Klassifikation und Clustering

Krainer Philipp

B A C H E L O R A R B E I T

Nr. 1310458007-A

eingereicht am
Fachhochschul-Bachelorstudiengang

Medizin- und Bioinformatik

in Hagenberg

im Mai 2016

Diese Arbeit entstand im Rahmen des Gegenstands

Softwareentwicklung mit klassischen Sprachen

im

Sommersemester 2015

Betreuer:

Stephan Winkler

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Hagenberg, am 30. Mai 2016

Krainer Philipp

Inhaltsverzeichnis

Kurzfassung

Die vorliegende Arbeit setzt sich mit den heuristischen Algorithmen der Klassifikation und des Clustering auseinander.
Das Clustering, welches auch als unüberwachte Klassifikation bekannt ist, wird als eine Methode zur Einteilung von großen Datensätzen angewandt. Hier wird auf den unterschiedlichen Bezug der Daten auf das Clustering und dessen Algorithmen eingegangen. Verschiedene Algorithmen geben einen guten Einblick in die komplexe Welt des Clustering. Grundsätzlich gilt, dass jede Methode ihr eigenes Einsatzgebiet innehat und an die vorliegenden Daten anzupassen ist.
In weiterer Folge wird mit der Klassifikation eine weitere Methode beschrieben Daten einzuteilen. Im Gegensatz zum Clustering werden bei Klassifikation Daten anhand vorgegebener Klassen zugeordnet.
Auch hier wurden die verschieden Algorithmen unter der Berücksichtigung der Klassifikationsgüte beschrieben und verglichen. Alle Algorithmen verwenden bereits vordefinierte Daten und Methoden um die Berechnung zu beschleunigen. Daher ist die Klassifikation weiterverbreiteter als das Clustering.
Es wurden die Algorithmen und Methoden anhand von Beispielen entsprechend dargestellt. Dabei kamen das Heuristiclabtool sowie Daten aus UCI Repository zur Anwendung. Diese Beispiele zeigen ziemlich gut welche Algorithmen für welche Datensätze geeignet sind und beste Ergebnisse liefern.

Abstract

This thesis deals with the comparison of heuristic algorithms of classification and clustering.
First we discuss clustering as a part of the unsupervised classification. This is used to divide big data into smaller and different parts. Different algorithms and Methods describe the common used strategies of clustering relatively well. Basically every data set is different and not every algorithm can be applied to it, so every method has to be customized to fit into the data set properties.
In contrast to the Clustering the Classification separates data sets based on predefined classes in groups by using train data and test data. Different algorithms are described to show a number of different methods to use classification. A main measurement for the algorithms is basically classification quality. All algorithms use predefined data to boost computing performance. That means Classification is used more often than clustering. Finally some examples and tests of the different algorithms of classification and clustering are described.
The HeuristicLab software is used for the calculation of the algorithms and the sample data is provided from the UCI repository. All samples show us which kind of algorithms can be used for the different data sets and the best results.

Kapitel 1

Einleitung

1.1 Einführung

In dieser Bachelorarbeit werden die Themen *Klassifikation* und *Clustering* in Bezug auf heuristische Methoden behandelt. In der heutigen Welt sind große Datenmengen an der Tagesordnung. Die Hardware ist zwar leistungsfähig genug, kann aber große Datensätze nicht auf einmal verarbeiten, da die zeitliche Datendurchsatzrate zu gering ist. Daher ist es entscheidend große Datenmengen in kleinere Einheiten mit Hilfe von Klassifikation und Clustering überzuführen. Der Begriff *Big Data* bestimmt die Welt der Analyse und der Verarbeitung von Informationen, welche meist als mehrdimensionale Daten beschrieben werden können. Daher ist es wichtig geeignete Algorithmen und Verfahren zu finden, welche große Datenmengen in anwendbare Einheiten einteilen.
Das Ziel der Analyse von eingeteilten Daten ist es eine verständliche und interpretierbare Konzeption zu erreichen, welche auf einem geeigneten Modell basiert. Dieses Modell kann nur auf der Basis von eingeteilten Daten funktionieren, da es sonst zu komplex wird. Das Clustering und die Klassifikation machen diese Vorgangsweise erst möglich. Sie sind daher ein zentraler Bestandteil bei der Analyse von Daten und in der Heuristik und in der Statistik nicht mehr wegzudenken.

1.2 Aufgabenstellung

In der Klassifikation geht es darum Samples in verschiedene Klassen einzuteilen. Beim Clustering dagegen geht es darum Gruppen von Datenpunkten zu identifizieren. Ist es sinnvoll Daten vor dem Anwenden von Klassifikationsalgorithmen zu gruppieren, also Clustering und Klassifikation zu kombinieren. Lässt sich so die Prognosegenauigkeit erhöhen? Ziel dieser Arbeit ist es, Antworten auf diese Fragen zu finden; Frameworks wie das HeuristicLab und WEKA sowie international bekannte Benchmark-Datensätze können für

die entsprechenden Tests verwendet werden.

1.3 Terminologien und Begrifflichkeiten

- **Heuristik:**
  Aus dem Griechischen heuriskein = finden, entdecken, bezeichnet eine Erfinderkunst. Heuristik ist die Lehre von verschiedenen Verfahren zum Lösen von Problemen, welche nicht mit mathematischen Algorithmen bzw. Formeln gelöst werden können.
- **Datensatz:**
  Ein Datensatz ist die Zusammenfassung von Daten, die in einer direkten Beziehung zueinander stehen oder gemeinsame Merkmale haben. Daten, die in einem Sinnzusammenhang stehen, können dabei in einem Ordnungssystem zusammengefasst sein.
- **Wahrscheinlichkeit:**
  Die Wahrscheinlichkeit ist ein Maß zur Quantifizierung der Sicherheit bzw. Unsicherheit des Eintretens eines bestimmten Ereignisses im Rahmen eines Zufallsexperiments
- **Algorithmus:**
  Ein Algorithmus ist eine eindeutige, ausführbare Folge von Anweisungen endlicher Länge zur Lösung eines Problems. Ein Algorithmus besteht aus einem Deklarationsteil und einem Anweisungsteil.
- **Sample (Sampling):**
  Teilmenge einer Grundgesamtheit, die für eine Untersuchung ausgewählt wird.
- **Satz von Bayes:**
  Der Satz von Bayes ist ein mathematischer Satz aus der Wahrscheinlichkeitstheorie, der die Berechnung bedingter Wahrscheinlichkeiten beschreibt. Formel:

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

- **Test:**
  Tests sind Methoden, mit denen eine Entscheidung über die Beibehaltung oder Zurückweisung einer Nullhypothese H_0 mithilfe eines Stichprobenbefundes getroffen wird.
- **Graph:**
  Ein Graph ist in der Graphentheorie eine abstrakte Struktur, die eine Menge von Objekten zusammen mit den zwischen diesen Objekten bestehenden Verbindungen repräsentiert. Die mathematischen Abstraktionen der Objekte werden dabei Knoten des Graphen genannt. Die paarweisen Verbindungen zwischen Knoten stellen Kanten dar.

- **Daten:**
  - numerisch:
    Daten, die mit Ziffern und zusätzlichen Sonderzeichen dargestellt werden.
  - nicht-numerisch:
    Daten, die aus Buchstaben und Ziffern zusammengesetzt sind.
- **Zeitreihe:**
  Ein Zeitreihe ist eine Serie von Messungen, Beobachtungen und Aufzeichnungen von Variablen an aufeinanderfolgenden Zeitpunkten. Zeitreihen ermöglichen eine strukturierte Darstellung von Daten. Eine visuelle Darstellung entspricht einer Kurve, die sich mit der Zeit entwickelt.
- **Dimension:**
  Der Begriff Dimension bezeichnet im Allgemeinen lediglich ein unabhängiges Merkmal eines Datensatzes. Dimensionen haben mit den Daten selbst gar keine echte Verwandtschaft, sondern stellen meist ein unabhängiges Gedankenkonstrukt dar, das Analogien zum Datensatz herstellt, um es berechenbar oder messbar zu machen.
- **Distanzmaß:**
  Als Distanzmaß wird ein Maß bezeichnet, wenn es die Unähnlichkeit zwischen zwei Objekten misst. Es besitzt die Eigenschaft, dass es mit zunehmender Unterschiedlichkeit zweier Objekte ansteigt.
- **Kostenfunktion:**
  Als Kostenfunktion wird jene Funktion beschrieben, welche bestimmt wie komplex und aufwendig ein Algorithmus oder Verfahren ist. Meist wird diese Funktion mit der O-Notation gleich gesetzt, welche vor allen in der Laufzeitmessung angewandt wird.
- **Lagemaße:**
  - Mittelwert:
    Der Mittelwert beschreibt den statistischen Durchschnittswert und wird auch arithmetisches Mittel genannt.
  - Median:
    Der Wert, der genau in der Mitte einer Datenverteilung liegt, nennt sich Median oder Zentralwert. Die eine Hälfte aller Daten ist immer kleiner, die andere größer als der Median.
  - Modus:
    Der Modus gibt an, welche Merkmalsausprägung in einem Datensatz am häufigsten vorkommt.
- **Heatmap:**
  Eine Heatmap ist ein Diagramm zur Visualisierung von Daten, deren abhängige Werte einer zweidimensionalen Definitionsmenge als Farben repräsentiert werden. Sie dient dazu, in einer großen Datenmenge

intuitiv und schnell besonders markante Werte zu erfassen.

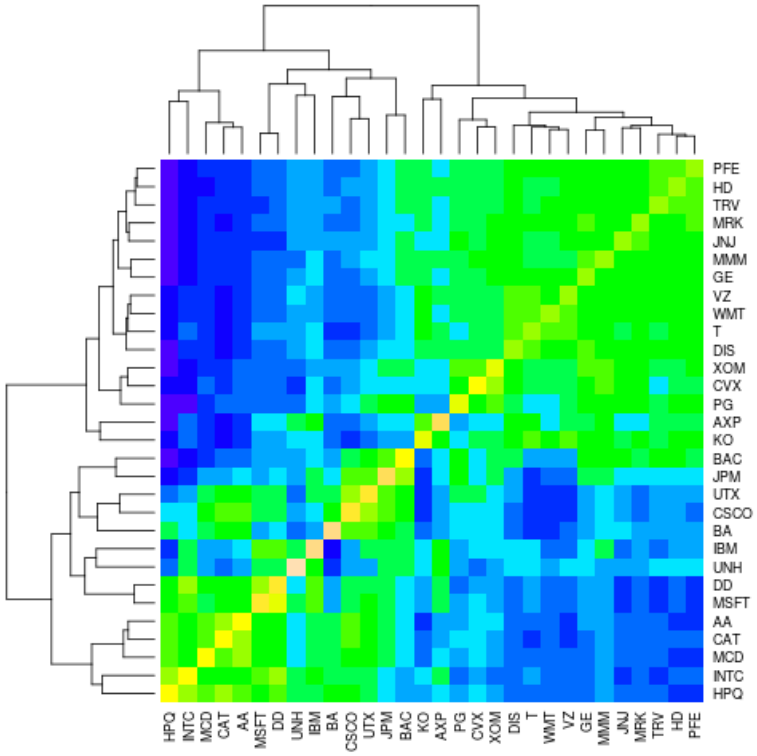

Abbildung 1.1: heatmap

- **Cluster:**
  Als Cluster bezeichnet man in der Informatik und Statistik eine Gruppe von Datenobjekten mit ähnlichen Eigenschaften.
- **Link Arten:**
  - Single-Link:
    Beschreibt die kleinste Entfernung von den Clustern und wird auch als nächster Nachbar bezeichnet.
  - Average-Link:
    Beschreibt die mittlere Distanz von Clustern zueinander
  - Complete-Link:
    Beschreibt die maximale Entfernung von Clustern zueinander.

Kapitel 2

Clustering

2.1 Einführung

In der Informatik und Statistik sind große Datensätze anzutreffen. Diese können analysiert werden, wenn entsprechende Tools und Algorithmen zur Verfügung stehen. Durch das Verfahren von Clustering und Bildung von eingeteilten Datensätzen wird das Arbeiten mit großen Daten erleichtert. Das Ziel ist ein geeignetes Modell für einen gegeben Datensatz zu finden. Durch diese Methoden wird das Interpretieren von Merkmalen und Besonderheiten erleichtert und ermöglicht außerdem eine begünstigte Aufbereitung von den Daten.[3]
Um die richtigen Modellrepräsentationen zu finden und diese zu identifizieren ist es notwendig die Daten zuerst in Klassen einzuteilen. Damit beschäftigt sich die Klassifikation. Dies ist sehr hilfreich wenn ein klassenorientiertes Clustering vorgenommen wird, da die Einteilung zu relevanten Merkmalen zuerst erfolgen muss. Dabei wird zwischen zwei Arten unterschieden: Die überwachte (*supervised*) und die unüberwachte(*unsupervised*) Klassifikation.[3]
Clustering gehört zu der unüberwachten Klassifikation, welche die Daten in relevante Klassen sogenannte Cluster einteilt.
Das Einsatzgebiet von Clustering-Algorithmen ist vielfältig. Die Anwendungsgebiete sind vor allem im Gebiet von Data-Mining und Data-Analysis zu finden. Weitere Anwendungsgebiete sind die Bioinformatik, Analyse von Datenbanken, Textmining u. Neuronale Netzwerke. Clustering spielt bei der Datenanalyse eine große und bedeutende Rolle.[3]

2.2 Datenrepräsentation

Als *Daten* wird eine Ansammlung bzw. eine Menge an Dingen oder Objekten bezeichnet. Diese Definition bezieht sich auf den Zusammenhang von Daten und Clustering. Jedes einzelne Objekt besitzt spezielle Eigenschaften,

welche eindeutig per Objekttyp sein können. Diese Eigenschaften werden oft auch als Attribute, Merkmale oder Dimensionen bezeichnet.[7]
In einen Datenraum befinden sich Objekte oder Elemente mit einer endlichen Anzahl von Merkmalen, welche bei allen Objekten denen des Datenraumes gleichen. Jedoch können sich Daten die sich innerhalb dieses vorgegebenen Raumes befinden sich unterscheiden, da die Ausprägung nicht vorgegeben ist. [7] Dabei gibt es eine mathematische Repräsentation:

D, D_d : Datenraum (auch Merkmalsraum) der Dimension d (auch $\mathbb{R}^d$)

S : Datenmenge, $S \subset D$

x_i : i-tes Objekt aus S

$|S|, n$: Mächtigkeit von S (Anzahl von Objekten)

$$\begin{array}{l} \\ \text{Objekte} \end{array} \overbrace{\begin{cases} x_1 = & (x_{1,1} & \cdots & a_{1,d}) \\ x_2 = & (x_{2,1} & \cdots & a_{2,d}) \\ \vdots & \vdots & \ddots & \vdots \\ x_n & (x_{n,1} & \cdots & a_{n,d}) \end{cases}}^{\text{Attribute}}$$

Jedes einzelne Objekt aus der Datenmenge S hat d Merkmale/Attribute. Diese werden durch die Art (des Typs) unterschieden. Dabei wird zwischen *numerisch* und *nicht-numerisch* Daten unterschieden.

Erstens betroffen sind die Daten bzw. Objekte in Vektoren von reelwertigen Zahlen. Als Beispiel sei eine Zeitreihe von einer Messung genannt. Zweitens betrifft den Rest, welcher in ein numerisches Format übertragen werden kann. Im Allgemeinen können *nicht-numerische* Merkmale durch spezifische Codierungen in numerische Merkmale übertragen werden. Dabei wird jedes einzelne Merkmal durch ein oder mehrere Attribute repräsentativ dargestellt.[7]
Diese Methode wird verwendet um *numerische* und *nicht-numerische* Daten gleich zu behandeln, da es dabei keinen Unterschied in der Anwendung bzw. Auswertung gibt. Diese Methode ist notwendig um die Daten in ein Format zu bringen, welches für das Clustering verwendet werden kann. Dann können die Daten von einem geeigneten Algorithmus verarbeitet werden.[7]
Doch in der realen Welt sind große und komplexe Daten an der Tagesordnung. Doch im Gegensatz dazu würde es sich sehr aufwändig gestalten, wenn

Algorithmen mit hoher Dimensionalität an Daten verwendet werden. Da sich dabei die Rechenzeit erheblich erhöhen würde müssen Methoden angewendet werden um die Datenkomplexität zu reduzieren. Bei vielen Datensätzen sind zu viele Merkmale vorhanden, die sogar irrelevant für die Berechnung sind. Manche können den Algorithmus sogar in eine falsche Richtung führen.[7]
Einerseits gibt es die Möglichkeit gewisse Merkmale auszuklammern d.h. diese werden nicht in die Berechnung aufgenommen um die Komplexität zu verringern. Dabei gehen keine wichtigen Attribute verloren. Dieses Verfahren wird als Merkmalsauswahl(*featuren selection*) genannt. Es ist oft hilfreich nur gewisse Merkmale auszuwählen, um eine Selektion von den Besten zu ermöglichen (auch Elitismus genannt), denn dann sind die Algorithmen performant und liefern annehmbare Ergebnisse.[7]
Eine weitere Möglichkeit ist es gewisse Merkmale aus einer Ansammlung auszuwählen. Dadurch kann auch eine Featurereduktion erreicht werden. Dies ist ebenso so effizient wie die Auswahl der Merkmale. Diese Methode wird Merkmalsextraktion(*feature extraction*) genannt. Dabei werden nur die wichtigsten Merkmale herangezogen, um die Performance zu steigern.[7]
Neben den beiden Methoden ist es auch notwendig die Daten zu normalisieren d.h es müssen die Daten auf die gleiche Weise umgerechnet werden, damit sie besser zusammenpassen. Dies wird durch skalieren und konvertieren erreicht. Die Normalverteilung der Daten wird angenommen um die Skalierung zu erleichtern; mit denselben Mittelwert ($\bar{x}$) und der selben Standardabweichung (σ), mit $\bar{x} = 0$ und $\sigma = 1$.[7]
Bevor die Algorithmen angewendet werden können, muss ein geeignetes Maß für den Abstand gefunden werden. Die Maße sind metrisch u. es wird daher der Überbegriff der Ähnlichkeitsbestimmung verwendet. Diese Maße geben an wie ähnlich sich zwei Objekte sind und dabei wird nicht zwischen *numerisch* und *nicht-numerisch* unterschieden. Das Distanzmaß (*distance measure*) oder Ähnlichkeitsmaß(*similarity/proximity/affinity measure*) definiert die Beziehung bzw. die Funktion $d : D \times D \to Z$. Die Maße sind essentiell für das Clustering und müssen vor den eigentlichen Clustering ausgeführt werden.[3]

2.3 Algorithmen

2.3.1 Allgemein

Nachdem ein geeignetes Distanzmaß gefunden ist, kann ein bestimmter Algorithmus auf die Daten angewendet werden. Es gibt zwei Gruppen von Verfahren bzw. Algorithmen, welche das *Hierarchisches Clustering* und die *Partitionierung* darstellen. Bei der Partitionierung werden die Objekte bzw. Daten in Gruppen eingeteilt und diese Gruppen enthalten keine weiteren verschachtelten Cluster und besitzen nur eine Ebene. Beim *Hierarchischen*

Clustering entsteht ein geschachtelter Aufbau bzw. eine Struktur, wo größere Cluster kleinere enthalten.[3]

Weiters können die beiden oben angeführten Methoden weiter aufgegliedert werden:[3]

- *divisiv:*
  Bei dieser Methode werden alle Objekte einen Cluster zugeordnet sowie schrittweise verkleinert, indem schrittweise zerteilt wird, bis ein vordefiniertes Abbruchkriterium eintritt.

- *agglomerativ:*
  Im Gegensatz zu der divisiven Methode wird bei der agglomerativen Methode mit kleinen Clustern begonnen. Jedes Objekt stellt einen Cluster für sich dar. Die kleinen Cluster werden schrittweise zusammengefügt bis ein Abbruchkriterium eintritt.

- *hard:*
  Algorithmen welche das Prinzip von einer strikten Vorgehensweise (*hard*) verfolgen ordnen einen Cluster ein Objekt zu.

- *fuzzy:*
  Im Gegenteil dazu gibt es das Prinzip von der ungenauen Vorgehensweise(*fuzzy*), dabei können Objekte verschiedenen Clustern zugeordnet werden.

- *stochastisch:*
  Bei dem Begriff stochastisch kann davon ausgegangen werden, dass der Zufall eine Rolle spielt und die Auswahl verschiedener Objekte oder Attribute keiner Regel folgt.

- *deterministisch:*
  Dabei handelt es sich um die Vorgabe keiner zufälligen Ereignisse. Hier muss alles vorgegeben sein damit es als deterministisch gilt.

- *monothetisch:*
  Wenn bei der Verarbeitung nur ein Cluster bzw. ein Objekt verarbeitet wird, dann wird dieses Verfahren monothetisch bezeichnet. Aber die Algorithmen arbeiten nur bedingt nach diesem Prinzip, da dadurch die Berechnungszeit erhöht sein kann.

- *polythetisch:*
  Bei der polythetischen Vorgangsweise werden Cluster bzw. Objekte

oder Daten schneller verarbeitet, da Vorgänge gleichzeitig ausgeführt werden. In Bezug auf das Clustering bezieht sich die Gleichzeitigkeit auf die Distanzberechnung der Merkmale.

2.3.2 k-means Algorithmus

Beschreibung:

Der *k-means* Algorithmus gehört zu den Partitionierungs-Algorithmen. Die Implementierung ist einfach und liefert trotzdem gut interpretierbare Ergebnisse für einfache Aufgabenstellungen. Grundlegend versucht der Algorithmus eine Partition in den Daten zu finden und daraus dann Cluster zu bilden. Die Anzahl der Cluster wird durch den Anwender festgelegt und während der Laufzeit nicht mehr geändert. Die Formel nach dem die Cluster gebildet werden lautet: [7]

$$x_r^i : r\text{-te Element des Clusters } C_i$$

Diese Methode wird auch als Sum-of-Squares bezeichnet. Dabei werden die quadratischen Abstände minimiert. Beim Clustering bedeutet dies, dass die Ähnlichkeit der Attribute, Merkmale oder Objekte bestimmt wird. Damit basieren Cluster auf der oben angeführten Kostenfunktion.

Algorithmus:

1. Wähle zufällig k Cluster-Zentren $\mu_1, \ldots, \mu_k$.

2. Berechne für jedes $x \in S$, zu welchen Clustermittelpunkt μ_i es am nächsten liegt.

3. Berechne für jeden Cluster C_i die Kostenfunktion:

$$c(C_i) = \sum_{r=1}^{|C_i|} (d(\mu_i, x_r^i))^2$$

4. Berechne für jeden Cluster C_i den eigenen neuen Mittelpunk:

$$\mu_i = \frac{1}{|C_i|} \sum_{r=1}^{|C_i|} x_r^i$$

5. Wiederhole 2., 3., 4. bis sich die Clusterzuordnung nicht mehr ändert.

Die Datensätze besitzen einen Mittelwert, da diese *numerisch* sind. Daher kann ein Mittelwert oder auch das arithmetisches bzw. geometrisches Mittel gebildet werden. Auch beim Clustering können *nicht-numerische* Datensätze verwendet werden. Diese besitzen meistens keinen numerischen Mittelwert.

Dennoch kann ein Lagemaß berechnet werden: der Median. Dieser gibt ähnlich wie der Mittelwert eine gute Aussage wie die Daten verteilt sind und es können auch damit *nicht-numerischen* Daten berechnet werden. Beim Clustering wird der ähnliche *k-medioids* Algorithmus angewandt, welcher nach dem oben genannten Prinzip funktioniert. Nur wird bei der Berechnung der Mittelwert μ durch den Median ersetzt. [7]

Zusammenfassung

Dieser Clusteringalgorithmus ist einfach in seiner Komplexität, da er sich nur auf die Mittelwerte der einzelnen Attribute bezieht. Doch bei der Bildung von Clustern können einfach sphärische Cluster entstehen, da die Berechnung relativ zum Mittelwert geschieht. Weiteres kann ein vorhandenes Rauschen in den Daten (Störung in den Daten) bei diesem Algorithmus nicht beseitigt werden, da der Mittelwert sehr ausreißerempfindlich ist.

2.3.3 Hierarchisch Clustering

Beschreibung:

Diese Methode wird verwendet wenn die Daten nicht offensichtlich in Gruppen bzw. Partitionen eingeteilt sind oder es keine separierte Cluster gibt. Diese Methode erstellt eine hierarchische Baumstruktur der Datenmenge. Dabei sind die einzelnen Knoten bzw. Enden jeweils eine Teilmenge des übergeordneten Knotens. Der Wurzelknoten repräsentiert die gesamte Menge und die Blätter die einzelnen Objekte.

Bei diesem Algorithmus werden *bottom-up* und *top-down* als Verfahren unterschieden

Bei der *bottom-up* Methode wird anfangs von kleinen Elementen bzw. Clustern ausgegangen. Diese werden immer weiter kombiniert bis ein gemeinsamer Megacluster entsteht, welcher den ganzen Datensatz enthält. Wenn ein großer Cluster entstanden ist, ist der Algorithmus durchlaufen bzw. das Clustering abgeschlossen.[5, 6]

Im Gegensatz dazu wird bei der *top-down* Methode von einen einzelnen Cluster ausgegangen, welcher den ganzen Datensatz enthält. Hier wird schrittweise der übergeordnete Cluster, auch *parent* genannt in mehrere kleinere Cluster (*child*)zerlegt. Diese stellen den Eltern-Cluster dar. Wenn in jeden

Cluster nur mehr ein Element vorhanden ist, ist das Clustering abgeschlossen.

Beide Methoden verwenden eine Baumstruktur im Hintergrund, in der die einzelnen Cluster als Knoten repräsentiert werden. Dadurch kann mit größeren und komplexeren Datensätzen gearbeitet werden. [5, 6]

Algorithmus

Der Algorithmus wird anhand der der *bottom-up* Methode erklärt. Dabei wird eine Vereinigung von zwei Clustern verwendet. Die zweite Methode *top-down* kann durch durch teilen der Cluster beschrieben werden:

1. Beginne mit n Clustern $C_1, \ldots, C_n$; wobei $C_i = x_i$.

2. Minimiere die Kostenfunktion $c(C_i, C_j)$, um die *beste* bzw. *günstigste* Vereinigung ($C_i \cup C_j$) zu finden.

3. Ersetze C_i und C_j durch die Vereinigung $C_i \cup C_j$.

4. Wiederhole die Schritte 2 und 3 bis alle Cluster zusammengefasst sind.

Das hierarchische Clustering ist nur ein Verfahren, dass einzelne Algorithmen implementiert. Die Vorgehensweise unterscheidet sich nur in der Ausführung des Verfahrens. Es wird unterschieden zwischen *Single-Link*, *Average-Link* und *Complete-Link*. Dabei unterscheiden sich die Verfahren nur in der Kostenfunktion:

- *Single-Link*:

$$c(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

- *Average-Link*:

$$c(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$

- *Complete-Link*:

$$c(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$

[3, 5, 6]

Zusammenfassung

Die Datenmatrix bzw. der Datenvektor spielen bei dieser Methodik eine geringe Rolle, da eigentlich beim Clustering speziell beim hierarchischen meistens auf die Distanzen Rücksicht genommen wird und diese als Eingabe eingesetzt werden. Meist werden diese Distanzen durch eine Matrix repräsentiert. Die Dimensionen werden als $n \times n$ dargestellt, welche bei großen Datenmengen oder großen n zu Speicherproblemen führen können, da die Datenmenge sehr schnell ansteigen kann.
Abhilfe kann geschaffen werden, indem ein Schwellenwert festgelegt wird. Damit werden unbedeutende Wertepaare vernachlässigt. Weiters können auch die Anzahl der verwendeten Elemente begrenzt werden und dadurch die repräsentative Menge der Elemente reduziert werden. Auch können die Verlinkungen zu Nachbarn begrenzt werden um die Anzahl der nächsten Nachbarn zu begrenzen.

2.3.4 Self-organizing Maps

Beschreibung:

Bei diesen Clusteringverfahren wird ein mehrdimensionaler Datensatz mit einer bestimmten Dimensionalität auf ein Gitter mit wenig Dimensionalität meist ein - oder zweidimensional projiziert. Die Anzahl der Cluster ist festgelegt und kann durch die Spalten und Zeilen im Gitter festgelegt werden. Die Referenzvektoren werden beim SOM Clustering aus den Knoten im Gitter gebildet und durch eine iterative Annäherung anhand des vorgegeben Algorithmus zu den Eingabevektoren geleitet. [8]

Algorithmus:

1. Wähle ein Gitter mit $k = k_v \times k_h$ Knoten $v, v \in \{1, \ldots, k\}$.

2. Initialisiere k d-dimensionale Vektoren $f_0(v)$, durch zufällige Wahl von Objekten $x \in S$ oder vollständig zufällig

3. Iteration i:
   (a) Für jedes Objekt $x \in S$ bestimme den Knoten v_x, für den $f_i(v_x)$ am nächsten zu x liegt.

   (b) Aktualisiere alle Referenzvektoren wie folgt:

$$f_{i+1} = f_i(v) + \eta(d(v,_x), i) \cdot (x - f_i(v))$$

$\eta(d, i)$: Lernrate; Die Lernrate nimmt mit der Distanz zwischen den Knoten und der Iteration ab.

(c) Wiederhole (a) und (b) bis keine Veränderung mehr eintritt.

[8]

Zusammenfassung

Bei dem SOM-Algorithmus ist die Reduktion der Dimensionalität an erster Stelle und kann somit sehr effizient zum Berechnen von großen Datensätzen herangezogen werden, da die Dimensionalität bzw. die Anzahl der Merkmale gering ist. Auch eignet sich der Algorithmus um Daten aufzuteilen, da die Clusteranzahl vorbestimmt ist und so ein vereinfachter Algorithmus angewendet werden kann.

2.3.5 Graph-based Clustering

Beim *Graph-based Clustering* wird von einem Graphen ausgegangen, welcher die Distanzmatrix repräsentiert. Die Knoten im Graphen werden Objekten aus der Datenmenge zugeordnet. Die verbindenden Linien oder auch Kanten entsprechen der Distanz zwischen den einzelnen Objekten. Diese können *gerichtet* oder *ungerichtet* sein.[3, 5]
Das Verfahren versucht den Graphen zu zerteilen und ist auch in der Heuristik als Graphpartionsproblem bekannt. Meistens wird eine rekursive Bipartitionierung angenommen, da diese sehr effizient zu berechnen ist. Dabei sind die Graphen eine repräsentative Darstellung von Ähnlichkeitsbeziehungen der einzelnen Objekte.[5]
Eine weit verbreitete Methode bei Clustering mit Hilfe von Graphen wird *Clique-based Clustering* genannt. Dabei wird durch die Cliquengraphen die Beziehungen zwischen Objekten und deren Ähnlichkeit gezeigt. Im Idealfall sind die Objekte in einen Cluster sehr ähnlich zueinander und die Objekte die zu anderen Clustern gehören sind unähnlich zueinander. Dabei sind die Knoten im Graph die Objekte und die Cliquen sind die Cluster. Die Kanten symbolisieren dass die Elemente ähnlich zueinander sind.[3, 5]
Doch in der Praxis können Ähnlichkeitsbeziehungen nur bedingt durch Cliquengraphen dargestellt werden , da Kanten fehlen oder Kanten mehrfach vorhanden sein können. Es gibt ein Modell welches sich *corrupted clique graph Model* nennt, bei dem die Kanten die Wahrscheinlichkeiten darstellen und gewichtet sind. Es wird versucht vom *corrupted* zum originalen Graphen zu gelangen, welcher die richtigen Cluster repräsentiert.
[3]

2.3.6 Spectral Clustering

Beschreibung:

Das *Spectral Clustering* ist ein Partitionierungsalgorithmus, welcher die Eigenvektoren der einzelnen Cluster verwendet und dann damit eine Beziehung zu anderen Clustern erstellt. Bei diesen Verfahren wird die Ähnlichkeitsmatrix herangezogen und die Anzahl der Cluster kann vom Benutzer festgelegt werden.

Algorithmus

Gegeben ist die Datenmenge $S = \{x_1, \ldots, x_n\}$ im $\mathbb{R}^d$; k : : Clusteranzahl

1. Berechne die Ähnlichkeitsmatrix $A_{n\times n}$

$$A_{ij} = \begin{cases} e^{\frac{-|x_i - x_j|^2}{2\sigma^2}} & \text{falls } i \neq j \\ 0 & \text{sonst} \end{cases}$$

σ^2 : Skalierungsfaktor

2. Berechne die Diagonalmatrix $D_{n\times n}$

$$D_{ij} = \begin{cases} \sum_{l=1}^{n} A_{ij} & \text{falls } i = j \\ 0 & \text{sonst} \end{cases}$$

In der Diagonale D stehen die Zeilensummen von A
Berechne $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$

$$D^{-\frac{1}{2}} = \begin{cases} \frac{1}{\sqrt{D_{ij}}} & \text{falls } i = j \\ 0 & \text{sonst} \end{cases}$$

3. Finde $v_1, \ldots, v_k$, die k größten Eigenvektoren von L, so dass alle v_i paarweise orthogonal sind. Erstelle daraus eine Matrix

$$X_{n\times k} = [v_1, \ldots, v_k] \in \mathbb{R}^{n\times k}$$

4. Konsruiere Matrix $Y_{n\times k}$ durch normalisierung von X

$$Y_{ij} = \frac{X_{ij}}{\sqrt{\sum_j X_{ij}^2}}$$

5. Jede Zeile Y_i von Y ist ein Punkt $\mathbb{R}^k$ im Clustere diese Punkte mit einem beliebigen Clusteralgorithmus

6. Weise jedem Originalpunkt x_i den Cluster j genau dann zu, wenn die Zeile Yi im Cluster $j$liegt.

[4]
Beim spektralen Clustering ist die Form des Cluster nicht so bedeutsam, da die Form von den eingegeben Daten abhängt und es eine recht einfache Implementierung mit verschieden Sprachen gibt. Die Clusteranzahl muss vorher gewählt werden und dies kann sich unter Umständen schwierig gestalten.
[3, 6, 1]

2.4 Anwendungsgebiete

Das Clustering hat viele Anwendungsgebiete, da in der Informatik und Statistik große Dateien und Datensätze vorkommen. Clustering und Klassifikation sind vor allen in der Heuristik sehr bedeutsam. Auch in der Medizin spielt Clustering eine wichtige Rolle, da in den Anwendungen in der Medizin Daten von Patienten in Klassen eingeteilt werden müssen wie z.B. AML/ALL Klassifikation der Krebsmerkmale. Auch können in der Biologie große Datenmengen anfallen. Diese müssen aufgeteilt werden und in Gruppen eingeteilt werden wie z.B. die Zuordnung von Primer an der DNA/RNA.[4]
Wie schon oben beschrieben spielt das Clustering als ein Verfahren für die Klassifikation in vielen Anwendungsgebieten eine bedeutende Rolle und wird auch gerne als Hilfsmittel für diverse Berechnungen herangezogen. Das Clustering wird häufig in Verbindung mit der Klassifikation eingesetzt und damit wird noch eine breitere Anwendung ermöglicht. [4]

2.5 Visualisierung

Um die Resultate und Ergebnisse betrachten und analysieren zu können, müssen die Daten und deren dazugehörigen Ergebnisse darstellt bzw. visualisiert werden. Dazu werden Diagramme, Graphen oder repräsentative graphische Darstellungen verwendet, welche nur einen kleinen Teil der verwendeten Daten darstellen. Dadurch kann man sehr repräsentative Ergebnisse darstellen.
Es wird für eine Reihe von Ergebnissen z.B. eine Messreihe mit einer zwei- oder dreidimensionale Darstellungsmethode gewählt, welche Heatmaps oder Fitnesslandschaften darstellen. Dabei können die Beziehungen und Zusammenhänge sehr gut dargestellt werden. Bei der Auswertung der Daten können hierarchische Daten bzw. Ergebnisse entstehen. Dann sollte eine andere Darstellungsform gewählt werden, welche die hierarchische Ordnung von den Daten berücksichtigt. Solche Art von Diagrammen werden Dendrogramm genannt. Dabei wird die hierarchische Ordnung als Baumstruktur verwendet. Damit kann der Verlauf von einzelnen Clusteringschritten nachverfolgt werden.

2.6 Zusammenfassung und Ausblick

Das Clustering, welches hier dargestellt wurde, ist eine Methode der unüberwachten Klassifikation. Die verschieden Algorithmen helfen große und komplexe Datenmengen besser zu analysieren und erleichtern die nachträgliche Verarbeitung. Diese müssen interpretiert werden wobei die verschieden Visualisierungsmöglichkeiten hilfreich sind. Wie gut das Clustering interpretiert werden kann hängt sehr stark von den gewählten Parametern bzw. vom gewählten Algorithmus ab.
Die verwendeten Daten entscheiden über die Art des gewählten Algorithmus, da bei manchen Datensätzen Algorithmen keine Ergebnisse liefern, da es für jede Methode Voraussetzungen gibt. So ist es wichtig zuerst abzuklären, welche Voraussetzungen gegeben sind. Dann sollte der richtige Algorithmus ausgewählt werden und nicht umgekehrt.
Diese Algorithmen würden bessere Ergebnisse liefern, wenn die Daten besser angepasst wären, da es bei der Effizienz und Performance meist nur auf die Beschaffenheit der Daten ankommt. Die richtige Wahl der Parameter ist die größte Herausforderung beim Clustering und auch bei den heuristischen Algorithmen und Verfahren. Da häufig kein Wissen über das Verhalten von den Daten *a priori* bekannt ist, ist eine Vorhersage nur schwer möglich.
In Zukunft wird die Entwicklung in Richtung selbst adaptiver Clusteringalgorithmen gehen, welche sich anhand der Daten anpassen und nicht mehr auf Annahmen basieren, welche meist nur sehr schlecht performante Ergebnisse liefern. Auch muss hier die Effizienz per Datensatz gesteigert werden

um zeitlich bessere Ergebnisse zu erhalten.

Kapitel 3

Klassifikation

3.1 Einführung

Bei der Klassifikation werden große Datensätze in Klassen eingeteilt und können durch die Anwendung von Clustering weiterverarbeitet werden. Diese Methoden erlauben es komplexe Daten weiterzuverarbeiten. Durch diese Schritt wird eine Erweiterung der Anwendungsbereiche ermöglicht, da bereits eingeteilte Daten einfacher zu bearbeiten und zu analysieren sind. Die Klassifikation ist ein Teil von Data Mining und Heuristik.
Im Gegensatz zum Clustering ist die Klassifikation eine Methode, welche anhand vorgegebener Trainingsdaten, die Auswahl des richtigen Algorithmus zu erleichtern. Die Zuordnung erfolgt dabei händisch. Damit lassen sich unbekannte Daten mit bestimmten Testdaten und Merkmalen eindeutig in Klassen einteilen. Dabei sind die Klassen und Trainingsdaten vorher bekannt und dies wird auch als überwachtes Lernen bezeichnet. [6]
Bevor mit der Klassifikation begonnen werden kann, müssen die Voraussetzungen definiert werden. Dabei wird von einer bestimmten Menge von Trainingsdaten ausgegangenen, welche bestimmte Merkmale bzw. Attribute aufweisen. Hier wird ein Klassentribut, welches die eindeutige Zuordnung zu einer bestimmten Klasse oder Gruppe besitzt vorgegeben. Das Attribut für die Zuordnung ist immer qualitativ, die restlichen Merkmale können auch quantitativ sein.[6]

Diese Verfahren laufen in zwei Phasen ab. Dabei wird in der ersten Phase anhand dem Vorliegen der Daten, welche Trainingsdaten genannt werden, ein Klassenmodell aufgebaut. Dieses Modell wird in der zweiten Phase zur Zuordnung von den Daten angewandt. Das Klassenattribut ist an sich nicht bekannt um auch diese Daten in Klassen einzuteilen. Ziel der Klassifikation ist es anhand von vorgegeben Modellen Daten zuordnen. [6]

3.2 Klassifikationsgüte

3.2.1 Beschreibung

Bei der Klassifikation ist das Einschätzen der Gütefunktion einfacher als beim Clustering. Da die Klassifikation die Objekte eindeutig zuordnen kann, ist es möglich die *wahre Fehlerrate(true error rate)* zu berechnen und damit den Anteil der falsch klassifizierten Objekte zu bestimmen. Die textuelle mathematische Formel lautet:

$$\textit{true error rate} = \frac{\text{Anzahl der falsch klassifizierten Objekte}}{\text{Anzahl aller Objekte}}$$

Doch wenn sich unter den Daten unbekannte Objekte befinden, gibt es keine Methode die wahre Fehlerrate zu berechnen. Da keine Informationen über die Klassen vorhanden sind, müssen andere Methoden gewählt werden, um eine etwaige Klassifikation zu bestimmen. Nur für die Trainingsdaten kann die wahre Fehlerrate *a priori* bestimmt werden, da diese vor der Berechnung die Klassenzugehörigkeit bekannt ist. Die Fehlerrate für die Trainingsdaten wird *offensichtliche Fehlerrate (apparent error rate)* genannt und lässt sich durch die folgende Formel beschreiben:[6]

$$\textit{apperent error rate} = \frac{\text{Anzahl der falsch klassifizierten Trainingsobjekte}}{\text{Anzahl aller Trainingsobjekte}}$$

In der Statistik wird häufig beschrieben, dass sich die offensichtliche Fehlerrate der wahren Fehlerrate annähert. Wenn genügend Trainingsdaten vorhanden sind, kann die wahre Fehlerrate mit der offensichtlichen Fehlerrate gleich gesetzt werden und so mit die Fitness bzw. Gesundheit der realen Daten bestimmt werden. Bei der Betrachtung von realen Problemstellungen ist die Anzahl der Trainingsdaten kleiner und daher müssen Varianten und Verfahren gesucht werden, welche die wahre Fehlerrate annähernd berechnen können.[10]

3.2.2 Train und Test

Die einfachste Methode ist die Eingabedaten in zwei Teile zu teilen und den einen Teil als Trainingsdaten und den anderen Teil als Testmenge zu verwenden. Die Trainingsmenge wird angewandt um den Klassifikationsalgorithmus die vorgegeben Klassen mitzuteilen und damit dann die Testdaten zu klassifizieren. Die beiden Datensätze müssen unabhängig voneinander sein, da

diese rein zufällig ausgewählt werden und damit kann die Fehlerrate recht gut angenähert werden. Die einzige Voraussetzung ist, dass die Testmenge relativ groß ist, da sonst die Klassifikation sehr schnell ungenau wird. Ansonsten muss auf andere Verfahren zurückgegriffen werden wie beispielsweise auf bestimmte Sampling Techniken. [6]
Auch andere Verfahren können die Klassifikationsgüte relativ gut aus dem Kontext des Anwendungsgebiets berechnen. Aber es ist es relativ schwer gute Ergebnisse zu erreichen und manchmal kann die Güte negativ von dem gewählten Verfahren beeinflusst werden. [10]

3.3 Algorithmen

3.3.1 Bayes-Klassifikatoren

Beschreibung:

Bei der Bayes-Klassifikation wird auf die mathematische Grundlage der Wahrscheinlichkeitsberechnung der einzelnen Klassen aufgebaut. Diese folgen dem *Satz von Bayes* , welcher mit der Formel

$$P(X|Y) = \frac{P(Y|X)\ P(X)}{P(Y)}$$

beschrieben werden kann.
Durch diese Formel kann die Wahrscheinlichkeit, dass ein unbekanntes Objekt einer Klasse angehört, berechnet werden. Die Wahrscheinlichkeit *a posteriori* einer Hypothese X kann unter der Annahme von einer anderen Hypothese Y und anhand der Wahrscheinlichkeiten von X und Y erklärt werden. [2, 6]

Algorithmus

Dieser Algorithmus wird *naiver Bayes-Klassifikator* genannt. Seine Funktionsweise ist, dass ein unbekanntes Objekt einer Klasse bei der die Wahrscheinlichkeit für die bestimmte Klasse am höchsten ist, zugeordnet wird. Die Formel dazu lautet:

C_i: Klasse

x: Unbekanntes Objekt

$$P(C_i|x) = \frac{P(x|C_i)\ P(C_i)}{P(x)}$$

Das Objekt wird nur dann einer Klasse zugewiesen, wenn die Wahrscheinlichkeit ($P(C_i|x)$) ein Maximum darstellt. Die Wahrscheinlichkeit für jede einzelne Klasse ist immer gleich. Dadurch muss nur der Zähler im Bruch maximiert werden und dadurch entsteht eine neue Regel welche lautet:

$$arg \max_{C_i \in \{C_1,\ldots,C_k\}} P(x|C_i)\ P(C_i)$$

Die Wahrscheinlichkeit der einzelnen Klassen kann anhand der Trainingsdaten geschätzt werden. Mit Hilfe nachfolgender Formel kann die Anzahl der Trainingsobjekte pro Klasse bestimmt werden:

$$P(C_i) = \frac{|\{o \in T | o \in C_i\}|}{|T|}$$

Um die Wahrscheinlichkeit der Zuordnung zu den Klassen schätzen zu können muss die Annahme getroffen werden, dass der *naiver Bayes-Klassifikator* die Attribute der einzelnen Objekte als unabhängig betrachtet. Das bedeutet dass sich die Merkmale bzw. die Eigenschaften nicht gegenseitig behindern. Daher lässt sich dann die Klassenspezifische Wahrscheinlichkeit wie folgt berechnen:

$$P(x_j|C_i) = \prod_{j=1}^{d} P(x_j|C_i)$$

Dabei lässt sich die wahre Wahrscheinlichkeit der Objekte mit Hilfe der Trainingsdaten berechnen bzw. abschätzen.

$$P(x_j|C_i) = \frac{|\{y \in T | y \in C_i \wedge y_j = x_j\}|}{|\{y \in T | y \in C_i\}|}$$

[2, 6]

Zusammenfassung

Der *naive Bayes-Klassifikator* geht grundsätzlich von dem Satz von Bayes aus und verwendet diesen Algorithmus um Objekte einer Klasse zuzuordnen. Er beruht auf dem Prinzip der Unabhängigkeit von Eigenschaften, da

sonst die Wahrscheinlichkeiten nicht vollständig aus den Testdaten berechnet werden können. Auch ist anzumerken, dass dieser Algorithmus nicht sehr effizient im Vergleich zur Anzahl der Trainingsdaten fungiert. [2, 6]

3.3.2 Entscheidungsbäume

Beschreibung

Bei den Entscheidungsbäumen läuft die Klassifikation aufgrund von Einteilungen der Objekte anhand von Baumstrukturen ab und so kann eine hierarchische Klassenordnung erzeugt werden. Die einzelnen Knoten repräsentieren die Klassen, welche zu Beginn leer sind oder enthalten diverse Tests, die einem bestimmten Attribut zugeordnet sind. Doch in der Praxis sind die Attribute der einzelnen Objekte nicht eindeutig zuordenbar und so kann es vorkommen, dass manche Objekte mehrfach in verschieden Klassen vorhanden sind.
Um ein Objekt mit dieser Methode zu klassifizieren, wird bei der Wurzel des Baumes begonnen und dann jedes Attribut pro Schritt durchgegangen, bis die gesamten Attribute in Klassen eingeteilt sind. Dies wird wiederholt bis die Objekte mit deren Attribute den richtigen Blättern zugewiesen sind.[2, 6]

Algorithmus

Es gibt nicht nur einen Algorithmus für Entscheidungsbäume, sondern nur eine Richtlinie wie so ein solcher Algorithmus auszusehen hat und folgt meistens einem generalisierten Schema:

1. Es wird definiert, dass die Wurzel (hier als K bezeichnet) und der dazugehörige Baum (bezeichnet als B) mit der Menge der Trainingsdaten (hier T) die Ausgangssituation bildet.

2. Es wird das Attribut (hier A_i) einem Test zugeordnet, welcher die Testmenge am besten in Objekte aufteilt und daraus die Teilmengen generiert (hier $T_1, \ldots, T_m$).

3. Die ganze Testmenge wird nach dem ausgewählten Test auf die Teilmengen hier T_j aufspaltet und daraus ein Knoten (K_j) als Unterordnung vom Wurzelknoten generiert.

4. Für alle abhängigen Knoten, welche alle derselben Klasse angehören, wird ein Blatt im Baum erzeugt. Andernfalls wird weiter rekursiv durch den Baum gegangen und weiter aufgeteilt bis keine Zuordnung mehr möglich ist

Die Aufspaltung in die Teilmengen geschieht nach dem Prinzip eines Tests, welcher am besten die Testdaten aufspaltet. Damit stellt sich die Frage wie die Qualität von einem solchen Test bewertet werden kann. Der erste Ansatz ist das Prinzip der Reinheit der Daten; das bedeutet, dass die Teilmengen nur Objekte von einer Klasse beinhalten. Ein gutes Maß ist die Entropie, die angibt wie groß die Unordnung in einer Menge ist.[2, 6]
Die Formel lautet:

$$entropie(T) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

p_i ist die Wahrscheinlichkeit mit der ein Objekt, welches einer Teilmenge angehört in einer Klasse vorhanden ist. Diese Wahrscheinlichkeit lässt sich anhand der Trainingsdaten abschätzen:

$$p_i = \frac{|\{x \in T | x \in C_i\}|}{|T|}$$

Mit Hilfe der Berechnung der Entropie kann nun ein bedeutenderes Maß berechnet werden, welches sich *Informationsgewinn (gain)* nennt und die Abnahme der Entropie während eines ganzen Teilungsschrittes beschreibt. Die Formel dafür lautet:

$$gain(T, A) = entropie(T) - \sum_{i=1}^{k} \frac{|T_j|}{|T|} entropie(T)$$

Durch die Berechnung von dem *gain* kann nun bestimmt werden wie der Algorithmus die Testdaten aufteilt. Durch Verfeinerung der Methode (*gain*) kann durch die *gain ratio* eine Kombination aus *split info* und dem *gain* wie folgt mittels Formeln beschrieben werden:

1. $$split\ info(T, A) = -\sum_{i=1}^{k} \frac{|T_j|}{|T|} \log_2 \frac{|T_j|}{|T|}$$

2. $$gain\ ratio(T, A) = \frac{gain(T, A)}{split\ info(T, A)}$$

Um gewisse begünstigte Aufteilungen zu verbieten wird für die *gain ratio* ein Schwellenwert festgelegt.
Es existiert neben der *gain ratio* auch ein weiteres Maß, welches sich *Gini-Index* nennt. Dieses Maß ist einfach zu bestimmen und liefert dennoch gut vergleichbare Ergebnisse. Dieser Index sich lässt nach einer einfachen Formel bestimmen:

$$gini(T) = 1 - \sum_{i=1}^{k} p_i^2$$

Mit Hilfe dieser Formel lässt sich der *Gini-Index* für die Gesamtheit der Klassen berechnen.Die einzelnen Partitionen werden als T_n bezeichnet. Hilfreich ist diesbezüglich folgende Formel:

$$gini(T_1, \ldots, T_m = \sum_{i=j}^{m} \frac{|T_j|}{T}\ gini(T_i)$$

Nachdem mit den oben beschriebenen Strategien ein geeigneter Entscheidungsbaum aufgebaut worden ist, können alle Objekte anhand deren Attribute einer bestimmten Klasse zugeordnet werden. [2, 6]

Overfitting

Ein Entscheidungsbaum kann mithilfe der Trainingsdaten korrekt aufgebaut werden. Aber es ist möglich, dass neue Daten nicht mehr vollständig klassifiziert werden können. Dadurch verschlechtert sich die Güte der Daten und es entsteht dann ein weniger komplexer Baum. Dieser Effekt wird dann als *Overfitting* bezeichnet. [6]

Fast alle Algorithmen implementieren eines der beiden Verfahren, mit welchen das *Overfitting* reduziert werden kann:

- Der Algorithmus wird vorher schon gestoppt, bevor ein *Overfitting* zustande kommen kann. So wird vermieden, dass die Klassifikationsgüte zu stark verschlechtert wird. Dieser Vorgang wird als *pre-pruning* bezeichnet und ist weniger verbreitet, da es schwer ist zu bestimmen wann der Aufbau des Entscheidungsbaums gestoppt werden muss.

- Der Entscheidungsbaum wird fertig aufgebaut und dann vereinfacht, indem Knoten durch Blätter ersetzt werden. Diese Methode wird als *post-pruning* bezeichnet und ist einfacher zum Ausführen, da bereits ein Entscheidungsbaum vorliegt.

[6]

Zusammenfassung:

Die Entscheidungsbäume sind eine komfortable Methode um Daten zu klassifizieren, da deren Aufbau einfach gestaltet ist. Die meisten Algorithmen sind binär ausgeführt, da im Zusammenhang mit Entscheidungsbäumen jede Wurzel zwei Kindknoten oder Blätter besitzt. Die Aufteilung in Trainingsdaten und Testdaten erfolgt vor der eigentlichen Berechnung. Diese Aufteilung ist das Grundkonzept von Entscheidungsbäumen. Dabei werden verschiedene Methoden implementiert wie oben bereits beschrieben.
Der eigentliche entscheidende Schritt dabei ist das Pruning, welches den Baum so optimiert, das die Güte der Klassifikation ausreichend ist. Ein aussagekräftiges Kriterium ist auch die Anzahl der Attribute welche ein Objekt besitzt. Es ist dabei wichtig dass der Entscheidungsbaum mit genügend Trainingsdaten aufgebaut wird, da sonst die Güte darunter leidet.
[2]

3.3.3 (k)-Nächste-Nachbarn-Klassifikatoren

Beschreibung:

Diese Methode ist auch bei Clustering bekannt und macht sich die Distanzberechnung der einzelnen Objekte zu Nutze. Durch die *Nähe* der einzelnen Nachbarn werden die einzelnen Objekte bestimmten Klassen zugeordnet. [2, 6]

Algorithmus

Die Methode beschreibt das Verfahren bei der unbekannte Objekte anhand der Distanz zu den Trainingsobjekten einer Klasse zugeordnet werden können. Mit dieser Formel lässt sich die Vorgangsweise beschreiben:

$$c(x) = c\left(\min_{y \in T} dist(x, y) \right)$$

Dabei bezieht sich das $dist(x, y)$ auf die Euklidische Distanz, welche sich

nach folgender Formel berechnet:

$$\sqrt{\sum_{i=1}^{1}(x_i - y_j)^2}$$

Eine andere Möglichkeit ist nicht nur die unmittelbaren Nachbarn sondern auch weiter entfernte Nachbarn für die Berechnung heranzuziehen und dadurch die Qualität zu steigern. Dabei wird ein unbekanntes Objekt einer Klasse zugeordnet. Die benachbarten Objekte gehören der jeweiligen Klasse an. Die Nächsten Nachbarn werden als y_n bezeichnet und können durch folgende Formel berechnet werden:

$$c(x) = \max_{C_i \in C} \sum_{j=1}^{k} \delta(C_i, c(y_i))$$

[2, 6]

Zusammenfassung:

Bei den (k)-Nächste-Nachbarn-Klassifikatoren werden Objekte anhand von Nachbarschaften bestimmten Klassen zugeordnet. Dabei sind die einzelnen Klassifikationsschritte unabhängig voneinander, da sich Nachbarn gegenseitig nicht beeinflussen. Die Wahl der richtigen Distanzfunktion ist entscheidend wie gut der Algorithmus arbeitet.

3.4 Zusammenfassung und Ausblick

Die Klassifikation ist neben den Clustering eine wichtige Methode Objekte in Klassen einzuteilen. Bei der Klassifikation werden die Daten mithilfe verschiedener Methoden in Trainingsdaten und Testdaten aufgeteilt und anhand von den Testdaten eindeutig einer Klasse zugeordnet. Das wichtigste bei der Klassifikation ist die richtige Wahl der Anzahl von den Trainingsdaten. Nur so ist sichergestellt, dass eine ausreichende Klassifikationsgüte erreicht wird.
In Zukunft wird sich an den hier vorgestellten Prinzipien wenig ändern, da diese effizient und auch performant sind. Daher müssen in Zukunft neue noch bessere Algorithmen gefunden werden, welche noch größere und noch komplexere Daten klassifizieren können.

Kapitel 4

Beispiele

4.1 Tools

In den nachfolgenden Beispielen kam das HeuristicLab als Tool zur Anwendung, welches bei der Berechnung komplexer heuristischer Daten Anwendung findet. Diese Software wurde von der FH Hagenberg entwickelt und steht kostenfrei zur Verfügung. Bei den Tests wurde diese Software zum Berechnen von Klassifikationen und Clustern genutzt um die verschiedenen Algorithmen zu vergleichen. Die Testdaten liegen in Format *.csv* vor und sind durch einen ; getrennt.

4.2 Klassifikation

4.2.1 Einführung

Nachfolgend wird die Klassifikation anhand mehrerer Datensätze gezeigt, dabei wird der *Nearest Neigbour* Algorithmus verwendet. Alle Datensätze wurden vom *UCI Repository* herangezogen und beinhalten jeweils die notwendigen Daten wie der Klassenzuordnung. Es werden immer die Datensätze anhand der *Confusion Matrix* bewertet, welche angibt wie gut der Klassifikator ist. [9]

4.2.2 Iris Datensatz

Dieser Datensatz ist der bekannteste auf dem Gebiet von Data Minig und der Klassifikation und wird zum Testen von Erkennungsmerkmalen von zusammengehörigen Formen verwendet. Daher ist dieser Datensatz auch zum Testen für die Klassifikation geeignet. [9]

Eingabedaten

- Klassen: 3

- Datengröße: 150

Parameter

- Train: 33% Test: 67%
- K: 3

Ergebnis

	Actual Class 0	Actual Class 1	Actual Class 2
Predicted Class 0	17	0	0
Predicted Class 1	0	15	1
Predicted Class 2	0	1	16

Abbildung 4.1: Confusion Matrix

Aus der Klasseneinteilung lässt sich die Genauigkeit bestimmen, welche hier 96% beträgt. Der *gini-index* daher ist 0,98 .

4.2.3 Wisconsin Diagnostic Breast Cancer Datensatz

Die Werte stammen aus einem digitalen Bild und beschreiben die Beschaffenheit der einzeln Zellen im Gewebe welche auf dem Bild zu sehen sind. [9]

Eingabedaten

- Klassen: 2
- Datengröße: 569

Parameter

- Train: 33% Test: 67%
- K: 3

Ergebnis

	Actual Class 0	Actual Class 1
Predicted Class 0	112	11
Predicted Class 1	5	61

Abbildung 4.2: Confusion Matrix

Aus der Klasseneinteilung lässt sich die Genauigkeit bestimmen, welche hier 91.5% beträgt. Der *gini-index* daher ist 0,86 .

4.2.4 Parkinsons Datensatz

Der Datensatz stammt von 31 Personen, bei denen die Stimmen gemessenen worden sind, diesbezüglich hatten 23 die Krankheit Parkinson. [9]

Eingabedaten

- Klassen: 2
- Datengröße: 195

Parameter

- Train: 33% Test: 67%
- K: 7

Ergebnis

	Actual Class 0	Actual Class 1
Predicted Class 0	7	3
Predicted Class 1	17	38

Abbildung 4.3: Confusion Matrix

Aus der Klasseneinteilung lässt sich die Genauigkeit bestimmen, welche hier 69.2% beträgt. Der *gini-index* ist daher 0,81.

4.2.5 Fazit

Aus den Ergebnissen kann ableitet werden, dass es bei der Klassifikation wichtig ist, das richtige Verhältnis zwischen Trainings- und Testdaten zu finden. Dies ist stark ausschlaggebend für die Qualität der Ergebnisse. Weiters ist auch zu beachten, dass das Verhältnis zwischen Datenreihen und Klassen entspricht, da sonst es zu Misklassifikationen kommen kann und dadurch zu einer Verschlechterung der Güte. Derzeitiger Stand ist, dass es keine wirkliche Regel für das Verhältnis zwischen Klassen und Anzahl der Daten gibt. Trotzdem sollte von einem relativ ausgeglichen Verhältnis ausgegangen werden.

4.3 Clustering

4.3.1 Einführung

Das Clustering wird anhand des *k-means* Algorithmus gezeigt. Da ein Vergleich zwischen Clustering und Klassifikation erfolgen sollte, wurden herbei dieselben Datensätze verwendet. Der Unterschied zur Klassifikation ist, dass dabei die Klassendefinition weggelassen wird und stellt daher eine unüberwachte Klassifikation dar. Bei der Auswertung werden die Cluster gezeigt und die Zugehörigkeit gebildet. [9]

4.3.2 Iris Datensatz

- Datengröße: 150

Parameter

- K: 3
- Wiederholungen: 0

Ergebnis

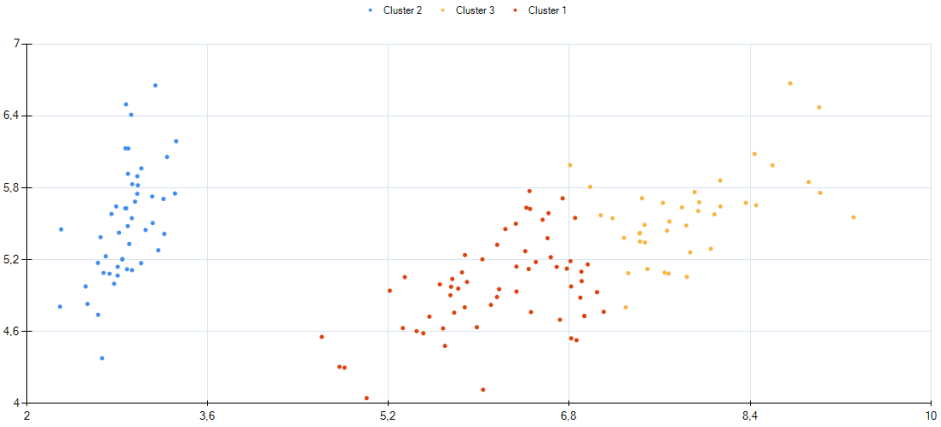

Abbildung 4.4: Cluster

Wie zu erkennen ist gibt es wenige Ausreißer, da die Daten ursprünglich von einem Datensatz bezüglich Klassifikation stammen und bereits in Klassen dargestellt worden sind. [9]

4.3.3 Wisconsin Diagnostic Breast Cancer Datensatz

- Datengröße: 569

Parameter

- K: 3
- Wiederholungen: 0

Ergebnis

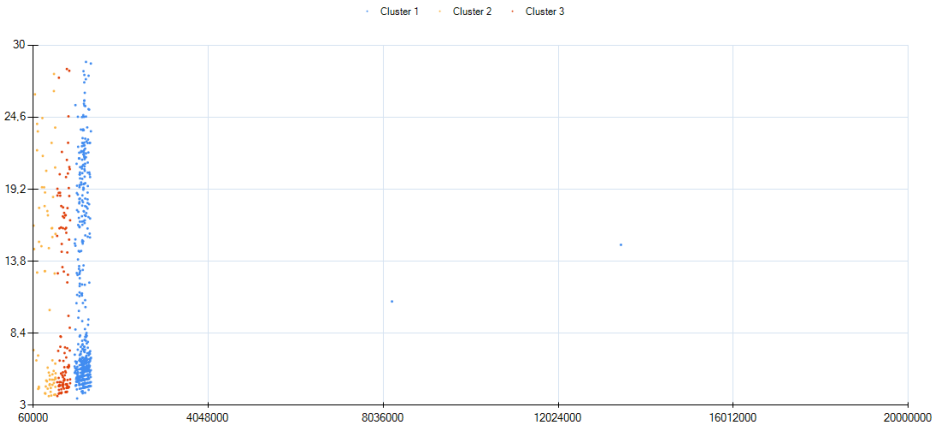

Abbildung 4.5: Cluster

Bei diesen Datensatz ist zu erkennen, dass es Ausreißer gibt, diese werden trotzdem wegen der *Ähnlichkeit* einiger Merkmale einem Cluster zugeordnet. [9]

4.3.4 Parkinsons Datensatz

- Datengröße: 195

Parameter

- K: 2
- Wiederholungen: 0

Ergebnis

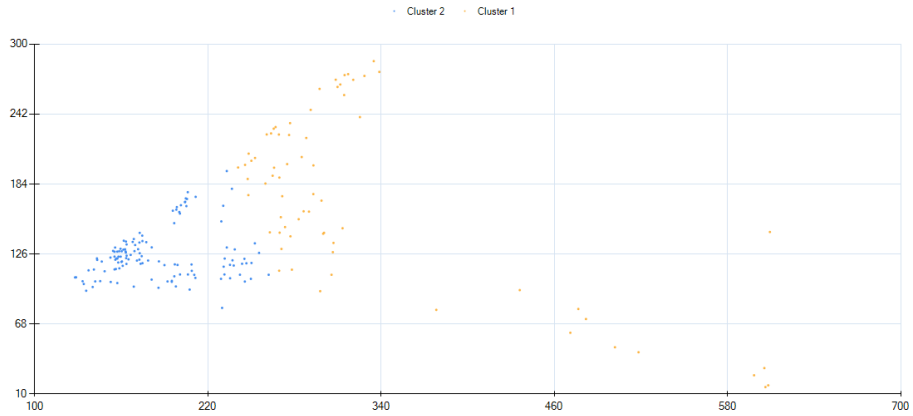

Abbildung 4.6: Cluster

Durch die Wahl von von zwei Clustern kann eine gute Klassifikation dargestellt werden, da die zwei vorgegeben Klassen relativ exakt dargestellt werden können
[9]

4.3.5 Fazit

Beim Clustering ist klar zu erkennen dass es Gemeinsamkeiten mit der Klassifikation gibt, da die meisten Clustereinteilungen die wahren Klassen voraussagen. Daher ist das Clustering ebenso bedeutend wie die Klassifikation. Beim Clustering kommt es auf die vorgegebene Clusteranzahl an um gute Ergebnisse zu liefern.

Kapitel 5

Schluss

5.1 Zusammenfassung

In dieser Arbeit wurden die Themen Clustering und Klassifikation behandelt. Ziel war es die verschieden Algorithmen aufzuzeigen und zu vergleichen. Grundlegend wird von Klassifikation ausgegangen, die sich in eine überwachte und unüberwachte Klassifikation einteilen lässt. Bei der unüberwachten Klassifikation kann auch von Clustering gesprochen werden, da es hier keine eindeutige Klasseneinteilung gibt. Im Gegensatz dazu ist bei der klassischen Klassifikation die Einteilung bzw. die Zuordnung bekannt. Doch arbeiten beide Verfahren nach demselben Prinzip. Umfangreiche Daten werden in verwendbare Teile eingeteilt.
Die beiden Methoden spielen in der heutigen Welt eine große Rolle und werden häufig in der Datenanalyse sowie im Bereich des Data Mining verwendet. Daher sind unterschiedliche Algorithmen implementiert worden, um dieses Problem zu lösen. Dabei kommt es auf die verwendeten Daten an, welcher Algorithmus die besten Ergebnisse liefert.
Abschließend wird zusammengefasst, dass in dieser Arbeit nicht alle Algorithmen beschrieben worden sind. In dieser Arbeit wurden nur die bedeutsamsten und die relevanten Algorithmen aufgelistet. Die Algorithmen werden nicht nur für Clustering und Klassifikation verwendet, sondern auch in Bereichen wo Datenanalysen und Heuristiken eine wesentliche Rolle spielen.

Quellenverzeichnis

Literatur

[1] Charles J. Alpert und So-Zen Yao. „Spectral Partitioning: The More Eigenvectors, the Better“. In: *Proceedings of the 32Nd Annual ACM/IEEE Design Automation Conference*. DAC '95. ACM, 1995, S. 195–200.

[2] Leo Breiman u. a. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, 1984.

[3] Daniel Fasulo. *An Analysis of Recent Work on Clustering Algorithms*. 1999.

[4] Usama Fayyad, Cory Reina und P. S. Bradley. „Initialization of Iterative Refinement Clustering Algorithms“. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. KDD'98. AAAI Press, 1998, S. 194–198.

[5] A. K. Jain, M. N. Murty und P. J. Flynn. „Data Clustering: A Review“. *ACM Comput. Surv.* 31.3 (Sep. 1999), S. 264–323.

[6] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., 1997.

[7] Dan Pelleg und Andrew Moore. „Accelerating Exact K-means Algorithms with Geometric Reasoning“. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '99. ACM, 1999, S. 277–281.

[8] P. Tamayo u. a. „Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation.“ *Proceedings of the National Academy of Sciences of the United States of America* (1999).

[9] *UCI Repository*. die Datensätze stammen von deren Webseite.

[10] Sholom M. Weiss und Casimir A. Kulikowski. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann Publishers Inc., 1991.