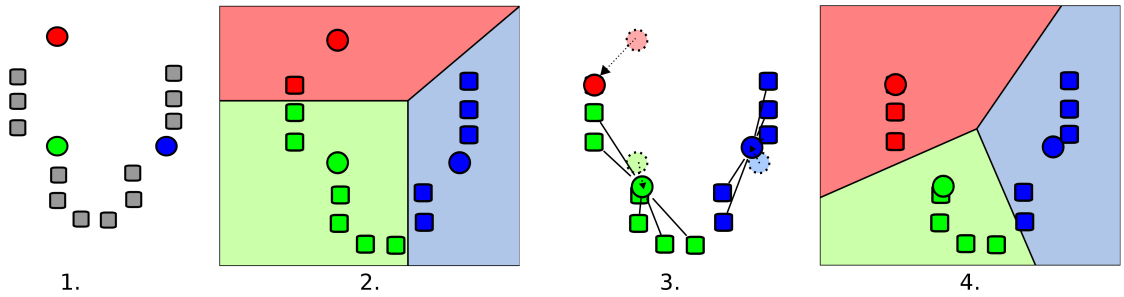

Abbildung 2.5: Veranschaulichung der Schritte zur Clusterberechnung mit k-Means. 1. Zufällig plazierte Clusterzentren [Wes07d], 2. Alle Elemnte werden entsprechendem Clusterzentrum zugeordnet [Wes07a], 3. Clusterzentren werden neu berechnet [Wes07b], 4. Elemente werden den neuen Zentren zugeordnet [Wes07c].

- $C_j \subseteq X$ sei die Menge der Elemente aus X, die zum Cluster C_i gehören.

Das Clustering mit k-Means wird von Arthur und Vassilvitskii [AV07] mit den folgenden Schritten beschrieben.

1. Zuerst werden zufällig eine festgelegte Anzahl k an Clusterzentren $C = \{c_1, ..., c_k\}$ im Vektorraum verteilt.
2. Jedes Element $x_i \in X$ wird dem Cluster j zugeordnet, dessen Clusterzentrum c_j dem Element x_i am nächsten ist.
3. Danach werden die Clusterzentern neu bestimmt. Das neue Zentrum wird mit $c_j = \frac{1}{|C_j|} * \sum_{x_i \in |C_j|} x_i$ berechnet.
4. Solange bis keine Veränderung der Clusterzentren C mehr auftritt, wieder bei Schritt 2. beginnen.

In Abbildung 2.5 sind die Schritte in einfacher zweidimensionaler Form veranschaulicht.

3 Existierende Arbeiten

In diesem Kapitel wird auf existierende Arbeiten eingegangen, die sich ebenfalls mit der Darstellung multivariater Daten beschäftigen.

Arbeit über interaktives Lernsystem

Hier wird eine Arbeit vorgestellt, die sich mit der Visualisierung und Analyse von Klasifikatormodellen maschineller Lernverfahren beschäftigt.

Hoferlin et al. [HNH+12] haben in ihrer Arbeit ein System für interaktives Lernen beschrieben. Beim aktiven Lernen wird vom Algorithmus eine Auswahl an aussagekräftigen Beispielen aus der unklassifizierten Datenmenge ausgewählt. Diese werden einem Experten zur Kathegorisierung in eine Klasse vorgelegt. Beim interaktiven Ansatz der in der Arbeit von Hoferlin et al. beschreiben wird, soll der Experte zusätzlich direkt das Modell bewerten. Mit dieser Kenntnis soll er selbst Beispiele auswählen, um das Modell zu verbessern.

Eingesetzt wird das System im Kontext von Videoanalysen, bei der komplexe Klassifikatoren nötig sind, um aufwändige Merkmale zu klassifizieren. Um die komplexen Merkmalsvektoren zu analysieren, haben die Autoren der Arbeit eine kaskadenförmige Streudiagrammvisualisierung in das Systrem integriert. Diese Streudiagramme enthalten die Verteilung der Klassen, die den Instanzen zugewiesen wurden. Für jede Phase des Klassifikators wird dabei ein weiteres Diagramm erzeugt.

Der Ansatz der in dieser Arbeit verfolgt wird, ist es die einzelnen Dateninstanzen genauer zu untersuchen. Um diese Klassifizier noch auf eine andere Art und Weise zu untersuchen, könnte es hilfreich sein die klassifizierten Instanzen genauer darzustellen und deren Eigenschaften in einer Glyphvisualisierung, wie sie in dieser Arbeit behandelt wird, anzuzeigen. Damit könnten dann die Eigenschaften der einzelnen Dateninstanze verglichen und analysiert werden, wenn sonst nicht klar wird, wie die Klasseneinordnung zu stande kam.

Arbeit zu Glyphvisualisierung

Im Bereich des Glyphdesigns gibt es bereits Arbeiten, die sich intensiver mit deren grafischer Darstellung, Positionierung und Interaktionsmöglichkeiten befassen. Im Folgenden wird eine Arbeit aufgegriffen, die sich in einem etwas anderen Kontext die Glyphvisualisierung zu nutze macht.

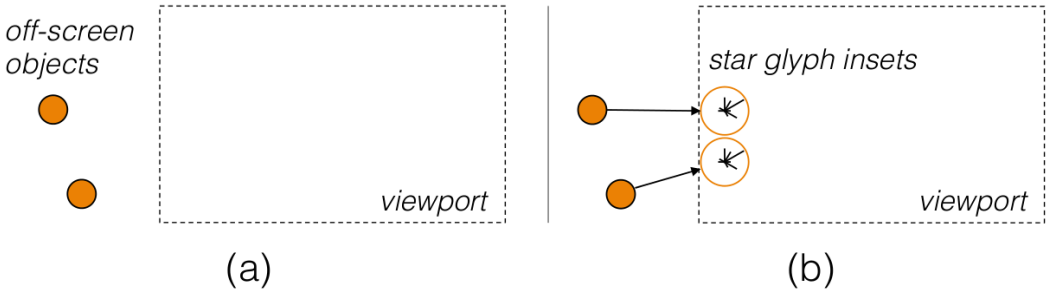

Abbildung 3.1: Die Abbildung von Jäckle, Fuchs und Keim [JFK16] zeigt das Schema, wie sie Daten außerhalb des Bildbereichs als Sternglyphen am Rad des sichtbaren Bereichs in ihrer Arbeit dargestellt haben.

Jäckle, Fuchs und Keim [JFK16] stellen in ihrer Arbeit ein Konzept vor, das mit Sternglyphen multivariate Daten darstellt. Dabei handelt es sich bei den verwendeten Daten, um welche die den geologischen Standort beinhalten. Zur Darstellung haben sie diese, deshalb auf eine Kartendarstellung abgebildet. Im sichtbaren Kartenbereich kann ein Auschnitt der Karte markiert werden. Alle Datenwerte der darin entahltenen Datenpunkte, werden dabei zu einer einzelnen Sternglyphe zusammengerechnet und in einer danebenliegenden Ansicht angezeigt.

Ein weitere Punkt, um den es bei ihrer Arbeit ging, war es Daten außerhalb des dargestellten Bildauschnittes anzuzeigen. Dafür zeichneten sie die Glyphen, die außerhalb des sichtbaren Bereichs liegen, an den Rand der Bildfläche. In Abbildung 3.1 haben sie in einer einfachen Schemadarstellung gezeigt, wie die Werte außerhalb in den Bildbereich abgebildet werden.

In dieser Arbeit soll das Interesse mehr auf dem Glyphdesign liegen. Deshalb wird im Folgenden das Glyphdesign von [JFK16] angeschaut. Die Autoren wählten eine einfache Sternglyphe als gewünschte Darstellungsform. Um anzuzeigen worum es sich bei den Achsen handelt, zeigen sie bei der Auswahl einer Glyphe, eine Wortwolke an. Die Worte in der Wolke sind die Namen der Attribute und deren Größe korreliert mit der Größe des Attributwertes. Bei

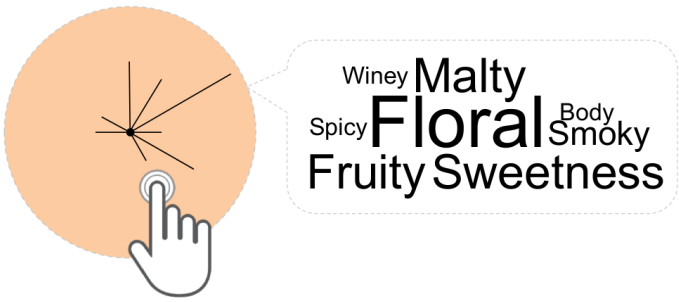

Abbildung 3.2: Die Abbildung von [JFK16] zeigt eine Wortwolke, die durch Auswahl einer Glyphe angezeigt wird.

gleichgroßen Worten ist allerding nicht genau zu erkennen, durch welche Achse sie in der Glyphe repräsentiert wird. In Abbildung 3.2 ist diese Darstellung wie sie beschrieben wurde dargestellt.

Da in meiner Arbeit eher abstrakte Daten verwendet werden, ist die Sternglyphe zwar ebenso eine gute Wahl, allerdings sollen in den Details genauere Daten sichtbar werden als eine Wortwolke. Dafür reicht eine einfache zeilenweise Anordnung, die die Glyphen nach Klassen und anhand der Ähnlichkeit sortiert, anstatt sie nach Positionsdaten anzuordnen. Dabei können bei einer großen Datenmenge zwar auch nicht alle Glyphen im Bildausschnit dargestellt werden, aber es reicht dafür eine einfache Scrollfuntion, um durch die Ansicht zu rollen.

4 Anforderungsanalyse

In diesem Kapitel wird auf die Anforderungen des Programms und die grundlegende Konzeptidee eingegangen. Die Hauptaufgabe des Programms soll die Analyse der Klassifizierung von Daten aus dem Bereich des maschinellen Lernens sein. Dafür wird eine passende Visualisierung der Daten und entsprechende Interaktionsmöglichkeiten benötigt, die in den folgenden Abschnitten beschrieben werden.

4.1 Anforderungen an die Benutzeroberfläche

Die Benutzeroberfläche soll einige Standardfunktionen enthalten. Dazu zählen ein Menü, sowie eine Toolbar, um diverse Interaktionen und Einstellungen vorzunehmen und Daten zu laden. Im weiteren Verlauf des Kapitels werden die benötigten Funktionen besprochen. Wie diese durch die Menüs verwendet werden, wird in Kapitel 5 gezeigt, in welchem die Umsetzung des Programms beschrieben wird.

Um eine Analyse der Daten zu ermöglichen, soll das Programm die grundlegenden Visualisierungs- und Interaktionskonzepte von Ben Shneidermans „Visual Information-Seeking Mantra“ [Shn96] abdecken. Es wird also eine Übersicht über die Daten, Filter- und Zoomfunktionen, sowie auch eine Möglichkeit mehr Details anzuzeigen, benötigt. In den nächsten Abschnitten wird beschrieben wie diese Konzepte in dieser Arbeit aussehen sollen.

4.1.1 Übersicht

Das Programm soll die folgenden beiden Übersichten beinhalten.

Glyphenübersicht

Das Programm soll eine Übersicht über einen geladenen Datensatz enthalten, in welcher die einzelnen Datenelemente durch Glyphen angezeigt werden sollen. Die geforderte Konzeption und das Desigen der Glyphen wird in Abschnitt 4.2 beschrieben. Falls es sich um einen bereits klassifizierten Datensatz handelt, sollen die Instanzen nach ihren zugeordneten Klassen aufgeteilt werden. Zu jeder Klasse soll eine repräsentative Glyphe wie in Abschnitt 4.2.2 berechnet werden, die den Durchnitt aller Instanzen der Klasse beinhaltet. Innerhalb einer Klasse sollen

die Daten zusätzlich durch eine weiter Metrik sortiert werden, die die Ähnlichkeiten der Instanzen in der Klasse zusätzlich gruppiert. Es bietet sich ein unüberwachter Lernalgorithmus, wie in den Grundlagen in Abschnitt 2.2.2 beschrieben, zur Clusterisierung der Datenelemente an. Innerhalb eines Clusters sollen die Daten nochmals nach der euklidschen Distanz zum Clusterzentrum sortiert werden.

Modellübersicht

Es soll noch eine zweite Übersicht im Programm enthalten sein, welche mehrere maschinelle Lernmodelle zugleich anzeigen und einen Vergleich dieser ermöglichen soll. Die einzelnen Dateninstanzen können dabei durch farbkodierte Zellen dargestellt werden, wobei jede Klasse eine eigene Farbe benötigt. Stellt man diese Zellen in einer Matrix $\{Instanzen\} \times \{Modelle\}$ dar, so liegen die gleichen Instanzen immer in der selben Zeile. Aus diesem Grund ist es möglich, bei unterschiedlichen Farben innerhalb einer Zeile direkt zu sehen, welche Instanzen von verschiedenen Modellen unterschiedlich klassifiziert wurden. Das Programm benötigt eine Eingabemaske, um verschiedene Algorithmen für die Modelle auszuwählen, mit welchen die Testdaten klassifiziert werden. Die Anforderungen dafür werden in Abschnitt 4.3 beschrieben.

4.1.2 Filter- und Zoomfunktion

In diesem Abschnitt werden die verschiedenen Methoden beschrieben, die dem Programm zum Filtern und Anpassen der angezeigten Daten zur Verfügung stellen soll.

Zoomfunktion

In der Glyphübersicht soll es ermöglicht werden, die Glyphen kleinstmöglich für eine sinnvolle Anzeige darzustellen, um einen Überblick über möglichst viele Glyphen und die Struktur der Daten zu erhalten. Um die Glyphen für eine genauere Vergleichbarkeit zu vergrößern, wird eine Zoomfunktion benötigt. Dadurch ergibt sich ein kleinerer Ausschnitt mit mehr Detailgenauigkeit des Datensatzes.

Glyphauswahl

Für die im nächsten Abschnitt 4.1.3 beschriebene Detailansicht sollen einzelne Instanzen markiert werden können. Dabei muss interaktiv unterschieden werden, ob das ausgewählte Element als Minuend oder Subtrahend für die Berechnung der Differenzglyphe in der Detailansicht verwendet wird.

Normalisierungsauswahl

Je nach Wertebereich der Attribute können unterschiedliche Normalisierungen sinnvoll sein. Es werden im Folgenden ein paar mögliche Enscheidungskriterien dafür aufgelistet.

- Beinhaltet ein Attribut einen besonders weiten Wertebereich gegenüber den anderen Attributen, hat es keine sichtbare Auswirkung auf die Glyphe, falls die Attribute gemeinsam normalisiert werden. Der Grund dafür ist, dass der enge Wertebereich der anderen Attribute nur einen minimalen Bereich auf der Achse einnimmt, wodurch sich die Unterschiede in einem zu kleinen Bereich bewegen. In diesem Fall wäre es sinnvoller die Attribute unabhängig zu normalisieren.
- Beinhaltet der Wertebereich aller Attribute positive und negative Werte, ist es sinnvoll einen gemeinsamen Nullpunkt auf den Achsmitten festzulegen. Enthalten sie allerdings nur postive Werte, so ist es besser, wenn der Nullpunkt in der Mitte der Glyphe ist, um die ganze Achse zur Darstellung zu verwenden.

Je nach Wertebereiche der Datensätze soll eine passende Normalisierung ausgewählt werden können. In Abschnitt 4.2.4 werden die Funktionen für die verschiedenen Normalisierungen, die im Programm verwendet werden sollen, erklärt. Wie sich die dort beschriebenen Normalisierungsfunktionen auswirken, ist in der Abbildung 4.1 zu sehen. Dort wird eine Beispielglyphe einmal mit jeder Normalisierung abgebildet.

Glyphkomponenten

Im Folgendenden sind die verschiedenen Komponenten der Glyphen aufgelistet, welche ein- und ausgeblendet werden können sollen.

- Achsen der Attribute
- Kreisumrandung um die Glyphe
- Die Kontur der Datenwerte

Ist die Glyphansicht ganz herausgezoomt, ist es sinnvoll nur die Kontur anzuzeigen, um auf den kleinen Glyphen noch etwas zu erkennen. Wird dagegen nah herangezoomt, kann man mit den Achsen die Attribute besser von einander unterscheiden.

Übersichtenverknüpfung

Die beiden Übersichtsansichten aus Abschnitt 4.1.1 sollen so verknüpft werden, dass das Markieren von einzelnen Instanzen gleichzeitig in beiden Ansichten angezeigt wird. Damit ist es in die eine Richtung möglich, durch die Auswahl zu sehen, wie eine bestimmten Glyphe von den verschiedenen Modellen klassifiziert wurde. In die andere Richtung können Daten

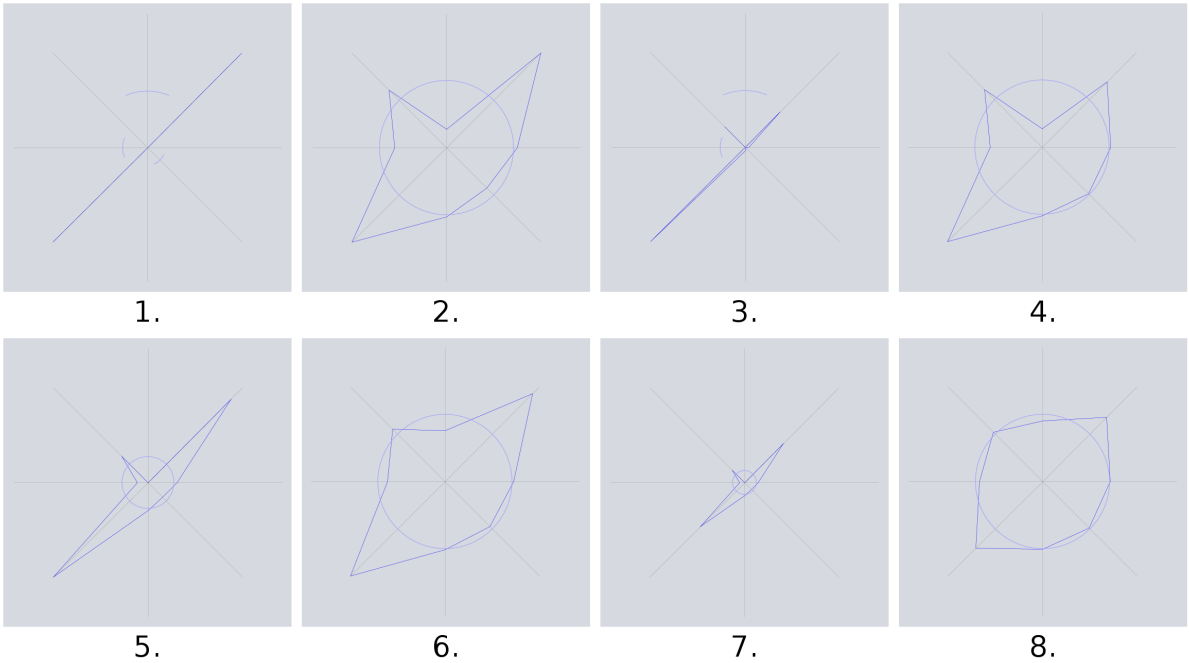

Abbildung 4.1: Hier sind zu den acht beschreiebenen Normalisierung jeweils eine Glyphe abgebildet. Dabei basiert jede Glyphe auf der selben Dateninstanz. Die Nummerierung entspricht der aus Abschnitt 4.2.4.

ausgewählt werden, bei denen es viele unterschiedliche Einordnungen durch die Modelle gab, um durch die Glyphdarstellung und deren Details zu analysieren, weshalb dies geschah.

4.1.3 Details

Für einen genauen Vergleich der Glyphen zweier Dateninstanzen soll eine Detailansicht angezeigt werden, welche es ermöglicht beide Glyphen nebeneinander zu setzen. Zusätzlich soll in der Detailansicht eine Glyphe erzeugt werden, die wie in 4.2.3 beschrieben, die Differenz der beiden anderen Glyphen angibt. Anhand dieser Differenzglyphe kann der Unterschied der beiden Glyphen besser untersucht werden.

Die folgenden Details und Metadaten sollen durch einen interaktiv ausgelößten Event im Zusammenhang mit einem einzelnen Datenelement angezeigt werden.

- Nummerierung des zugehörigen Datensatzes
- Nummerierung der zugeordneten Klasse
- Nummerierung der Position innerhalb des Datensatzes
- Bezeichnung der Attribute

- Datenwerte der einzelnen Attribute
- Maximal- und Minimalwerte der Attribute
- Bezeichnung der Attribute

4.2 Anforderungen an das Glyphdesign

In diesem Abschnitt wird die Auswahl der verwendeten Glyphen beschrieben und auf die Konzeptionierung dieser eingegangen. Es wird eine Glyphe benötigt, die mehrere Attribute darstellen kann und eine überdeckungsfreie Anordnung ermöglicht. Bei Variation der Anzahl der Attribute soll die Fläche, die die Glyphe beansprucht gleich bleiben, es sei denn es wird durch den Benutzer eine Zoomfunktion verwendet.

4.2.1 Sternglyphe

Die Auswahl fiel auf die Sternglyphe, da sie eine Glyphe ist, die sich für abstrakte Daten eignet. Auf Grund der radialen Anordnung der Attribute nimmt sie immer die gleiche Gesamtfläche ein und ermöglicht eine gleichmäßige Anordnung. Dadurch ist es auch möglich viele Variablen auf kleinem Raum unterzubringen. Desweiteren ist es einfach einzelne Glyphen über die sich ergebende Sternenform miteinander zu vergleichen. Beim Überfliegen der Glyphenansicht, können Ausreißer erkannt werden, da diese durch die abweichende Form ins Auge stechen.

4.2.2 Durchschnittsglyphe

Zur Veranschaulichung der durchnittlichen Glyphform einer Klasse, soll eine repräsentative Glyphe berechnet werden und einen Überblick über übliche Aussehen der Glyphen innerhalb der Klasse geben.

Für die Berechnung der Durchnittsglyphe gelten die folgenden Annahmen.

- K sei die Menge der Klassen.
- $k_j \in K$ sei die j-te Klasse.
- G sei die Menge der geladenen Glyphen.
- $G_{k_j} \subseteq G$ sei die Menge der j-ten Glyphe.
- $g_l \in G_{k_j}$ sei die l-te Glyphe, innerhalb der j-ten Klasse.
- A sei die Menge der Attribute.
- $a_{i,g_l,k_j} \in A$ sei das i-te Attribut der l-ten Glyphe, innerhalb der j-ten Klasse.

- n_{k_j} sei die Anzahl der Glyphen in der j-ten Klasse.
- sum_{i,k_j} sei die Summe aller i-ten Attribute aus der j-ten Klasse.
- avg_{i,k_j} sei der Durchschnittswert des i-ten Attributs, der j-ten Klasse.

Mit den vorrangegangenen Annahmen kann für jedes Attribut die Summe innerhalb einer Klasse berechnet werden.

$$sum_{i,k_j} = \sum_{j=0}^{n_{k_j}} a_{i,g_l,k_j}$$

Anschließend kann, durch das Teilen durch die Anzahl der Glyphen innerhalb einer Klasse, der Durchschnittswert für jedes Attribut berechnet werden.

$$avg_{i,k_j} = \frac{sum_{i,k_j}}{n_{k,j}}$$

Mit den neu berechneten Durchschnittswerten kann die Glyphe wie jede andere Glyphe gezeichnet werden. Sie soll allerdings farblich oder durch eine andere Abgrenzung gegenüber den Datenglyphen hervorgehoben werden.

4.2.3 Differenzglyphe

Für die Detaildarstellung der Glyphen soll eine zusätzliche Differenzglyphe berechnet werden, welche den Unterschied zweier Glyphen abbilden soll.

Für die Berechnung der Differenz zwischen den beiden Glyphen werden die folgenden Annahmen getroffen.

- G sei die Menge der geladenen Glyphen.
- $g_{l_1} \in G$ sei die l_1-te Glyphe, die als erstes gewählt wurde.
- $g_{l_2} \in G$ sei die l_2-te Glyphe, die als zweites gewählt wurde.
- A sei die Menge der Attribute.
- $a_{i,g_{l_x}} \in A$ sei das i-te Attribut der l_x-ten Glyphe.
- $diff_{i,g_{l_1},g_{l_2}}$ sei die Differenz, des i-ten Attributs, zwischen der l_1-ten und l_2-ten Glyphe.

Mit diesen Annahmen kann durch die folgende Formel für jedes Attribut die Differenz zwischen den Werten der beiden Glyphen gebildet werden.

$$diff_{i,g_{l_1},g_{l_2}} = a_{i,g_{l_1}} - a_{i,g_{l_2}}$$

Da sich bei einer Subtraktion immer sowohl positive als auch negative Werte ergeben können, muss die repräsentative Glyphe diesen Wertebereich passend darstellen. Dazu soll die Achsmitte aller Attribute den Nullwert abbilden. Die positiven Werte sollen nach außen und die negativen Werte nach innen aufgetragen werden.

4.2.4 Normalisierung

Ein Problem, das bei der Visualisierung durch Glyphen auftritt, ist die Frage der Normalisierung der Datenwerte. Dabei ist zu beachten, ob die Wertebereiche der Attribute abhängig oder unabhängig voneienader sind. Um eine möglichst generische Lösung zu erhalten, wurden verschiedene Normalisierungsvarianten ausgewählt.

Die Normalisierungsfunktion sei dabei $f(x) = m * x + b$. Der minimale Wert des Definitionsbereichs sei min und der maximale Wert sei max. Für die die verschiedenen Möglichkeiten zur Normalisierung werden die Werte für m und b unterschiedlich berechnet.

Es gibt acht verschiedene Varianten zur Auswahl:

- Jedes Attribut wird einzeln mit seinem eigenen Definitionsbereich normalisiert.

  1. Als Erstes wird Definitionsbereich über jede Klasse einzeln mit $f(x)$ auf den Wertebereich $\{0...1\}$ normalisieren.

     Dafür wird als Zwischenwert $\mathit{diff_1} = max - min$ berechnet. Als Nächstes werden $m_1 = \frac{1.0}{\mathit{diff}}$ und $b_1 = -1.0 * (m * min)$ berechnet. Damit lassen sich mit $f(x)$ die normalisierten Werte für alle x der Daten berechnen.

  2. Der Definitionsbereich wird über jede Klasse einzeln mit $f(x)$ auf $\{0...1\}$ normalisieren, wobei die 0.5 des Wertebereichs auf der Mitte zwischen max und max liegt. Dabei wird die 0 des Definitionsbereichs auf die 0.5 des Wertebereichs abgebildet.

     Jetzt wird der Zwischenwert mit einer Fallunterscheidung,
     $\mathit{diff_2} = \begin{cases} |max| & \text{falls } |max| < |min| \\ |min| & \text{sonst} \end{cases}$, berechnet.
     $m_2 = \frac{0.5}{\mathit{diff}}$ und $b_2 = 0.5$ werden als Nächstes bestimmt und damit kann für alle Datenwerte der entsprechende normalisierte Wert berechnet werden.

  3. Als Drittes wird der Definitionsbereich über den kompletten Datensatz festgelegt und mit $f(x)$ auf den Wertebereich $\{0...1\}$ normalisiert.

     Die Berechnung der Funktionsparameter m und b wird dabei wie bei 1. vorgenommen.

  4. Der Definitionsbereich wird wieder über den ganzen Datensatz mit $f(x)$ auf $\{0...1\}$ normalisiert, wobei wieder die 0.5 des Wertebereichs in der Mitte zwischen max und max liegt und die 0 auf die 0.5 abgebildet wird.

     Die Berechnung der Funktionsparameter m und b wird dabei wie bei 2. vorgenommen.

- Für alle Attribute wird ein gemeinsamer Definitionsbereich verwendet.

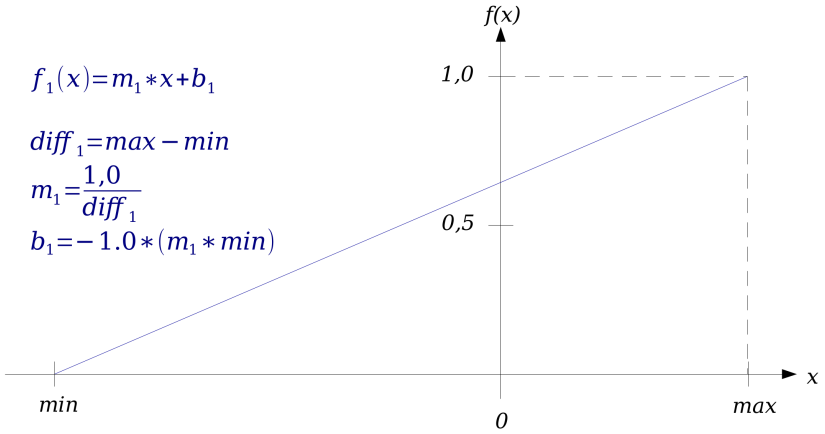

Abbildung 4.2: Einfache Normalisierungsfunktion, um vom Definitionsbereich der Attribute auf $\{0...1\}$ abzubilden.

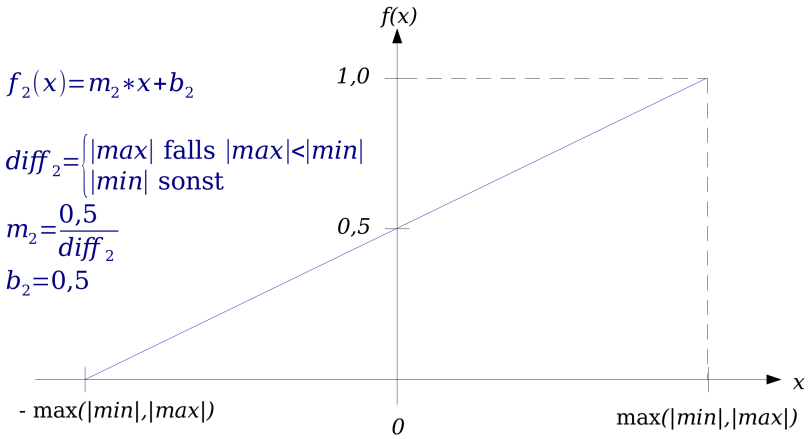

Abbildung 4.3: Normalisierungsfunktion um vom Definitionsbereich der Attribute auf $\{0...1\}$, mit 0.5 in der Mitte des Wertebereichs, abzubilden, wobei die 0 aus dem Definitionsbereich auf die 0.5 abgebildet wird.

5.-8. Die Normalisierungen 5. bis 8. werden jeweils wie bei 1. bis 4. nur mit gemeinsamen Definitionsbereich über alle Attribute berechnet.

4.3 Anforderungen an das maschinelle Lernen

Die maschinellen Lernverfahren müssen nicht implementiert werden, denn dafür gibt es bereits ausgereifte Bibliotheken. Für diese Arbeit soll das Weka (Waikato Environment for Knowledge Analysis) Framework [HFH+09] verwendet werden. Das Framework kann als Programmbibliothek verwendet werden und die Funktionen können in das eigene Programm

integriert werden. Ein Bestandteil davon ist der vorhandene Dateileser, der aus den Dateien der Datensätze die Werte in eine Datenstruktur des Frameworks lädt. Am Beseten sollen dafür Dateien im Attribute-Relation File Format (ARFF) verwendet werden. Dieses Dateiformat ist für die Verwendung mit maschinellen Lernverfahren ein gebräuchliches Standartformat.

Mit geladenen Daten die bereits klassifiziert wurden, sollen verschiedene Klassifikatormodelle trainiert werden. Dafür soll es eine Eingabemaske geben, die es ermöglicht, verschiedene Algorithmen mit unterschiedlichen Parametereinstellungen aus der Weka Bibliothek auszuwählen. Mit dem trainierten Modell sollen anschließend Testdaten in die Klassen eingeordnet werden.

5 Implementierung

In diesem Kapitel wird auf das erarbeitete Visualisierungskonzept eingegangen. Dabei wird zunächst in Abschnitt 5.1 auf den grundsätzlichen Aufbau der gesamten Benutzeroberfläche eingegangen. Darauf folgend werden die verschiedenen Einzelansichten, sowie deren Funktion und Interaktionsmöglichkeiten beschreiben.

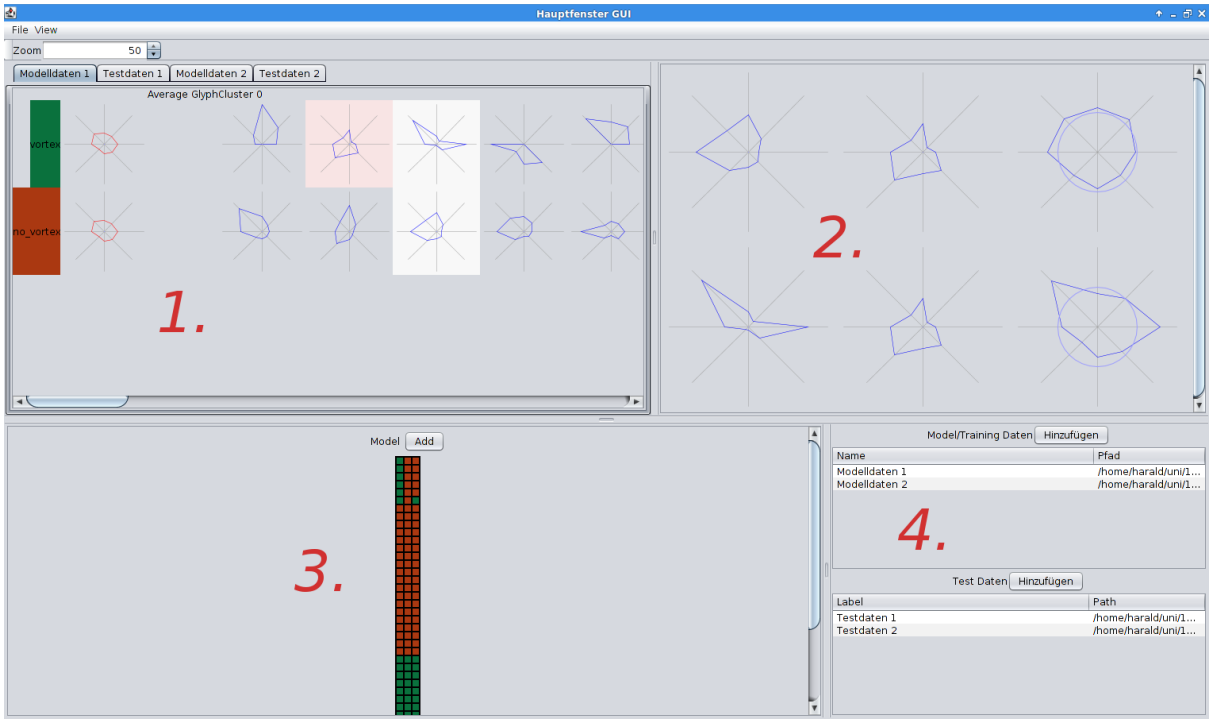

Abbildung 5.1: Das ist der Programm Prototyp zur glyphbasierten Visualisierung und Modellanalyse. Die Nummerierung der vier Ansichten, entsprech derer in Abschnit 5.1. Das sind 1. die Glyphübersicht, 2. die Detailansicht, 3. die Modelvergleichsübersicht und 4. die Datensatzübersicht.

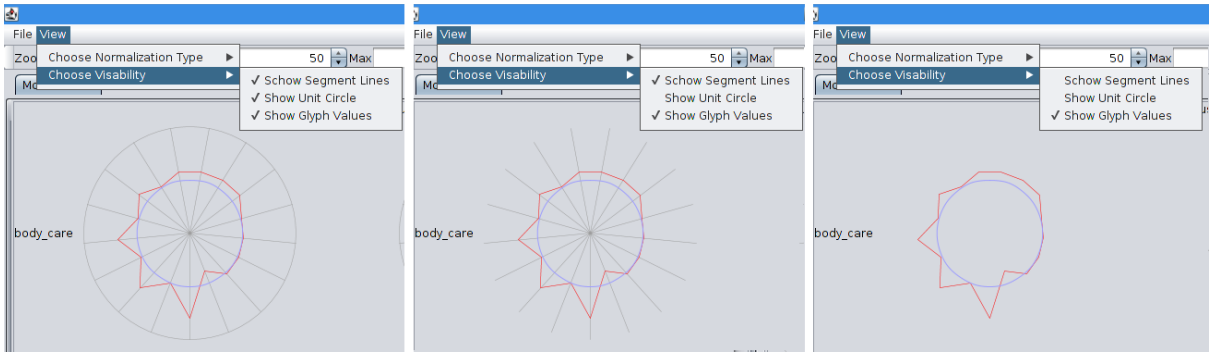

Abbildung 5.2: In dieser Abbildung ist das Menü zur Auswahl der Glyphkomponenten abgebildet. Dabei sind drei verschiedene Varianten zu sehen, um den Unterschied zu veranschaulichen.

5.1 Benutzeroberfläche und Interaktion

Die Benuteroberfläche des Programms besteht wie ein übliches Programm aus einer Menüleiste, sowie einer Werkzeugleiste, um allgemeine Einstellungen vorzunehmen. Desweiteren teilt sich das Programmfenster in vier geteilte Unteransichten ein. Für die vier Ansichten wird das Fenster einmal horizontal, und einmal vertikal geteilt, so dass vier Teilansichten darin platz finden. Die Konzeption der Ansichten und Interaktionsmöglichkeiten die in das Programm implementiert wurden, werden in Kapitel 4.1 beschrieben.

1. Die erste Ansicht oben links ist zur Übersicht aller Daten eines Datensatzes durch ihre Glyphrepräsentation. In 5.1.2 wird beschrieben was diese Ansicht für Möglichkeiten bietet.
2. Die zweite Ansicht oben rechts ist eine Detailansicht, in der sich ausgewählte Glyphen zu einer weiteren Glyphe vergleichen lassen. In 5.1.3 ist eine nähere Beschreibung dieser Ansicht zu finden.
3. Die dritte Ansicht unten links zeigt eine Übersicht zu den ausgewählten Lernalgorithmen, um diese vergleichen zu können.
4. Die vierte Ansicht listet die geladenen Datensätze mit dem Dateipfad auf.

5.1.1 Menü- und Werkzeugleiste

In der Menüleiste gibt es die Möglichkeit, die Glyphenansicht anzupassen. Dazu zählen zum Einen das Ein- und Ausblenden verschiedener Glyphkomponenten, wie die Achsen der Attribute und den Einheitskreis, der mit dem Maximalwert die Glyphe umrandet. In Abbildung 5.2 sind drei verschiedene Einbledungen dargestellt, einmal mit allen Komponenten, dann ohne den Einheitskreis und zuletzt auch ohne die Achsen der Attribute.

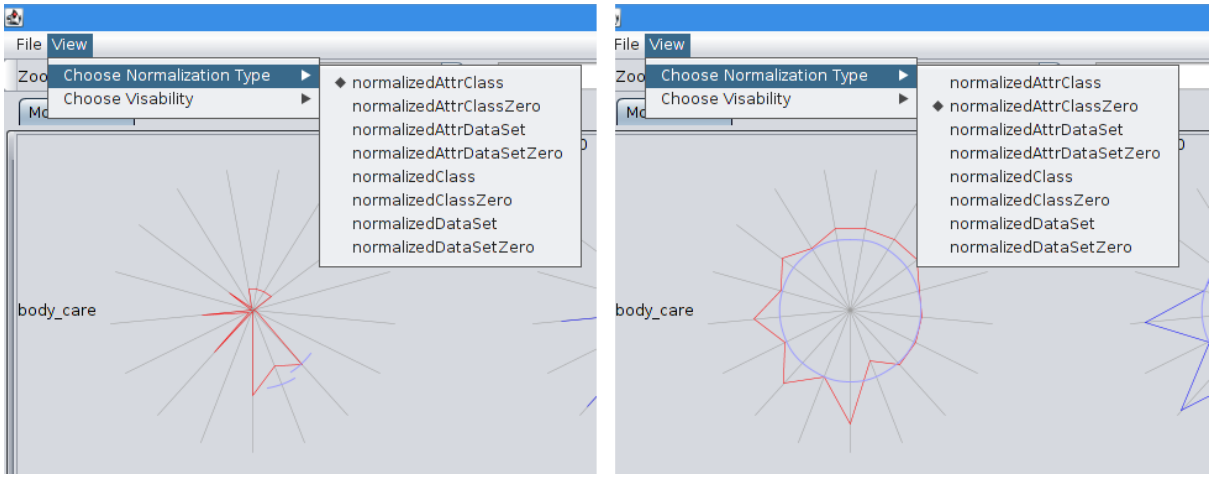

Abbildung 5.3: Hier ist das Auswahlmenü für die verschiedenen Normalisierungen der Glyphen zu sehen. Es sind zwei verschiedene Auswahlen abgebildet, um zu sehen wie sich die Glyphen anpassen.

Desweiteren ist es über das Ansichtenmenü möglich, verschiedene Varianten der Normalisierung auszuwählen, wie es in 4.1.2 im Abschnitt zur Normalisierungsauswahl beschrieben wurde. In Abbildung 5.3 sieht man zwei Beispiele, wie die Auswahl über das Menü die Normalisierung in den Glyphen verändert. Die verschiedenen Normalisierungsfunktionen wurden in Abschnitt 4.2.4 in den Anforderungen konzipiert und so in das Programm implementiert. In der Abbildung 4.1 kann man ein Beispiel sehen, bei welchem eine Glyphe mit den verschiedenen Varianten normalisiert wurde.

In der Werkzeugleiste kann man über ein Eingabefeld die Größe der Glyphen in der Übersicht verändern.

5.1.2 Glyphübersicht

In der Glyphübersicht werden alle Dateninstanzen eines Datensatzes durch ihre entsprechende Glyphrepräsentation dargestellt. Die Anforderungen an diese Ansicht wurden in Abschnitt 4.1.1 erarbeitet und werden hier auf die Implementierung bezogen.

Dabei wird in der Ansicht für jede Klasse eine neue Zeile begonnen. Innerhalb einer Zeile werden die Glyphen nochmals zusätzlich durch einen Clusteralgorithmus sortiert. Dafür wird in der Implementierung das Clustering mit k-Means aus der Klasse „weka.clusterers.SimpleKMeans“ der Weka Bilbiothek [HFH+09] verwendet. Am Anfang jeder Zeile wird, wie in Abbildung 5.4 abgebildet, eine Glyphe zur Repräsentation der Klasse abgebildet. Diese Glyphe soll den Eindruck vermitteln, wie eine Glyphe, die der Klasse zugehöhrt, im Idealfall aussehen soll. Wie die repräsentative Glyphe berechnet wird, wurde in den Anforderungen in Abschnitt 4.2.2 erklärt.

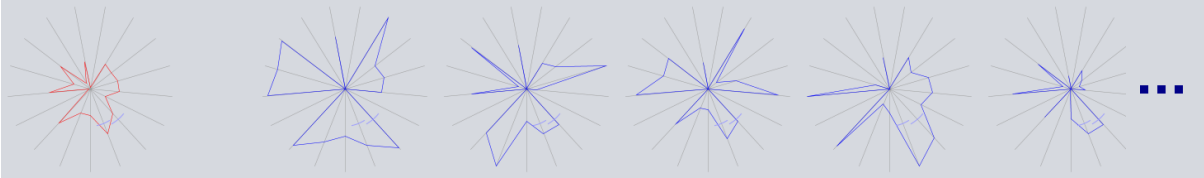

Abbildung 5.4: Ganz links ist die repräsentative Durchschnittsglyphe zu sehen welche aus dem Durchschnitt aller rechts daneben liegenden Glyphen gebildet wird. Es sind nicht alle Glyphen rechts zu sehen, da es sehr viele sein können.

In Abbildung 5.5 ist eine niedrige Zoomstufe ausgewählt, welche es ermöglicht sich einen Überblick durch alle Glyhen des Datensatzes zu verschaffen. In dieser Ansicht können schon mögliche Ausreißer anhand abweichender Glyphgformen erkannt werden. Um feinere Unterschiede erkennen zu können, ist es gegebenenfalls erforderlich in die Ansicht hineinzuzoomen. Wie die herangezoomte Ansicht aussieht ist in Abbildung 5.6 zu sehen.

Glyphdarstellung

Hier wird kurz auf die Umsetzung des Glyphdesigens, das in Abschitt 4.2 konzipiert wurde, eingegangen, wie die Sternglyphe ins Programm integriert wurde.

Die Werte der Glyphe werden alle auf das Intervall $[0, 1]$ abgebildet und anschließend über eine Zeichenmethode gezeichnet. Dabei wird jedes Attribut durch einen Vektor in die entsprechende Achsrichtung interpretiert. Unter dieser Bedingung wird für die Zoomfuktion ein Multiplikator in die Zeichenmethode integriert, der die Vektoren mit dem Zoomfaktor multipliziert und die Glyphe damit vergrößert. Es ist zu beachten, dass der erste Wert unten in der Mitte der Glyphe beginnt und gegen den Uhrzeigersinn in equidistanten Abständen auf die Glyphe aufgetragen werden.

Je nach Normaisierung werden die Glyphen unerschiedlich dargestellt. Dabei wird bei den Varianten, bei welchen sich die Null immer in der Mitte der Achsen befindet, ein Kreis zur Markierung der Null eingezeichnet. Damit kann leicht erkannt werden, welche Werte positiv und welche negativ sind. Werden die Werte dagegen nur auf den Wertebereich abgebildet, in dem sich die Werte eines einzelnen Attributs befinden, kann es dazu führen, dass nur auf manchen Achsen auch negative Werte vorhanden sind. Nur wenn positive und negative Werte bei einem Attribut vorhanden sind, werden die Nullstellen auf den Achsen durch einen Teilkreis markiert. Wie die verschiedenen Normalisierungen berechnet werden, wurde in Abschnitt 4.2.4 abgehandelt.

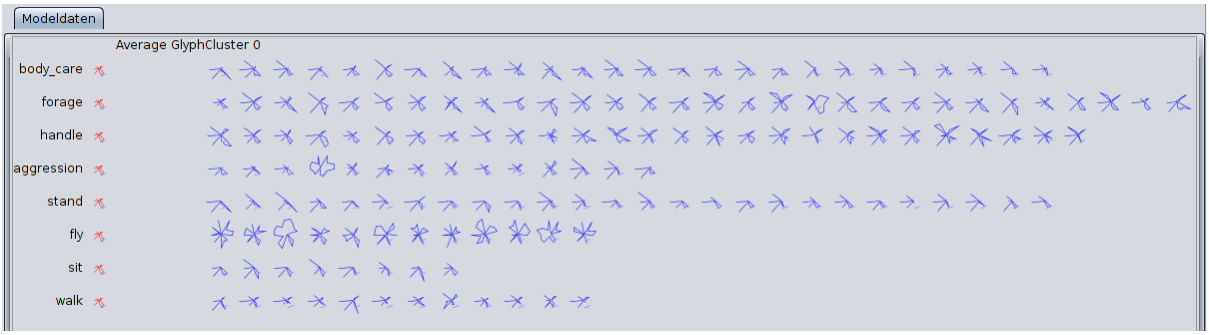

Abbildung 5.5: In dieser Abbildung ist die Glyphübersicht in einer niedrigen Zoomstufe, für einen größeren Überblick, zu sehen.

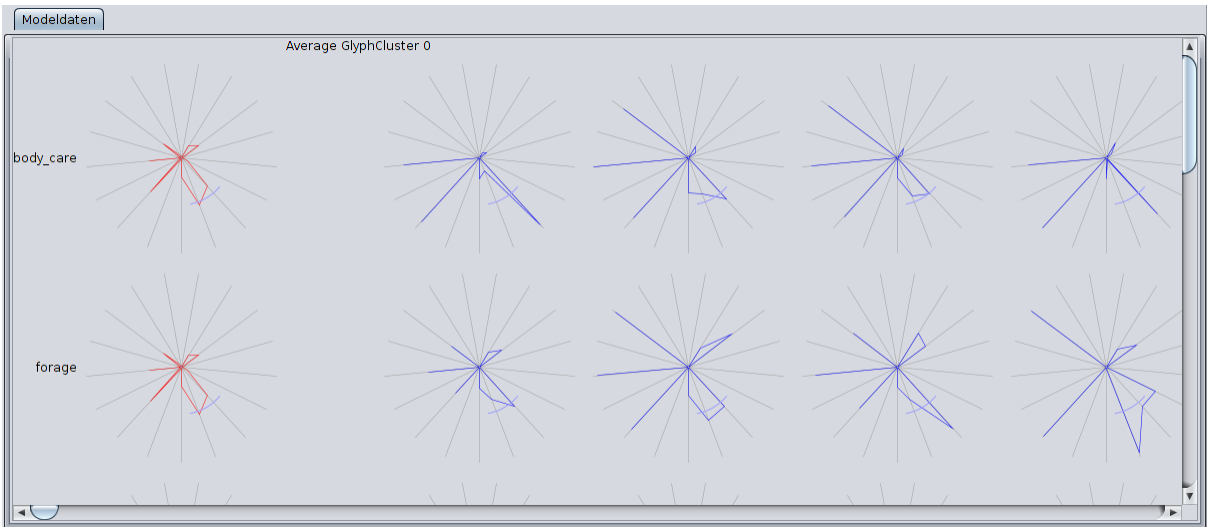

Abbildung 5.6: Hier ist die Glyphansicht in einer vergrößerten Zoomstufe zu sehen. Das bietet die Möglichkeit einen kleineren Ausschnitt genauer anzusehen.

5.1.3 Detailansicht mit Differenzglyphe

In diesem Abschnitt wird die in den Anforderungen zu den Details 4.1.3 geforderte Detailansicht, wie sie implementiert wurde, beschrieben.

Die in der Detailansicht angezeigten Glyphen werden zuvor in der Glyphansicht ausgewählt. In Abbildung 5.7 ist links in einem Auschnitt der Übersicht eine Auswahl der Glyphen zu sehen. Dabei sind die hellgrau hinterlegten Glyphen eine Auswahl an Minuenden, von welchen die rot hinterlegte Glyphe als Subtrahend abgezogen wird. Rechts in Bild 5.7 ist die Detailansicht zu sehen, in welcher die Minuenden in der linke der drei Spalte dargestellt werden. Der Suptrahend steht ist in der mittleren Spalte und wird von den Glyphen links davon abgezogen. In der rechten Spalte werden die daraus berechneten Differenzglyphe abgebildet. Die Berechnung dafür wurde in Abschnitt 4.2.3, im Kapitel zu den Anforderungen an das Programm, beschrieben.

Wird eine Glyphe mit der Maus fokussiert, so wird ein Tooltip eingeblendet, der die in den Anforderungen 4.1.3 gewünschten Daten enthält. In der Abbildung 5.7 ist der Tooltip einer Glyphinstanz, im gelben Kasten, abgebildet. Dies beinhaltet einige Metadaten und Datenwerte zu der entsprechenden Glyphe, die im Folgenden in der Reihenfolge, wie sie auch im Tooltip stehen, aufgelistet werden.

1. Eine Indentifikationsnummer aus welchem Datensatz die Daten stammen.
2. Eine Indexnummer an welcher Stelle im Datensatz sich die Instanz befindet. Dies entspricht auch der Zeile in der geladenen Datei.
3. Zur zugehöhrige Klasse, der Instanz eine Indentifikationsnummer.
4. Eine Indexnummer, an welcher Stelle sich die Daten innerhalb der Klasse befindet.
5. Eine Liste mit den Werten der einzelnen Attribute.
6. Eine Liste mit den Minimalwerten der Attribute, je nach Normalisierung über alle Attribute gleich oder für jedes Atribut einzeln.
7. Die Maximalwerte der Attribute in einer Liste, wie davor von der Normalisierung abhängig.
8. Für alle Attribute eine Liste mit den Durchnittswerten, die über die Klasse hinweg gebildet wurden.
9. Die Anzahl der Klassen im zugehörigen Datensatz.
10. Anzahl der Instanzen in Klasse, in welcher sich die Glyphinstanz befindet.
11. Die Anzahl der Attribute die ein Dateninstanz beinhaltet.
12. Anzahl aller Instanzen die im selben Datensatz entahlten sind.

5.1.4 Modelvergleichsübersicht

In der Modellvergleichsübersicht werden jeweils zu den ausgewählten Algorithmen, eine farbkodierte Spalte mit den einzelnen Dateninstanzen angezeigt. Die Instanzen sind dabei über die Zeile hinweg gleich, um direkt zu sehen, welche Modelle die gleiche Instanz in verschiedene Klassen eingeordnet haben. Wie in den Anforderungen zum maschinellen Lernen, in Abschnitt 4.3, erwähnt, werden die maschinellen Lernverfahren aus dem Weka Framework [HFH+09], der Universität von Waikato, benutzt. In Abbilddung 5.8 wurden zur Veranschaulichung ein Datensatz mit drei verschieden Algorithmen aus dem Weka Framework klassifiziert. Dabei wurden von von links nach rechts die folgenden Lernverfahren verwendet, deren grundlegenden Funktionsweisen in Abschnitt 2.2 beschrieben sind.

1. Ein Algorithmus mit Suport Vector Machine, der in der Weka Klasse „weka.classifiers.functions.SMO“ zu finden ist.

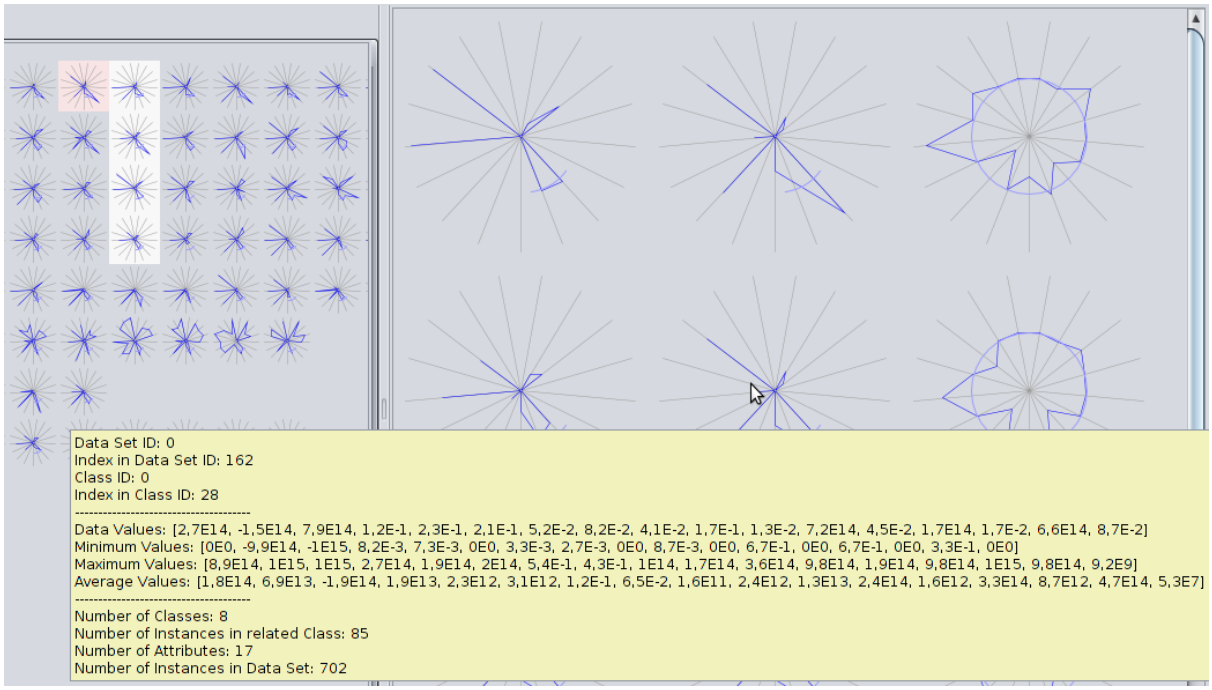

Abbildung 5.7: Hier ist die Detailansicht, mit einem Ausschnitt der Glyphübersicht links daneben abgebildet. In dem Übersichtsausschnitt sind die Glyphen farblich hervorgehoben, die für die Detailansicht ausgewält wurden. In der Detailansicht werden die markierten Glyphen und die dazu berechneten Differenzglyphe angezeigt. Zusätzlich ist noch der Tooltip mit weiteren Metadaten und Werten, von der mit der Maus fokussierten Glyphe, zu sehen.

2. Als Nächtes ein Algorithmus der einen Entscheidungsbaum verwendet, der in Weka als „weka.classifiers.trees.J48“ zu finden ist.
3. Zuletzt noch ein Algorithmus der Neuronale Netze verwendet, und in Weka unter „weka.classifiers.functions.MultilayerPerceptron“ zu finden ist.

Es ist zu sehen, dass bei dem Beispiel in der Abbildung der erste Algoritmus die ersten sechs Instanzen, gegenüber dem Zweiten, anders eingeordnet hat. Gegenüber dem Dritte unterscheiden sich die Klassen allerdings nur bei den ersten fünf Daten. Zwischen dem ersten und dritten Modell unterscheidet sich die Klasseneinordnung allerdings nur bei der sechsten Instanz. Unter zuhilfenahme der Glyphen kann damit untersucht werden, worin sich die ersten sechs Dateninstanzen von den aderen unterscheiden und unterschiedlich eingeordnet wurden.

In der Abbildung 5.9 ist die Eingabemaske zum Eingeben der Algorithmen mit Komandozeilen Parametern als Optionen, wie es auch beim Weka Framework direkt möglich ist. Experten, die die Algorithen und Parameter kennen können diese direkt eingeben. Andernfalls können auch einige Algorithmen und die Parametrer im Manual des Fraemeworks [BFH+16] nachgeschlagen oder über die Grafikschnittstelle des Weka Frameworks herauskopiert werden.

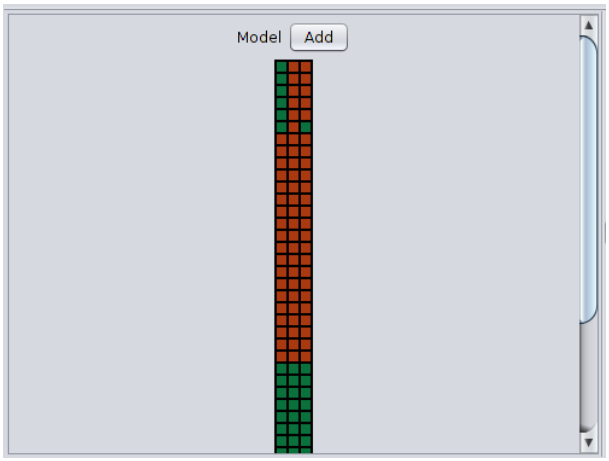

Abbildung 5.8: Die Abbildung zeigt die Modellvergleichsansicht, die spaltenweise die durch verschiedene Lernmodelle klassifizierten Testdaten anzeigt. Innerhalb einer Zeile befinden sich immer die gleichen Dateninstanzen.

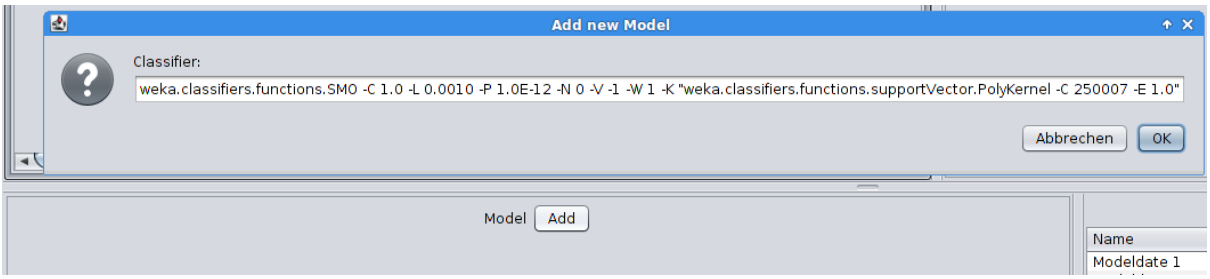

Abbildung 5.9: Hier ist die Eingabemaske für die Auswahl eines Lernalgorithmus zu sehen, welche über den Add-Button in der Modellansicht aufgerufen wird.

5.1.5 Datensatzübersicht

In der Datensatzübersicht werden die geladenen Datensätze mit einem gewählten Namen und dem Dateipfad angezeigt. Die Ansicht ist dabei in eine Liste für Trainingsdaten und eine für Testdaten aufgeteilt, wie man in der Abbildung 5.10 sehen kann. Um eine neue Datei einzulesen, kann entweder bei den Trainingsdaten oder den Testdaten über den Hinzufügen Button ein entsprechender Eingabedialog aufgerufen werden, der ebenfalls in Abbildung 5.10 abgebildet ist. In dem Dialog muss der Pfad zur Datei angeben werden. Dies kann direkt durch die Eingabe in das dafür vorgesehen Textfeld geschehen oder es kann ein Datei-Browser zur Auswahl aufgerufen werden. Es muss noch die Spalte, in der die Klassenlabels stehen, spezifiziert werden. Bei ARFF Datein ist das standardmäßig die letzte. Die Testdaten müssen ebenfalls eine Spalte für die Klassen enthalten und im Kopf der Datei müssen die Klassennamen spezifiziert sein, um die Testdaten durch die Algorithmen klassifizieren zu können.

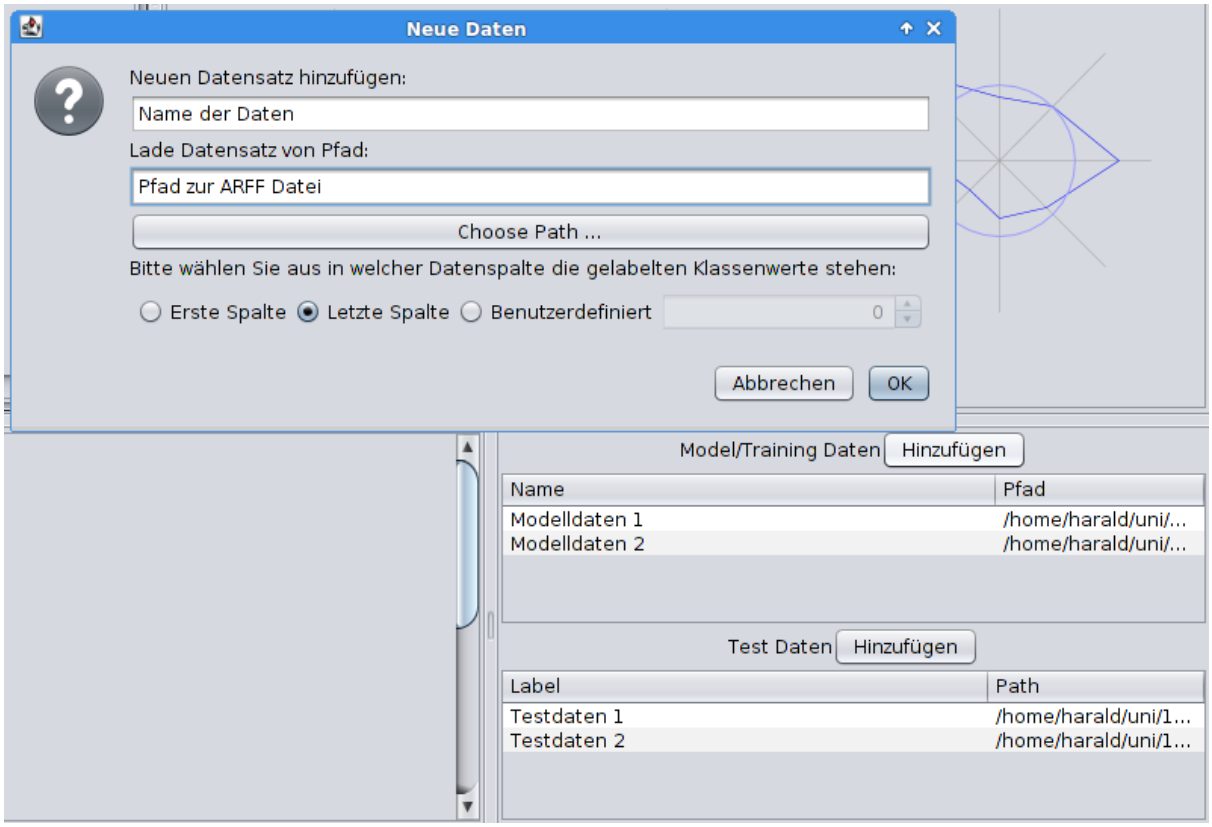

Abbildung 5.10: Diese Abbildung Zeigt die Datensatzübersicht, in der die geladenen Datensätze aufgelistet sind. Desweiteren ist der Eingabedialog zum Laden der Daten abgebildet.

6 Mögliche Arbeitsabläufe

In diesem Kapitel werden mögliche Arbeitsabläufe für einige Anwendungsfälle des Visualisierungstools beschrieben. Zuerst wird beschrieben, wie das Tool zur Analyse von Trainingsdaten verwendet werden kann, um eine bessere Auswahl eines Klassifikators zu treffen. Danach wird ein Szenario beschrieben, bei welchem die Klassifizierung von Testdaten durch mehrere verschiedene Modelle verglichen wird und dabei Eigenheiten der Testdaten analysiert werden können.

6.1 Analyse klassifizierter Daten

Es werden klassifizierte Daten mit dem Programm geladen, und können mit diesem analysiert werden.

Ziel

Ziel ist das Analysieren vorklassifizierter Daten, um festzustellen ob diese schon Eigenheiten enthalten, die Probleme beim Trainieren des Lernmodells verursachen können. Experten können damit im Vorfeld die Auswahl von passenden Klassifikatoren einschätzen.

Ausführungsschritte

1. Klassifizierte Daten in die Übersicht mit ihren Glyphrepräsentationen laden.
2. Die Glyphen auswählen, die unterschiedlichen Klassen zugeordnet sind, aber in der Sortierung im selben Cluster liegen. Dabei die Instanzen wählen, die in der Klasse liegen, die am wenigsten Instanzen im Cluster hat.
3. Die repräsentative Glyphe, die die meisten Elemente innerhalb der Clustergruppe hat, als Subtrahend auswählen.
4. Anhand der erzeugten Differenzglyphe untersuchen, auf Grund welcher Attribute die gewählten Datenelemente nicht in die gewählte Klasse eingeordnet wurden und wie groß deren Abweichungen sind.

5. Ein Experte kann damit überprüfen, ob die entsprechenden Instanzen richtig klassifiziert sind und einschätzen wie wichtig die Attribute für die Klassifizierung sind. Dies kann ihm Aufschlüsse über die Auswahl eines passenden Klassifikatormodells geben.

6.2 Vergleich von Klassifikatoren

Es werden Datenelemente in Testdaten untersucht, welche durch verschiedene Klassifikationsalgorithmen in unterschiedliche Klassen eingeordnet wurden.

Ziel

Anhand der Glyphen soll herausgefunden werden, welche Attribute für unterschiedliche Klassenzuordnungen des gleichen Elements verantwortlich sein können. Damit sollen Experten unterstützt werden, herauszufinden, ob Attribute für die Einordnung relevant sind. Andernfalls können die Attribute für bestimmte Klassifizierer herausgenommen, um ein bessere Zuordnung zu erreichen.

Ausführungsschritte

1. Klassifizierte Daten laden, die zum Trainieren der Klassifikatormodelle verwendet werden.
2. Testdaten ohne Klassenzuordnung laden, auf welche die trainierten Klassifikatoren angewendet werden.
3. Zu vergleichende Klassifikationsalgorithmen in der Modellvergleichsübersicht hizufügen.
4. In der nach Klassen farbkodierten Tabelle nach Elementen suchen, bei welchen innerhalb einer Zeile unterschiedliche Farbwerte angezeigt werden.
5. Datenelement auswählen und die zugehörige Glyphe mit denen der Klassen, in die sie eingeordnet wurden, vergleichen.
6. Die repräsentativen Glyphen der zugeordneten Klassen für die Datailansicht auswählen.
7. Die zu analysierende Glyphe als Subtrahend für die Detailansicht markieren.
8. Aus der Differenzglyphe die berechnet wird ablesen, welche Attribute untypische Werte für die eingeordneten Klassen haben.

9. Ein Experte kann anhand dieser Informationen überprüfen, ob die entsprechenden Attribute für die Klassifizierung relevant sind und kann diese gegebenenfalls herausfiltern und den Klassifikator diesbezüglich anpassen.

7 Zusammenfassung und Ausblick

In diesem Kapitel werden zunächst die wichtigsten Punkte, die in dieser Arbeit behandelt wurden, zusammengefasst. Im Anschluss daran wird im Ausblick auf einige Ideen eingegangen, welche Möglichkeiten die Arbeit zur Erweiterung der Konzepte bietet.

Zusammenfassung

Zu Begin wurden in Kapitel 2 die Grundlagen der Visualisierung und des maschinellen Lernens behandelt. Dabei wurde im Speziellen auf die Informationsvisualisierung zur Darstellung multivariater Daten eingegangen. Es wurden einige Möglichkeiten dazu vorgestellt, wovon im späteren Verlauf der Arbeit die Sternglyphe als Visualisierungsform ausgewählt wurde. Es wurden auch grundlegenden Interaktionskonzepte beschrieben. Danach wurden noch die Funktion einiger maschineller Lernalgorithmen beschrieben, die später ebenfalls verwendet wurden.

In Kapitel 3 wurden Bezug auf andere Arbeiten genommen, die sich zum Einen mit der visuellen Analyse von Klassifikatoren und Modelldaten beschäftigten und zum Anderen mit der Visualisierung multivariater Daten.

Die Anforderungen an die Visualisierung und Interaktion, zur Analyse der maschinellen Lernverfahren, wurden in Kapitel 4 festgelegt. Dabei wurde ein Konzept zusammengestellt, welches zunächst ein Grundgerüst mit verschiedene Ansichten für die Analyse beinhaltet. Danach wurden die einzelnen Ansichten spezifiziert. Dabei wurde eine Übersichtsansicht durch die Visualisierung mit Sternglyphen beschrieben, die durch Interaktionen mit den Glyphen eine Detailansicht ermöglicht. Es wurde auch das Glyphdesign beschrieben, das verwendet wurde. Dabei wurde die Darstellung als Sternglyphefestgelegt und verschiedene Möglichkeiten zur Normalisierung der Datenwerte beschrieben. Zusätzlich wurden noch eine Durchschnittsglyphe und eine Differenzglyphe für die Detailanalyse spezifiziert. Es wurde auch noch eine Übersicht beschrieben, die durch eine einfache farbkodierte Matrix den Vergleich verschiedener Lernmodelle ermöglicht.

Im Anschluss wurden in Kapitel 5 die Anforderungen in die Implementierung umgesetzt. Das beginnt mit der typischen Benutzeroberfläche mit Menü- und Werkzeugleiste. Diese enthalten die Interaktionskonzepte zum Anpassen der Darstellungen innerhalb der Visualisierungsansichten, wie sie in der Anforderungsanalyse festgelegt wurden. Danach wird gezeigt wie die

Glyphübersicht mit den zusätzlichen Durchschnitsglyphen für jede Klasse, die Detailansicht mit den Differenzglyphen, die Modelvergleichsübersicht und die Datensatzübersicht nach den Angaben aus den Anforderungen implementiert wurden.

Zum Schluss wurden in Kapitel 6 noch Möglichkeiten beschrieben, wie das Programm angewendet werden kann. Dabei wird erklärt, wie die verschiedenen Ansichten und Interaktionen zur Analyse der Daten verwendet werden.

Ausblick

In diesem Abschnitt werden Ideen für die Erweiterung und Verbesserung der Konzepte beschrieben. Das gilt sowohl für das Glyphdesign, als auch für die Benutzeroberfläche und Interaktion, um die Analysemöglichkeiten zu verbessern.

Glyphdesign

An dieser Stelle werden ein paar Erweiterungsmöglichkeiten des Glyphdesigns beschrieben.

Es könnten noch die Varianz in die Attribute der Glyphen integriert werden. Dazu könnte sie entweder durch Markierungen an den Achsen abgebildet werden oder es könnten beispielsweise zusätzliche Glyphkonturen eingefügt werden. Dadurch kann direkt gesehen werden, wie eng beieinander die Werte eines Attributs liegen. Je größer die Varianz der Werte ist, desto größer ist die Wahrscheinlichkeit, dass die Achse keine Relevanz für die Klasseneinordnung hat.

Für die Detailansicht wäre es zusätzlich zur Differenzglyphe hilfreich, die Konturen der zwei zu vergleichenden Glypen in einer Glyphgrefik übereinander zu legen. Das Ganze kann auch noch erweitert werden, dass die Konturen beliebig vieler Glyphen übereinander gelegt werden. Daraus ist ähnlich wie bei der Varianz zu sehen, wie stark sich die Werte verteilen. Dies ermöglicht ebenfalls bessere Rückschlüsse auf die Relevanz der einzelnen Attribute. Werden dabei nah beieinander liegende Werte durch die Grafik immer dicker dargestellt, ist direkt zu sehen ob viele Werte nahe beieinander liegen. Gibt es nur wenige Werte, die weiter auseinander liegen, sind diese als leicht als Ausreißer zu erkennen.

Ein weiterer Ansatzpunkt wären zusätliche Möglichkeiten zur Normalisierung. Die Werte könnten beispielsweise auch durch nicht lineare Funktionen normalisiert werden. Würden die Attribute mit einem Gausfilter normalisiert, so könnten Ausreißer am Rand des Wertebereichs etwas abgeschwächt werden. Dafür würden aber in der Mitte des Wertebereichs die kleinere Unterschiede verstärkt werden. Dort befinden sich oft die meisten Werte, wodurch die Unterschiede in diesem wichtigen Bereich besser sichtbar werden.

Benutzeroberfläche und Interaktion

Hier werden ein paar Erweiterungen zur Verbesserung des Bedienoberflächen- und Interaktionskonzepts beschrieben.

Wenn die Relevanz der Attribute beurteilt werden kann, wäre es sinnvoll eine Möglichkeit zu bieten, dies an das Lernmodell weiter zu geben. Es könnte die Möglichkeit hinzugefügt werden, die einzelnen Attribute zu gewichten oder Attribute ganz von der Klassifizierung auszuschließen.

Wird bei der Analyse der Daten festgestellt, dass eine Instanz falsch klassifiziert ist, so wäre es praktisch, diese über eine Benutzerinteraktion ändern zu können. Über ein Kontextmeü könnte dem Benutzer eine Auswahl der Klassen zum Auswählen angeboten werden. In diesem Zusamenhang wäre es auch gleich sinnvoll, die Änderungen in einer neuen Datei auf der Festplatte abspeichern zu können.

Das Konzept der Detailansicht birgt ebenfalls noch Potenzial für Erweiterungen. Es wäre möglicherweise sinnvoll, für die ausgewählten Glyphen in der Detailansicht eine eigene Sortierung zu berechnen. Dazu könnte wieder ein Clustering verwendet werden, allerdings nur auf der Datenbasis der ausgewählten Glyphe. Dadurch werden Ähnlichkeiten unter Umständen besser sichtbar, da ähnliche Glyphen näher beieinander angezeigt werden. Eine weitere hilfreiche Ergänzung wäre das Zeichnen einer Durchschnittsglyph der ausgewählten Glyphen. Dies würde es ermöglichen, eine Repräsentation einer Teilmenge einer Klasse zu bekommen. Damit könnten beispielsweise nur Glyphen ohne Ausreißer, die die Durchschnittsglyphe verzerren würden, ausgewählt werden.

Eine weiteres praktisches Interaktionskonzept könnte die Auswahl verschiedener Clustering Algorithmen sein. Damit könnte eine, zum Datensatz passende, Sortierung innerhalb der Klassen ausgewählt werden.

Literaturverzeichnis

[AV07] D. Arthur, S. Vassilvitskii. „k-means++: The advantages of careful seeding“. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial und Applied Mathematics. 2007, S. 1027–1035 (zitiert auf S. 18, 19).

[Ave10] Avenue. *Chart showing Chernoff faces for data selected from the ÜSJudgeRatings"dataset in R, which contains ratings of state judges in the US Superior Court by lawyers who have had contact with them.* https://commons.wikimedia.org/wiki/File:Chernoff_faces_for_evaluations_of_US_judges.svg. 2010. URL: https://commons.wikimedia.org/wiki/File:Chernoff_faces_for_evaluations_of_US_judges.svg (zitiert auf S. 15).

[BFH+16] R. R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald, D. Scuse. „WEKA Manual for Version 3-9-0“. In: *Hamilton, New Zealand* (2016). URL: http://www.cs.waikato.ac.nz/ml/weka/documentation.html (zitiert auf S. 41).

[Che73] H. Chernoff. „The Use of Faces to Represent Points in k-Dimensional Space Graphically“. In: *Journal of the American Statistical Association* 68.342 (1973), S. 361–368. DOI: 10.1080/01621459.1973.10482434. eprint: http://www.tandfonline.com/doi/pdf/10.1080/01621459.1973.10482434. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1973.10482434 (zitiert auf S. 15).

[Coo08] D. Cook. *Screenshot of GGobi, showing a parallel coordinate plot.* https://commons.wikimedia.org/wiki/File:Ggobi-flea2.png. 2008. URL: https://commons.wikimedia.org/wiki/File:Ggobi-flea2.png (zitiert auf S. 14).

[Enn10] Ennepetaler86. *Zwei Vektor-Klassengebiete mit zwei möglichen Trenn-Geraden, wobei Gerade A das größere leere Randgebiet zwischen den Klassengebieten besitzt.* https://commons.wikimedia.org/wiki/File:Svm_intro.svg. 2010. URL: https://commons.wikimedia.org/wiki/File:Svm_intro.svg (zitiert auf S. 17).

[HFH+09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. „The WEKA data mining software“. In: *ACM SIGKDD Explorations Newsletter* 11.1 (Nov. 2009), S. 10. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278. URL: http://dx.doi.org/10.1145/1656274.1656278 (zitiert auf S. 17, 18, 32, 37, 40).

[HNH+12] B. Hoferlin, R. Netzel, M. Hoferlin, D. Weiskopf, G. Heidemann. „Inter-active learning of ad-hoc classifiers for video visual analytics“. In: *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Okt. 2012). DOI: 10.1109/vast.2012.6400492. URL: http://dx.doi.org/10.1109/VAST.2012.6400492 (zitiert auf S. 21).

[HW13] J. Heinrich, D. Weiskopf. „State of the art of parallel coordinates“. In: *STAR Proceedings of Eurographics* 2013 (2013), S. 95–116 (zitiert auf S. 14).

[JFK16] D. Jäckle, J. Fuchs, D. Keim. „Star Glyph Insets for Overview Preservation of Multivariate Data“. In: *Electronic Imaging* 2016.1 (2016), S. 1–9 (zitiert auf S. 22).

[Llo82] S. Lloyd. „Least squares quantization in PCM“. In: *IEEE transactions on information theory* 28.2 (1982), S. 129–137 (zitiert auf S. 18).

[Pla98] J. Platt. „Fast Training of Support Vector Machines Using Sequential Minimal Optimization“. In: *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Jan. 1998. URL: https://www.microsoft.com/en-us/research/publication/fast-training-of-support-vector-machines-using-sequential-minimal-optimization/ (zitiert auf S. 17).

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1-55860-238-0 (zitiert auf S. 17).

[RHW88] D. E. Rumelhart, G. E. Hinton, R. J. Williams. „Learning representations by back-propagating errors“. In: *Cognitive modeling* 5.3 (1988), S. 1 (zitiert auf S. 18).

[Shn96] B. Shneiderman. „The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations“. In: *Proceedings of the 1996 IEEE Symposium on Visual Languages*. VL '96. Washington, DC, USA: IEEE Computer Society, 1996, S. 336–. ISBN: 0-8186-7508-X. URL: http://dl.acm.org/citation.cfm?id=832277.834354 (zitiert auf S. 16, 25).

[War02] M. O. Ward. „A taxonomy of glyph placement strategies for multidimensional data visualization“. In: *Inf Vis* 1.3-4 (Dez. 2002), S. 194–210. DOI: 10.1057/palgrave.ivs.9500025. URL: http://dx.doi.org/10.1057/palgrave.ivs.9500025 (zitiert auf S. 15).

[Wer12] M. Wertheimer. „Experimentelle studien über das sehen von bewegung“. In: *Zeitschrift für Psychologie* 61 (1912), S. 161–265 (zitiert auf S. 12).

[Wes07a] Weston.pace. *This image is part of a series of images showing an example of the operation of the k-means algorithm. This is the second step in which data points are associated with the nearest centroid.* https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_2.svg. 2007. URL: https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_2.svg (zitiert auf S. 19).

[Wes07b] Weston.pace. *This image is part of a series of images showing an example of the operation of the k-means algorithm. This is the third step where the centroids are moved to the average of all the data points.* https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_3.svg. 2007. URL: https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_3.svg (zitiert auf S. 19).

[Wes07c] Weston.pace. *This image is part of a series of images showing the operation of the k-means algorithm. This is the fourth step (a repetition of the second step) where the data points are associated with their nearest centroids.* https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_4.svg. 2007. URL: https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_4.svg (zitiert auf S. 19).

[Wes07d] Weston.pace. *This image is part of an example of the K-means algorithm. This is the first step, where the points and centroids are randomly placed.* https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_1.svg. 2007. URL: https://commons.wikimedia.org/wiki/File:K_Means_Example_Step_1.svg (zitiert auf S. 19).

[xpl06] M. xplore. *scatterplot ms.* https://commons.wikimedia.org/wiki/File:Scatterplot.jpg. 2006. URL: https://commons.wikimedia.org/wiki/File:Scatterplot.jpg (zitiert auf S. 13).

Alle URLs wurden zuletzt am 29. 09. 2016 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift