

Dominik Brunmeir

DECISION MAKING IN AMBIENT ASSISTED LIVING ENVIRONMENTS BASED ON UNCERTAIN AND FUZZY DATA



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Technische Universität Wien

Institut für Analysis und Scientific Computing

Head: Univ.Prof. Dipl.-Ing. Dr.techn. Anton Arnold

Supervisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker

Supervisor: Univ. Lektor Dipl.-Ing. Dr. Johannes Kropf

Vienna, September 2015

ERKLÄRUNG

Ich erkläre, dass die vorliegende Diplomarbeit von mir selbst verfasst wurde und ich keine anderen als die angeführten Behelfe verwendet bzw. mich auch sonst keiner unerlaubter Hilfe bedient habe. Ich versichere, dass ich diese Diplomarbeit bisher weder im In- noch im Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe. Weiters versichere ich, dass die von mir einereichten Exemplare (ausgedruckt und elektronisch) identisch sind.

.....011.09.2015.....
(date)

.....
(signature)

DANKSAGUNG

Diese Diplomarbeit wurde durch das AIT Austrian Institute of Technology ermöglicht.

Weiters möchte ich mich besonders bei meinen Betreuern Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker sowie Dipl.-Ing. Dr. Johannes Kropf bedanken, denn ohne sie wäre diese Diplomarbeit niemals fertig geworden. Vielen Dank für Eure Geduld und Unterstützung.

ZUSAMMENFASSUNG

Diese Diplomarbeit behandelt die Anwendung von Datamining-Techniken auf Fuzzy Daten im AAL Kontext. Es wurde sowohl ein Empfehlungssystem als auch ein System zur Entscheidungsfindung erforscht und implementiert. Diese Systeme sind Teil des *Reasoning* Systems einer AAL Anwendung und versuchen gesammelte Informationen zu verknüpfen und daraus Schlüsse zu ziehen.

Der erste Teil dieser Diplomarbeit beschäftigt sich hauptsächlich mit Fuzzy Clustering und Fuzzy Control, da es ein naheliegender Zugang ist, um menschliche Verhaltensweisen zu beschreiben. Zuerst wird eine Einführung in die Theorie des Clustering behandelt, danach die Grundlagen von Fuzzy Control erläutert um schlussendlich das *Adaptive Online Fuzzy Inference System (AOFIS)* vorzustellen und zu analysieren. *AOFIS* ist ein Fuzzy Control System, das menschliches Verhalten lernen und in der Folge auch imitieren soll. Es beruht auf einer Doppelclustering Technik, die aus Daten interpretierbare Zugehörigkeitsfunktionen extrahiert um mit diesen ein *if-then*-Regelsystem aufzubauen.

Der zweite Teil behandelt mehrere Ansätze um Empfehlungssysteme zu erstellen, wobei hier ein Hauptaugenmerk auf die Aufbereitung von Daten gelegt wurde, da in diesem Zusammenhang oft große Mengen von Daten anfallen. Weiters wurde ein solches Empfehlungssystem implementiert und mit einfachen Daten getestet. Es beruht auf Fuzzy Clustering und ist ein hybrides System. Dieses Empfehlungssystem soll in der *YouDo it! - we help* Platform integriert werden um dem Benutzer Vorschläge für verwandte Themen und Filme geben zu können.

ABSTRACT

This Thesis analyses the application of data-mining techniques on fuzzy data in an AAL context. The goal was to research decision making and recommender systems as part of the *Reasoning* subsystem of an AAL service. It tries to relate information and deduce valid behavioural patterns.

The first part of the thesis explores fuzzy clustering and fuzzy control as a mean to describe human behaviour. After a theoretical introduction to clustering and fuzzy control, the *Adaptive Online Fuzzy Inference System (AOFIS)* is presented and analysed. AOFIS tries to learn from human behaviour and imitate it after a certain learning period. It is based on a double-clustering technique, which extracts interpretable fuzzy granules. This granules are used to build an *if-then*-Rulebase as part of an inference system as described in the fuzzy control chapter.

The second part of the thesis presents theoretical foundations of recommender systems and several popular techniques to process large amounts of data. It is a necessity to reduce the runtime of such a system and thus improve the performance. Furthermore a fuzzy recommender system was implemented for use as a module in the *YouDo it! - we help* platform as a way to present the user similar videos and topics to the already watched or rated content. This recommender system is a hybrid system and makes use of fuzzy clustering methods to combine the advantages of content-based and collaborative filtering approaches.

INHALTSVERZEICHNIS

1 Einführung	1
1.1 Motivation	1
1.2 Ambient Assisted Living	2
1.3 Techniken von AAL	4
1.4 Umfang und Ziele	5
1.5 Verwandte Arbeiten	5
2 Clustering	9
2.1 Ähnlichkeitsmaße	9
2.2 Kategorien von Clustering Algorithmen	13
2.3 Clustering Algorithmen	16
3 Fuzzy Control	23
3.1 Unscharfe Menge	23
3.2 Unscharfe Logik	27
3.3 Fuzzy-Regelung	28
4 Adaptive Online Fuzzy Inference System	33
4.1 Sammeln von Daten	34
4.2 Extraktion von Zugehörigkeitsfunktionen	35
4.3 Extraktion von Fuzzy Regeln	38
4.4 Anwenden der Fuzzy Regeln	39
4.5 Adaptieren der Ausgabe	40

4.6	Implementierung	41
4.7	Ergebnisse und Ausblick	45
5	Empfehlungssysteme	49
5.1	Collaborative filtering	49
5.2	Content-based	50
5.3	Hybride Verfahren	51
5.4	Techniken zum Datamining	51
6	Empfehlungssystem für <i>YouDo it - we help!</i>	57
6.1	<i>YouDo it - we help!</i>	57
6.2	User Adaption Module	57
6.3	Daten	58
6.4	Content Based Filtering	59
6.5	Collaborative Filtering	60
6.6	Hybrides Verfahren	61
6.7	Implementierung	61
6.8	Ergebnisse und Ausblick	63

1. EINFÜHRUNG

Der Inhalt dieser Diplomarbeit behandelt Techniken, um einerseits automatisiert Entscheidungen auf Basis von gesammelten Aktivitätsdaten eines Benutzers zu fällen und andererseits Empfehlungen aufgrund gesammelter Präferenzen und Kategorien abzugeben. Die hierbei gesammelten Daten werden durch Clustering aufbereitet um so Aussagen treffen zu können. Die im Rahmen der Diplomarbeit erstellten Programme sollen später in HOMER [1] bzw. in die YOUDO [27] Plattform einfließen.

1.1. MOTIVATION

Angesichts einer schnell alternden Gesellschaft, ist es notwendig bereits jetzt unterstützende Maßnahmen zu erforschen und zu entwickeln, die es einerseits erlauben Personen länger selbstständig und selbstbestimmt leben zu können [3] [21] und andererseits die Effizienz der Pflege zu erhöhen, um bei gleichbleibender Anzahl von Ressourcen mehr Leute zu betreuen [20]. *Ambient Assisted Living (AAL)* versucht sich diesen Problemen anzunehmen und Lösungen zu entwickeln, die unter Anderem in den Haushalten selbst unterstützend wirken sollen und so dazu führen, dass Menschen länger in diesen verbleiben können. Außerdem sollen diese Techniken auch den Pflegediensten zur Verfügung stehen und durch ihre unaufdringliche Art (*ambient*) zu einer besseren Abschätzung des tatsächlichen Pflegebedarfs führen [20].

1.2. AMBIENT ASSISTED LIVING

Der Kernpunkt von *AAL* ist die Erhöhung der Lebensqualität von älteren Personen, was nicht nur körperliche und geistige Gesundheit bedeutet, sondern auch die Einbettung in ein funktionierendes soziales Umfeld, wobei das technische System selbst in den Hintergrund tritt. Um dieses zwar junge, aber doch umfangreiche Gebiet abzudecken bietet sich an, es in Teilgebiete zu unterteilen [31]:

- *Ageing well at home*
- *Ageing well in the community*
- *Ageing well at work or “active ageing at work”*

Diese stellen verschiedene Herausforderungen an die technischen Systeme, sind aber für ein ganzheitliches AAL Konzept miteinander zu betrachten. Während bei ersterem das Ziel ist, das Leben im eigenen Haushalt zu erleichtern, ist bei zweitem der Aspekt der Interaktion mit anderen im Vordergrund um unter anderem das Problem der zunehmenden Isolation von älteren Personen in ländlichen Gegenden zu reduzieren. Letzteres ist das Bestreben ältere Personen länger am Arbeitsprozess integrieren zu können. Alle Systeme sollten aber jedenfalls folgende Voraussetzungen erfüllen [31]:

- Eingebettet
- Verteilt
- Personalisiert
- Adaptiv

- Vorausschauend

Die notwendigen Geräte für assistierende Maßnahmen sollten weitgehend nicht wahrnehmbar sein. Dies dient dazu die Person nicht zu beeinträchtigen, aber auch um die Umgebung der Person nicht zu verändern. Außerdem wird die Akzeptanz solcher Systeme erhöht, wenn diese nicht direkt sichtbar sind. [20] Dies wird durch die fortschreitende Miniaturisierung von Sensoren und Mikrochips, sowie der Integration in Wänden, Möbeln und Haushaltsgeräten erleichtert. Die Systeme sollten ubiquitär sein um in jeder Phase des Lebens zur Verfügung zu stehen. Der Einbau in Möbel oder ähnliche Gegenstände des täglichen Lebens ist hierbei wünschenswert, da es vielfältige Möglichkeiten bietet um Aktivitäten des täglichen Lebens zu messen. Weiters sollten alle solchen Systeme auf den jeweiligen Benutzer zugeschnitten sein damit sich der Benutzer besser mit diesen identifizieren kann und genau die Hilfestellung leistet, die die Person benötigt. Adaptiv bedeutet einerseits, wenn sich nun die Verhaltensmuster des Benutzers ändern, sollte das System in der Lage sein, sich daran anzupassen, andererseits auf das ad-hoc Verhalten des Benutzers reagieren können. Und letztlich sollte es für das System möglich sein, Wünsche und Bedürfnisse des Benutzers zu erkennen, bevor diese von der Person aktiv ausgedrückt werden [31].

BENUTZERFREUNDLICHKEIT

Ein wichtiger Faktor für die Akzeptanz von AAL ist die Benutzerfreundlichkeit. Es hat sich gezeigt, dass Systeme mit geringerer Interaktionszeit zu einer höheren Akzeptanz seitens des Benutzers führen [20]. Bei notwendiger Interaktion muss aber sowohl auf Bedürfnisse des Benutzers eingegangen werden, aber auch die Art

der Benutzeroberfläche den Umständen, wie dem Ausgabegerät, angepasst werden. Hierzu gibt es Ansätze, die auch eine automatisierte Anpassung an die jeweilige Gerätekategorie (PC, TV, Smartphones) erlauben. *AALuis* [26] ist ein solches System, das versucht Personen ohne Erfahrung mit digitalen Benutzeroberflächen, aber auch technikaffinen Benutzern gleichermaßen Zugang zu AAL Diensten zu ermöglichen. Hierzu wird eine zusätzlich Abstraktionsschicht eingeführt, welcher über mehreren Transformationsschritten die gewünschte, auf das jeweilige Ausgabegerät zugeschnittene Oberfläche erzeugt.

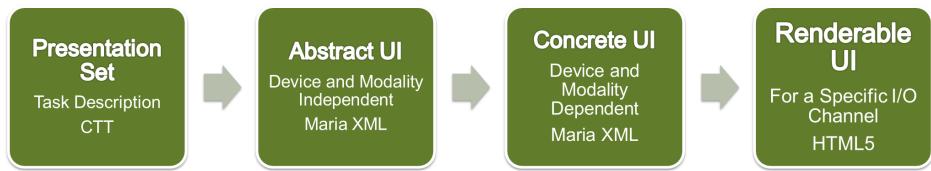


Abbildung 1: AALuis Ablauf

1.3. TECHNIKEN VON AAL

AAL Technologien lassen sich in folgende Kategorien unterteilen [31]:

- *Sensing*: Dies beinhaltet alle Techniken bezüglich Sensoren oder Aufzeichnung von Informationen, wie zum Beispiel Smart Home Sensoren
- *Reasoning*: Um gesammelte Informationen verknüpfen und verarbeiten zu können, sowie Schlüsse aus diesen zu ziehen, werden verschiedene Techniken angewandt. Diese können regelbasiert, aber auch probabilistisch sein.
- *Acting*: Aktuatoren und Robotik fallen in diese Kategorie
- *Interacting*: Interaktion mit dem Benutzer ist eine Kernkomponente. Dies beinhaltet Apps, Schnittstellen und Oberflächen.

- *Communicating*: Alle Technologien, die ein effiziente Kommunikation zwischen den einzelnen Geräten ermöglichen. Damit sind diverse Protokolle und Netzwerke gemeint.

1.4. UMFANG UND ZIELE

Diese Diplomarbeit beschäftigt sich hauptsächlich mit dem *Reasoning* im AAL Kontext, also dem Verknüpfen von Informationen sowie dem Ziehen von Schlüssen aus diesen. Es werden hierfür Methoden des Datamining sowie von Fuzzy Systemen benutzt. Das Ziel hierbei ist minimale Interaktion mit dem Benutzer zu benötigen, allerdings dem Benutzer stets die Möglichkeit zu geben, das Verhalten des Systems aktiv zu verändern und anzupassen [9]. In der Folge werden zwei verschiedene Techniken vorgestellt, wie diese Methoden in einem AAL Kontext verwendet werden können. Einerseits wird eine *Decision Making Engine* auf Basis von *AOFIS* vorgestellt [8]. Dies basiert auf dem Prinzip des Fuzzy Clustering und ist mit einem Fuzzy System verwandt. Andererseits wird ein einfaches *Empfehlungssystem* vorgestellt, wo ähnliche Techniken eingesetzt werden. Beide Systeme versuchen den Alltag des Benutzers zu Vereinfachen, ohne dass der Benutzer aktiv mit dem System interagieren muss.

1.5. VERWANDTE ARBEITEN

ADAPTIVE-NETWORK-BASED FUZZY INFERENCE (ANFIS)

ANFIS ist ein Inferenzsystem, das einen ähnlichen Anwendungszweck wie AOFIS besitzt. Hierbei werden adaptive Netzwerke erstellt, um *if-then*-Regeln zu erhalten

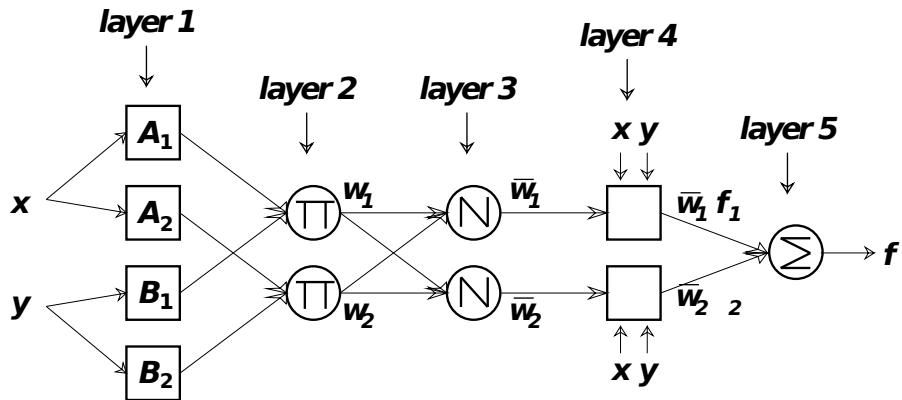


Abbildung 2: Funktionsweise von ANFIS

Wenn x und y zwei Eingabewerte sind, werden im ersten Schritt Zugehörigkeitsfunktionen erstellt, im zweiten Schritt diese multipliziert. Danach wird das Verhältnis zwischen der Stärke der gefeuerten Regel und der Stärke der Summe aller Regeln berechnet (Layer 3). Im vierten Schritt wird der Output berechnet und schlussendlich aufsummiert um die endgültige Konsequenz zu erhalten [14].

MOVIELENS

Das MovieLens-Verfahren ist eines der bekanntesten Verfahren, das Empfehlungen für Filme generiert. Es ist ein hybrides Verfahren, welches in mehreren Stufen versucht einerseits Verwandte Benutzer zu ermitteln und mit Hilfe von umfangreichen Trainingsdaten die besten Filme ermittelt. [11]

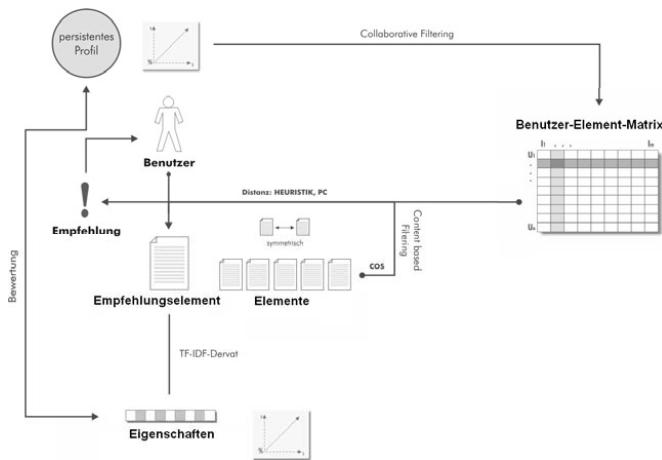


Abbildung 3: Schematische Darstellung des MovieLens Empfehlungssystem

2. CLUSTERING

Cluster Analysis versucht Zusammenhänge oder Ähnlichkeiten aus Daten zu extrahieren und zu Strukturen zusammenzufassen, die wir als *Cluster* bezeichnen. Es ist eine Methode die einerseits versucht noch nicht kategorisierte Daten einzufügen, aber auch den Informationsgehalt von gegebenen Daten zu vereinfachen [29], wobei die Daten in einem Cluster so ähnlich wie möglich zueinander sind, aber der Unterschied zwischen zwei Clustern so groß als möglich [4]. Clustering ist eines der Grundkonzepte von *maschinellen Lernen (machine learning)* sowie von *Datamining* [23]. Um eine sinnvolle Anwendung von Clustering zu ermöglichen, müssen hier zwei Entscheidungen gefällt werden [6]:

- Wahl des Ähnlichkeitsmaßes
- Wahl des Algorithmus

2.1. ÄHNLICHKEITSMASSE

Um Ähnlichkeiten zwischen verschiedenen Daten quantifizieren zu können muss ein gewisses Maß dafür eingeführt werden. Es stellt sich die Frage nach einer Minimalanforderung an eine solche Funktion. Ausgehend von Unähnlichkeitsmaßen, also Funktionen die das Maß der Unähnlichkeit quantifizieren, also Metriken sehr ähnlich sind, werden wir ein Ähnlichkeitsmaß definieren.

Um das Maß der Unähnlichkeit zu quantifizieren muss anschaulich gesehen gelten, dass der Abstand zwischen zwei Elementen immer größer 0 sein muss und der Abstand zwischen einem Objekt und sich selbst 0 ist. Weiters sollte sich der Ab-

stand nicht ändern wenn man sie vom ersten Objekt zum zweiten Objekt misst, oder vom zweiten Objekt zum ersten (*Symmetrie*). Wir sehen also, die Voraussetzungen die an eine solche Funktion gestellt werden sind sogar schwächer als die einer *Metrik* oder *Norm*.

Als Beispiel führen wir hier Punkte im zweidimensionalen linearen Raum \mathbb{R}^2 an, bei dem das Maß der Unähnlichkeit als euklidischer Abstand zwischen zwei Elementen x und y , also $d(x, y) = \|x - y\|_2$, definiert ist. Wir sehen sofort, dass $d(x, y)$ immer größer oder gleich 0 ist. Der einzige Fall, der $d(x, y) = 0$ bedeuten würde, ist $x = y$. Der euklidische Abstand erfüllt also die gewünschte Minimalanforderung und ist ein Kandidat für eine solche Distanzfunktion.

Für ein Ähnlichkeitsmaß gelten die gleichen Symmetrieeigenschaften, allerdings soll gelten, dass zwei Elemente ähnlicher sind, je höher der Wert der Ähnlichkeitsfunktion ist. Der höchste Wert soll dabei beim Vergleich von einem Element mit sich selbst erreicht werden [5] [28].

Definition 1 (Unähnlichkeitsmaß). *Sei X ein Vektorraum über einen Körper \mathbb{K} . Die Funktion $d : X \times X \rightarrow \mathbb{R}$ heißt Unähnlichkeitsmaß zwischen x und y , falls $\forall x, y$ aus X gilt:*

$$d(x, y) = d(y, x) \quad (\text{Symmetrie})$$

$$d(x, y) \geq 0 \text{ und } d(x, y) = 0 \Leftrightarrow x = y \quad (\text{Positive Definitheit})$$

Ein Unähnlichkeitsmaß, das die Dreiecksungleichung erfüllt, ist eine *Metrik* [28]. Es gibt also eine Vielzahl von solchen Funktionen, wobei nicht jede Metrik für gegebene Daten gleich gut geeignet ist [5].

Definition 2 (Dreiecksungleichung [28]). *Sei X ein Vektorraum über einen Körper*

\mathbb{K} , $d : X \times X \rightarrow \mathbb{R}$ definiert wie oben, $\forall x, y, z$ aus X gilt:

$$d(x, y) \leq d(x, z) + d(z, y)$$

Um nun vom Abstand zu einer Ähnlichkeitsfunktion zu kommen, definieren wir diese zuerst.

Definition 3 (Ähnlichkeitsmaß [5]). *Sei X ein Vektorraum über einen Körper \mathbb{K} . Die Funktion $s : X \times X \rightarrow [0, 1]$ heißt Ähnlichkeitsmaß zwischen x und y , falls $\forall x, y$ aus X gilt:*

$$s(x, y) = s(y, x) \quad (\text{Symmetrie})$$

$$s(x, y) \leq s(x, x)$$

Wir sehen, dass jedes Unähnlichkeitsmaß, welches auf $[0, 1]$ normiert wurde, als Ähnlichkeitsmaß aufgefasst werden, wenn man $s = 1 - d$ setzt [5].

Als letztes brauchen wir noch den Begriff der Norm.

Definition 4 (Norm [28]). *Sei X ein Vektorraum über einen Körper \mathbb{K} , dann heisst eine eine Abbildung $\|\cdot\| : X \rightarrow \mathbb{R}_0^+$ Norm, wenn folgende Eigenschaften erfüllt sind:*

$$\|x\| = 0 \Rightarrow x = 0$$

$$\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$$

$$\|x + y\| \leq \|x\| + \|y\|$$

Jede Norm ist auf natürliche Weise eine Distanzfunktion indem man folgenden Zusammenhang betrachtet:

$$d(x, y) = \|x - y\|$$

Die am häufigsten verwendeten Ähnlichkeitsfunktionen sind die *Kosinus Ähnlichkeit* sowie die *Pearson Ähnlichkeit*. Als Distanzfunktion wird zumeist der *euclidische Abstand* verwendet [15].

KOSINUS ÄHNLICHKEIT

Diese beschreibt den Winkel zwischen zwei Vektoren und ist folgendermaßen definiert:

Definition 5 (Kosinusähnlichkeit [15]). *Sei $X \subset (\mathbb{R})^n$. $\cos(\theta) : X \rightarrow [-1, 1]$ heisst Kosinusähnlichkeit zwischen den Vektoren $x, y \in X$, wenn*

$$\cos(\theta) = \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

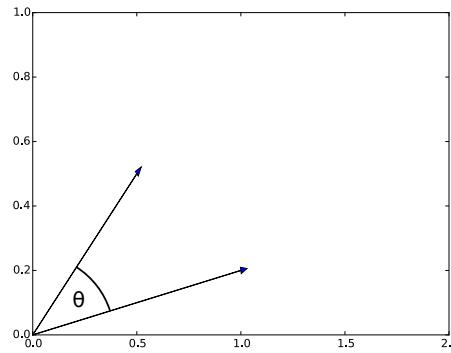


Abbildung 4: Theta

Diese Ähnlichkeitsfunktion hat definitionsgemäß Werte in $[-1, 1]$, für $x, y \in X$.

PEARSONÄHNLICHKEIT

Definition 6 (Pearson Ähnlichkeit [15]). *Sei $X \subset (\mathbb{R})^n$. $s(x, y) : X \rightarrow [-1, 1]$ heisst Pearsonähnlichkeit zwischen den Vektoren $x, y \in X$, wenn*

$$s(x, y) = \frac{\sum_{i=1}^n ((x_i - \bar{x}) \cdot (y_i - \bar{y}))}{\sqrt{\sum_{i=1}^n ((x_i - \bar{x})^2 \cdot \sum_{i=1}^n (y_i - \bar{y})^2)}}$$

\bar{x}, \bar{y} bezeichnet hier $\frac{1}{n} \sum_{i=1}^n x_i$ bzw. $\frac{1}{n} \sum_{i=1}^n y_i$.

2.2. KATEGORIEN VON CLUSTERING ALGORITHMEN

Da sich verschiedene Verfahren zum Clustering oftmals grundlegend unterscheiden und somit gewisse Vorteile aber auch Nachteile mit sich bringen, wurden eine Vielzahl solcher Verfahren entwickelt. Die Auswahl des Verfahrens sollte dem Anwendungszweck entsprechen und ist mitunter nicht einfach [18].

Zuerst werden die Hierarchischen Clusteringverfahren eingeführt, die in der Praxis oft eingesetzt werden und deren Vorteil die Flexibilität bezüglich der Distanzfunktion ist. Weiters kann der Ablauf des Verfahrens sehr einfach durch Dendrogramme visualisiert werden.

Als zweites grundlegendes Verfahren werden die Partitionierenden Clusteringverfahren näher beleuchtet, wo mit Hilfe einer Zielfunktion der Datenraum partitioniert wird, sodass eine Zielfunktion minimal wird. Hier werden Datenpunkte zwischen den Clustern hin- und herbewegt bis ein Minimum erreicht wird.

Schlussendlich werden noch dichtebasierete Methoden erwähnt, bei dem die Daten nach der Dichte ihres Vorkommens zusammengefasst werden.

HIERARCHISCHE CLUSTERINGVERFAHREN

Bei diesem Verfahren wird die Grundmenge rekursiv in entweder immer kleinere Cluster zerlegt oder zu immer größeren Cluster zusammengefasst. Man spricht dabei von entweder *divisiven* oder *agglomerativen* Verfahren. Man erhält dadurch eine Hierarchie von Clustern.

- *Agglomerative Verfahren*: Jedes Element der Grundmenge ist ein Cluster. Diese werden so lange zusammengefasst bis die gewünschte Struktur erreicht

wurde

- *Divisive Verfahren*: Die Grundmenge wird als einziger Cluster aufgefasst.

Dieser wird so oft geteilt, bis die gewünschte Struktur erhalten wurde.

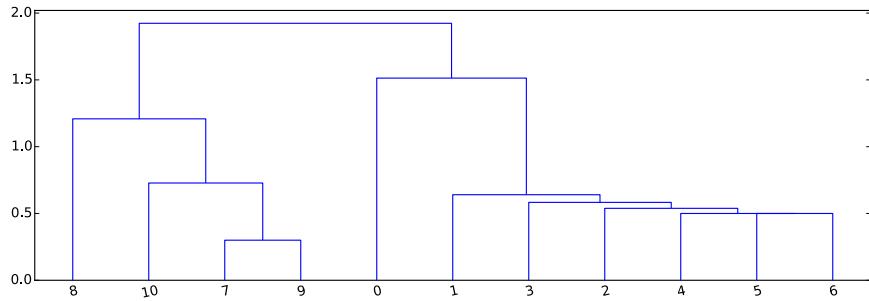


Abbildung 5: Dendrogramm eines Agglomerativen Verfahrens

Wie man in der Abbildung 5 sieht, entsteht ein Binärbaum. Wenn man das Verfahren nach $n - k$ Schritten Abbricht erhält man die gewünschte Anzahl an Cluster. Bei diesen Verfahren kann es zu sogenannten Ketteneffekten (*chaining effects*) kommen, bei dem zwei Cluster durch einzelne Punkte verbunden werden. Außerdem werden Ausreißer nicht erkannt sondern meist als einzelne Cluster behandelt. Weiters sind hierarchische Verfahren strikt nichtlinear bezüglich der Laufzeit. Andererseits haben sie gute Eigenschaften bei streng getrennten Clustern und verhalten sich relativ gut bei verschiedenen Distanzfunktionen. Bekannte Vertreter sind hierbei der *Single Linkage*, *Complete Linkage* sowie *Average Linkage* Clustering Algorithmus, die sich durch die Wahl des Kriteriums zur Vereinigung der Cluster unterscheiden [18] [6] [28] [13].

PARTITIONIERENDE CLUSTERINGVERFAHREN

Bei partitionierenden Clusteringverfahren wird der Raum der Daten in Partitionen aufgeteilt und die einzelnen Elemente der Partitionen herumgeschichtet bis eine optimale Verteilung der Daten vorliegt. Die Optimalität ist durch eine Zielfunktion sichergestellt die meist iterativ minimiert wird. Die meisten Algorithmen dieser Art erfordern die Anzahl der Cluster, die gefunden werden sollen, im vorhinein.

Die bekannteste Familie von Algorithmen aus dieser Klasse, sind die auf *K-Means* basierenden Algorithmen, zu denen auch so genannte *soft*-Varianten wie der *Fuzzy C-Means* Algorithmus gehören. Diese sind allerdings stark abhängig von der initialen Wahl der Clusterzentren sowie empfindlich gegenüber Ausreißern [18] [6] [28] [13] [5].

DICHTEBASIERTE VERFAHREN

Die Idee hinter den dichtebasierten Verfahren ist, dass ein Cluster eine Region ist, in der Daten in höherer Dichte vorkommen, während die nähere Umgebung eine Region ist, mit niedrigerer Dichte. Beim bekanntesten Vertreter, *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)* muss hierzu die Anzahl der Cluster nicht im Vorhinein bekannt sein [29].

FUZZY VS. CRISP CLUSTERING

Üblicherweise wird bei einem Clustering Algorithmus jedem Punkt eindeutig ein Cluster zugeordnet. Man spricht hierbei von *hartem* beziehungsweise *crisp* Clustering. Dem gegenüber stehen die *soft* oder *fuzzy* clustering Algorithmen wie *Fuzzy*

C-Means, bei dem ein Datenpunkt, jedem Cluster zu einem gewissen Grad zugeordnet wird. Man kann allerdings jede weiche Zuordnung intuitiv zu einer harten umwandeln indem man einfach den Cluster mit der höchsten Zugehörigkeit für jeden Datenpunkt wählt [18].

2.3. CLUSTERING ALGORITHMEN

Hier wollen wir eine kurze Auswahl von gebräuchlichen Algorithmen vorstellen und näher erläutern, sowie ihre Vor- und Nachteile etwas beleuchten.

SINGLE-LINKAGE CLUSTERING

Single-Linkage Clustering ist ein agglomeratives hierarchisches Clusteringverfahren, in welchem die einzelnen Daten zu Beginn des Verfahrens in einem separaten Cluster sind. In jedem Schritt werden diejenigen Cluster vereinigt, welche den geringsten Abstand zueinander haben [18] [28].

Sei also $\pi^0 = \{U_0, \dots, U_n\}$ die initiale Partitionierung des Ausgangsraumes (X, d) , wobei d eine Abstandsfunktion ist. Im k -ten Schritt des Clustering definieren wir die Partitionierung als

$$\pi^k = \{U_1^k, \dots, U_{n_k}^k\}$$

Im nächsten Schritt $k + 1$ werden nun diejenigen U_p^k und U_q^k gesucht, die das Selektionskriterium erfüllen. In unserem Fall ist dies der geringste Abstand. U_p^k und U_q^k werden nun vereinigt, also

$$\pi^{k+1} = \{U_1^k, \dots, U_{p-1}^k, U_p^k \cup U_q^k, U_{p+1}^k, \dots, U_{q-1}^k, U_{q+1}^k, \dots, U_{n_k}^k\}$$

Der Algorithmus baut also eine Hierarchie auf, da gilt $\pi^k \prec \pi^{k+1}$, weiters bricht er nach spätestens n Schritten ab. Durch die Hierarchie kann der Ablauf des Algorithmus mit Hilfe eines Dendrogrammes visualisiert werden.

Um die gewünschte Anzahl von Cluster k zu erhalten, bricht man das Verfahren bei $n - k$ Schritten ab. Die Laufzeit des Algorithmus ist $\mathcal{O}(n^2 \log n)$, da die Berechnung der Distanzmatrix, sowie die Suche der Daten mit der geringsten Distanz aufwändig sind. Dies macht den Algorithmus für eine große Anzahl an Daten natürlich unattraktiv [18] [28].

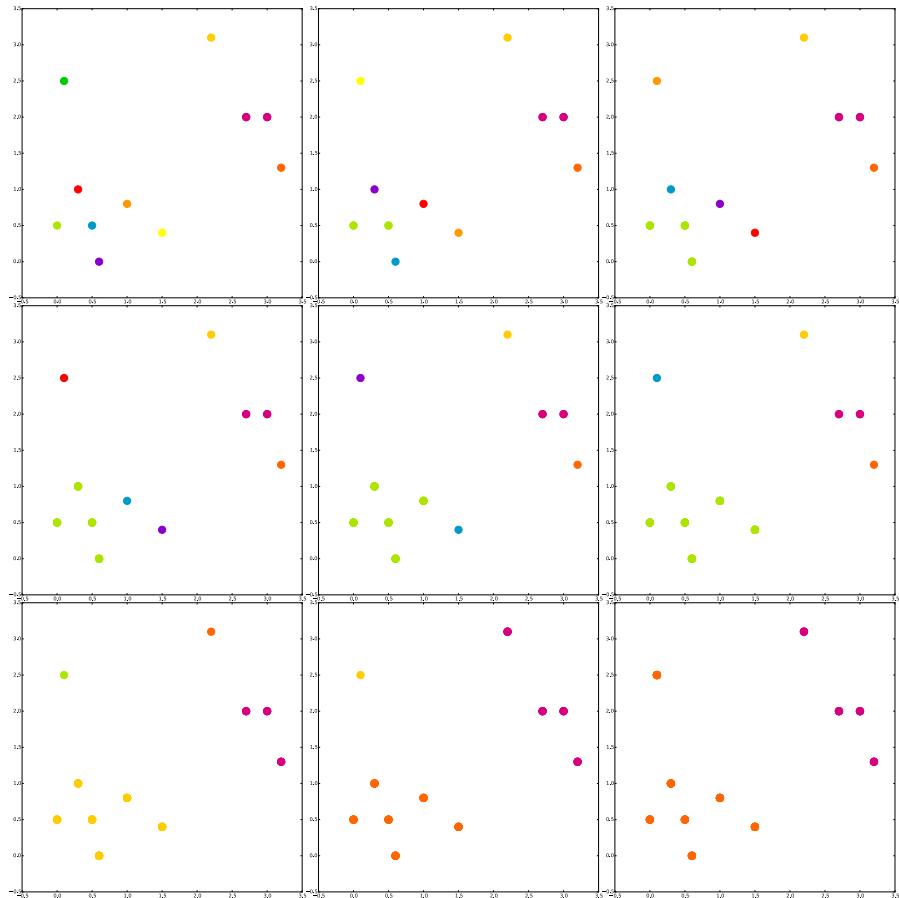


Abbildung 6: Single Linkage Clustering

K-MEANS CLUSTERING

Der k-Means Clusteringalgorithmus ist ein Standardverfahren aus der Familie der partitionierenden Clusteringverfahren. Das Ziel ist es den Datenraum derart zu partitionieren, dass die Summe der Distanzen der Zentren $C = \{c_1, c_2, \dots\}$ der Daten $X = \{x_1, x_2, \dots\}$ in einer Partition S_j minimal wird [18] [28]. Das heisst:

$$J = \sum_{j=1}^k \sum_{x_i \in S_j} d(x_i, c_j)^2 \rightarrow \min$$

Wobei d hier die euklidische Distanz ist. Der Algorithmus konvergiert gegen ein lokales Minimum in finiter Zeit, wie von Selim und Ismail [25] gezeigt wurde.

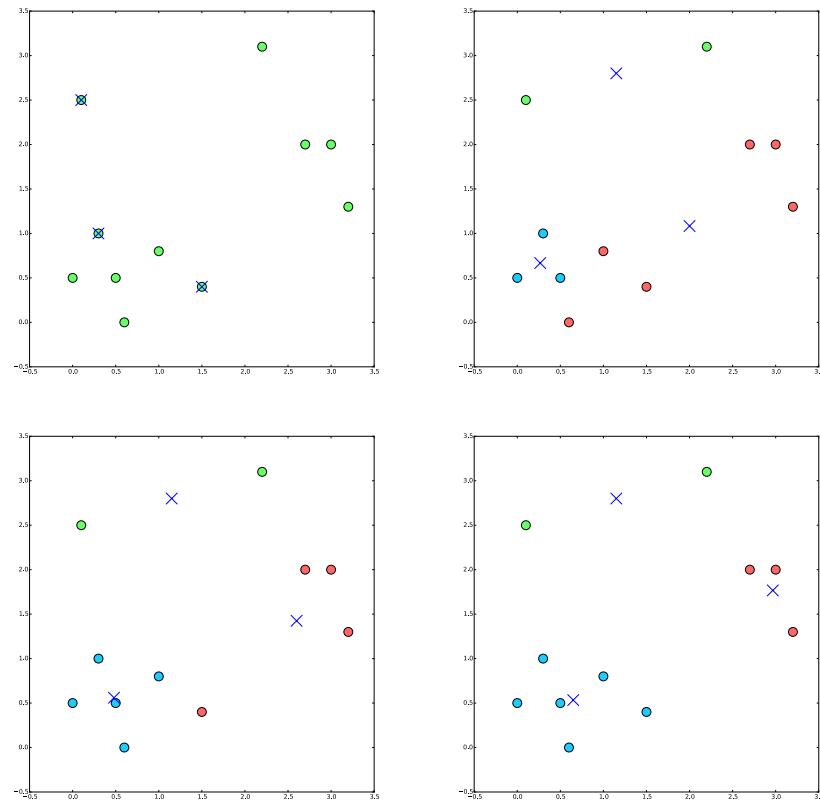


Abbildung 7: K-Means Clustering

Die Anzahl der gesuchten Cluster k muss zu Anfangs festgelegt werden. Im ersten Schritt des Algorithmus werden k Datenpunkte aus X als initiale Zentren festgelegt. Eine erste Partitionierung erfolgt, indem die übrigen Datenpunkte dem am nächsten liegenden Zentrum zugeordnet werden. Diese Zuordnung kann man als eine Matrix b_{ij} darstellen, deren Elemente wie folgt definiert sind:

$$b_{ij} = \begin{cases} 1 & x_i \in S_j \\ 0 & \text{sonst} \end{cases}$$

Im nächsten Schritt werden die Zentren neu berechnet:

$$c_j = \frac{\sum_{i=1}^n b_{ij} x_i}{\sum_{i=1}^n b_{ij}}$$

Dies wird so lange wiederholt, bis ein Abbruchkriterium erfüllt ist. Dies kann je nach Implementation nach einer gewissen Anzahl von Iterationen sein, oder wenn J unter einen gewissen Wert fällt.

Der Algorithmus reagiert sehr empfindlich auf die initialen Zentren, weshalb mehrere Erweiterungen, wie der k -Means++ Algorithmus, entwickelt wurden um ein besseres Verhalten zu erzielen. Da die Zuweisung der Datenpunkte zu den Clustern eindeutig ist, ist dies ein hartes Clusteringverfahren. Weiters muss die Anzahl der Cluster schon zu Beginn des Verfahrens bekannt sein, da die Anzahl der Cluster k ein wesentlicher Bestandteil des Algorithmus ist. Die Laufzeit des Algorithmus ist $\mathcal{O}(ikln)$, also nur linear von der Anzahl der Daten abhängig, sodass dieser Algorithmus auch für große Datensätze geeignet ist. i bezeichnet hier die Anzahl der Dimension, l die Iterationsschritte [18] [28].

FUZZY C-MEANS CLUSTERING

Der Fuzzy c-Means Algorithmus ist mit dem k-Means Algorithmus verwandt, auch hier wird mit Hilfe einer Zielfunktion J die Summe des quadrierten Fehlers minimiert:

$$J = \sum_{j=1}^k \sum_{x_i \in S_j} (u_{ij})^m d(x_i, c_j)^2 \rightarrow \min$$

u_{ij} ist hier eine $n \times k$ Matrix, die die Zugehörigkeit des Datenpunktes x_i zum j -ten Cluster beschreibt, $m \geq 1$ eine Konstante, die die Unschärfe der Cluster parametrisiert. d ist eine von einem Skalarprodukt induzierte Norm. Ein $m = 1$ würde hier ein hartes Clustering bedeuten. Sowohl k als auch m müssen im Vorhinein festgelegt werden.

Zur Initialisierung wird u_{ij} mit zufälligen Werten befüllt und aus dieser Zugehörigkeitsmatrix die Clusterzentren berechnet:

$$c_i = \frac{\sum_{j=1}^n (u_{ij})^m x_j}{\sum_{j=1}^n (u_{ij})^m}$$

Als nächstes wird die Zugehörigkeitsmatrix aktualisiert:

$$u_{ij} = \frac{1}{\sum_{n=1}^k \left(\frac{d(x_i, c_n)}{d(x_i, c_j)} \right)^{\frac{2}{m-1}}}$$

d bezeichnet hier eine beliebige Norm, welche von einem *Skalarprodukt* induziert wurde. Diese zwei Schritte werden wiederholt bis eine geeignete Abbruchbedingung erfüllt ist. Diese kann entweder die Anzahl der Iterationen sein, oder wenn J klein genug ist.

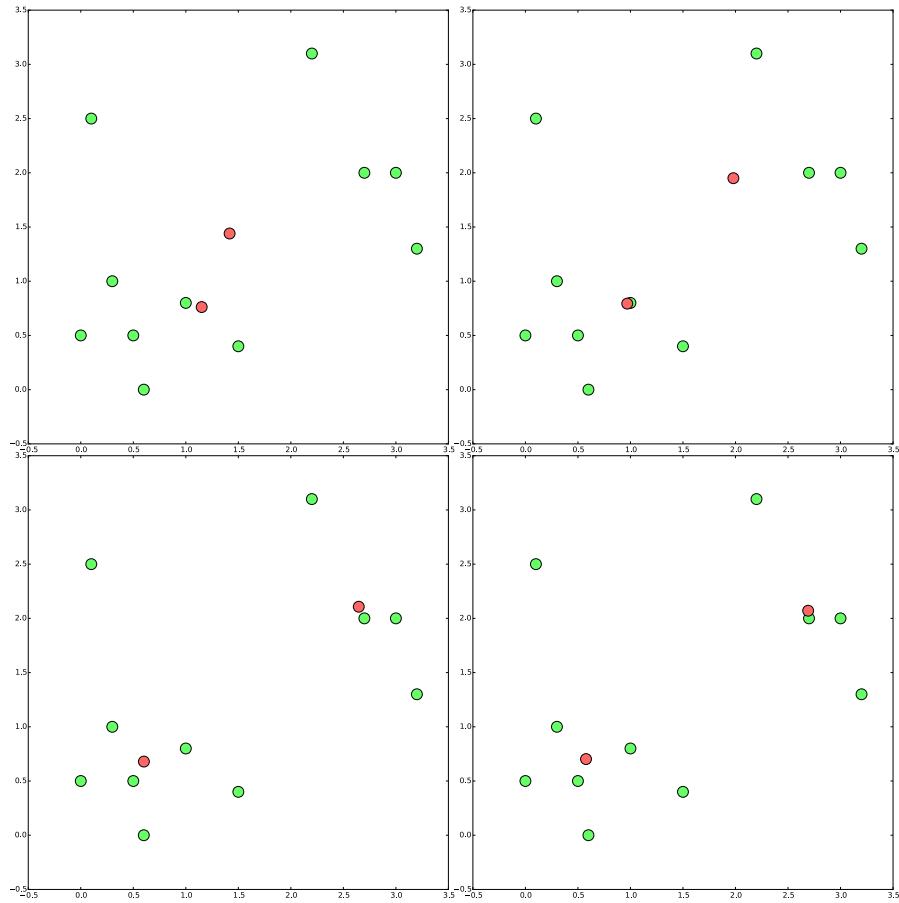


Abbildung 8: FCM Clustering

Wie auch der k-Means Algorithmus hängt dieser stark von den Initialwerten ab, ist aber weniger anfällig für etwaige Ausreißer. Die Laufzeit beträgt im Gegensatz zum k-Means $\mathcal{O}(ik^2ln)$. Dies ist den zusätzlichen Rechenschritten geschuldet. Auch hier ist i die Anzahl der Dimensionen, l die Anzahl der Iterationsschritte. Weiters ist es oftmals schwierig den optimalen Wert für den Parameter m zu finden, außerdem muss auch hier k schon vorher bekannt sein [5] [18].

3. FUZZY CONTROL

In der Regelungstechnik ist Fuzzy-Regelung mittlerweile ein etabliertes Verfahren um nichtlineare Probleme zu behandeln. Ihre Grundlage ist die unscharfe Logik (*fuzzy logic*), deren Theorie 1965 das erste Mal formuliert wurde [32]. Erst in den siebziger Jahren kamen erste Anwendungen im Bereich der Regelungstechnik auf, die sogenannte Fuzzy-Regelung [33]. Ziel hierbei war es oftmals menschliches Verhalten nachzuempfinden und in einen mathematischen Rahmen zu übersetzen [24].

3.1. UNSCHARFE MENGE

Ein scharfe Menge A kann über ihre Elemente beschrieben werden:

$$A = \{a_1, a_2, a_3, \dots\}$$

Wenn $A \subset X$, so kann man A auch über die *charakteristische Funktion* beschreiben.

Definition 7 (Charakteristische Funktion [30]). *Sei $X \subset \mathbb{R}^n$. Die Funktion $\mu : X \rightarrow \{0, 1\}$ heißt Charakteristische Funktion von A , wenn gilt:*

$$\mu_A(x) = \begin{cases} 1 & x \in A \\ 0 & \text{sonst} \end{cases}$$

Wenn wir zulassen, dass die Charakteristische Funktion nicht nur Werte aus $\{0, 1\}$ besitzt sondern beliebige Werte aus $[0, 1]$ annehmen kann, so sprechen wir von einer *Zugehörigkeitsfunktion* (*membership function*). Mit Hilfe dieser können wir nun unscharfe Mengen definieren:

Definition 8 (Unscharfe Menge [30]). *Sei $X \subset \mathbb{R}^n$. Eine unscharfe Menge $A \subset X$ ist definiert als eine Menge von geordneten Paaren*

$$A = \{(x, \mu_A) : x \in X\}$$

Wie wir sehen ist jede scharfe Menge ein Sonderfall einer unscharfen Menge. Analog zu den scharfen Mengen definieren wir auch hier die leere Menge und auch die Potenzmenge:

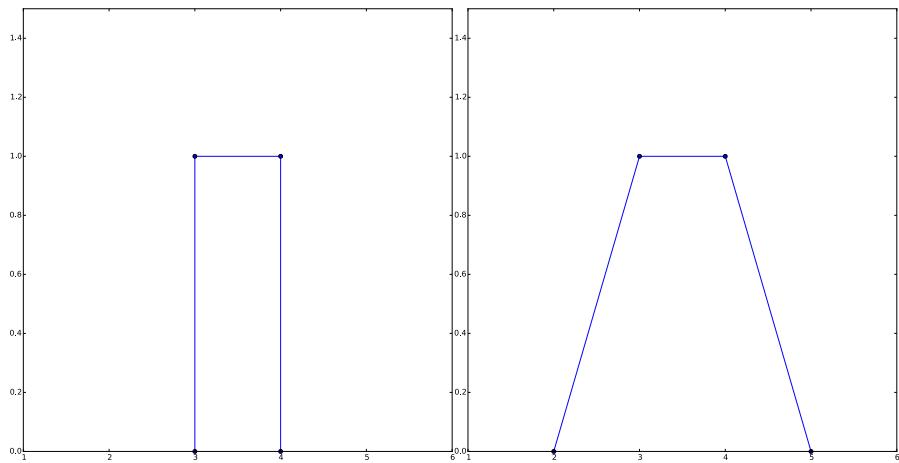


Abbildung 9: Scharfe und unscharfe Menge

Definition 9 (Leere Menge [24]). *Eine unscharfe Menge A heisst leere Menge, wenn gilt*

$$\mu_A(x) = 0 \quad \forall x \in X$$

Definition 10 (Potenzmenge [24]). *Die Menge $\mathcal{P}(A)$ aller unscharfen Mengen auf einer Grundmenge X heisst unscharfe Potenzmenge (fuzzy power set) der Menge X .*

ZUGEHÖRIGKEITSFUNKTIONEN

Unscharfe Mengen werden durch ihre Zugehörigkeitsfunktionen definiert. Die Zugehörigkeitsfunktionen werden meistens auf einem gewissen Bereich definiert, dem Träger:

Definition 11 (Träger [24]). *Sei $X \subset \mathbb{R}^n$. Der Träger einer unscharfen Menge $A \subset X$ ist definiert als:*

$$supp(A) = \{(x \in X) : \mu_A(x) > 0\}$$

Wenn der Träger einer unscharfen Menge nur aus einem einzigen Element besteht, also $supp(A) = x$, so nennen wir diese Menge einen *Singleton*. In der Regel ist der Träger aber eine größere Teilmenge von X und die Zugehörigkeitsfunktion entweder eine stetige oder stückweise lineare Funktion. Beliebte Funktionen hierfür sind Dreiecksfunktionen, Trapezfunktionen oder Gauß-Funktionen [24] [30].

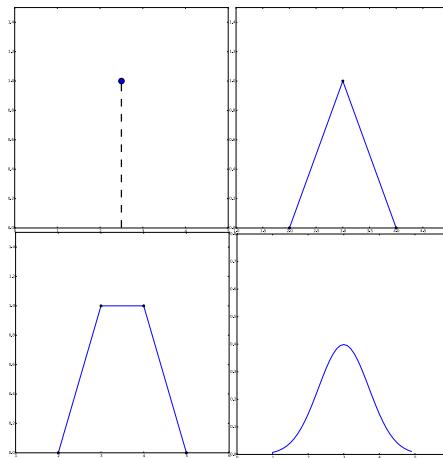


Abbildung 10: Häufige Zugehörigkeitsfunktionen: Singleton, Dreieck, Trapez, Gaußfunktion

MENGENOPERATIONEN

Wie auch für scharfe Mengen können darauf Operationen definieren.

Definition 12 (Durchschnittsmenge [24]). *Sei $X \subset \mathbb{R}^n$. Die Schnittmenge zweier unscharfen Mengen $A \subset \mathcal{P}(X)$ und $B \subset \mathcal{P}(X)$ ist definiert als:*

$$A \cap B = \{(x, \mu_{A \cap B}(x)) : x \in X\}$$

wobei $\mu_{A \cap B}$ definiert ist als

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(x))$$

Definition 13 (Vereinigungsmenge [24]). *Sei $X \subset \mathbb{R}^n$. Die Vereinigungsmenge zweier unscharfen Mengen $A \subset \mathcal{P}(X)$ und $B \subset \mathcal{P}(X)$ ist definiert als:*

$$A \cup B = \{(x, \mu_{A \cup B}(x)) : x \in X\}$$

wobei $\mu_{A \cup B}$ definiert ist als

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(x))$$

Definition 14 (Komplement [24]). *Sei $X \subset \mathbb{R}^n$. Das Komplement einer unscharfen Menge $A \subset \mathcal{P}(X)$ und $B \subset \mathcal{P}(X)$ ist definiert als:*

$$A^c = \{(x, \mu_{A^c}(x)) : x \in X\}$$

wobei μ_{A^c} definiert ist als

$$\mu_{A^c} = 1 - \mu_a(x)$$

Man kann aber diese Mengenoperationen auch als logische Operatoren ODER und UND betrachten.

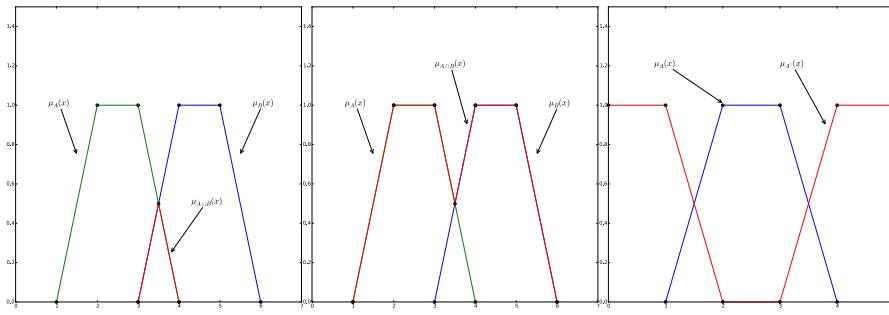


Abbildung 11: Durchschnittsmenge, Vereinigungsmenge und Komplement

3.2. UNSCHARFE LOGIK

Klassische Logik beschäftigt sich mit Aussagen und ihrem Wahrheitswert, der entweder 1 (wahr) oder 0 (falsch) ist. Wir sprechen von unscharfer Logik (*fuzzy logic*), wenn Aussagen einen Wahrheitswert einen beliebigen Wert aus $[0, 1]$ annehmen können. Sei nun δ die Funktion, die jeder Aussage a ihren Wahrheitswert aus $[0, 1]$ zuordnet, so können wir analog zur klassischen Logik Junktoren definieren [17]:

Definition 15 (Fuzzy-logische Junktoren).

$$\text{Fuzzy-logische Negation:} \quad \neg \quad \delta(\neg a) = 1 - \delta(a)$$

$$\text{Fuzzy-logische Disjunktion:} \quad \vee \quad \delta(a \vee b) = \max(\delta(a), \delta(b))$$

$$\text{Fuzzy-logische Konunktion:} \quad \wedge \quad \delta(a \wedge b) = \min(\delta(a), \delta(b))$$

$$\text{Fuzzy-logische Sujunktion:} \quad \rightarrow \quad \delta(a \rightarrow b) = \min(1, 1 + \delta(b) - \delta(a))$$

$$\text{Fuzzy-logische Bijunktion:} \quad \leftrightarrow \quad \delta(a \leftrightarrow b) = 1 - |\delta(a) - \delta(b)|$$

Mit Hilfe dieser Junktoren können auch zusammengesetzte Aussagen berechnet werden. Um nun eine Aussage der Art "X ist A" treffen zu können, führen wir nun den Begriff der Possibilitätsverteilung.

Definition 16 (Possibilitätsverteilung [17]). *p ist eine Possibilitätsverteilung für*

die Aussage “ X ist A ”, wenn gilt:

$$\text{poss}(X = u : X \text{ ist } A) = \mu_A(u)$$

IF-THEN-REGEL

Um ein gewissen kausalen Zusammenhang zwischen zwei Possibilitätsverteilungen A und B herzustellen, führen wir *if-then-Regeln* (*if-then-rules*) ein. Diese haben im einfachsten Fall folgende Form [17]:

IF X ist A THEN Y ist B

X und Y werden hierbei als linguistische Variable und A und B als linguistische Werte bezeichnet. Die Relation, die diesen Zusammenhang beschreibt wird wie folgt beschrieben:

$$\mu_R(x, y) = \max(\min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x))$$

3.3. FUZZY-REGELUNG

Aus den vorangegangenen Definitionen können wir nun Fuzzy-Systeme erstellen, die mit Hilfe von einem Regelsystem versuchen aus (un)scharfen Eingangsgrößen x_i , die (un)scharfe Ausgangsgröße y zu erhalten. Die Basis für das Verfahren stellen hier die *if-then-Regeln* dar. Ein solcher so genannter *Mamdani-Controller* besteht aus folgenden Komponenten [19]:

- Fuzzifier
- Regelbasis

- Fuzzy-Inferenzmaschine

- Defuzzifier

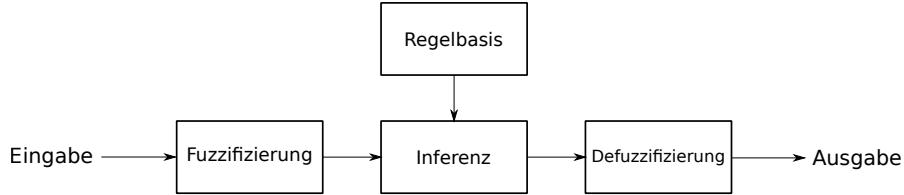


Abbildung 12: Schema einer Fuzzy-Regelung nach Mamdani

FUZZIFIER

Da oftmals scharfe Daten vorliegen, müssen diese zuerst in unscharfe Werte umgewandelt werden, um in der Folge von der Fuzzy-Inferenzmaschine verarbeitet werden zu können. Diesen Vorgang nennt man *Fuzzifizierung*. Hierbei werden dem Eingangswert x die Zugehörigkeitswerte $\mu_{A_i}(x)$ zugeordnet. Der Wahrheitswert der Aussagen X ist A_i ist dann gleich dem Zugehörigkeitsgrad von x zur unscharfen Menge, die dem Term A_i zugeordnet ist. Die Festlegung dieser unscharfen Mengen ist deswegen wichtig, da so die Einteilung des Wertebereichs erstellt wird. Außerdem muss noch die Form der Zugehörigkeitsfunktion gewählt werden. Wie schon erwähnt werden hierzu meist Singletons, aber auch stückweise lineare Funktionen, wie etwa Dreiecksfunktionen oder Trapezfunktionen sowie Glockenkurven gewählt [16] [17].

REGELBASIS

Dies sind die linguistischen Regeln, die später in der Fuzzy-Inferenzmaschine zur Anwendung kommen. Sie bestehen aus if-then-Regeln, wobei im Normalfall in der

Prämissen ausschließlich die Konjunktion als Verknüpfung verwendet wird und meist genau eine Fuzzy-Menge als Konklusion enthalten ist [17] [16].

FUZZY-INFERENZMASCHINE

Die Fuzzy-Inferenzmaschine ist nun die Anwendung der Regelbasis auf die fuzzifizierten Eingangswerte. Man berechnet zuerst den *Erfülltheitsgrad* der Prämissen. Dies geschieht über die Berechnung des Zugehörigkeitswertes der Einzelkomponenten, danach werden diese, abhängig von der Art der Verknüpfung, miteinander verknüpft. Wenn also die Regel die Form

IF x^1 is X^1 **AND** ... **AND** x^n is X^n

THEN y^1 is Y^1 **AND** ... **AND** y^n is Y^n

besitzt, wird erst die Zugehörigkeit der einzelnen x^n berechnet, danach der “ \wedge ” oder der “ \vee ” Operator angewandt.

Um danach eine Konklusion zu ziehen, wird zuerst der Erfülltheitsgrad der Prämissen mit den Fuzzy-Mengen Y^n verknüpft. Hier sind zwei Methoden üblich:

- *Minimumverfahren*: $\mu_{Y^n}^* = \min a_n, \mu_{Y^n}(y^n)$, wobei a_n der Erfülltheitsgrad der Prämissen ist.
- *Produktverfahren*: $\mu_{Y^n}^* = a^n \mu_{Y^n}(y^n)$. Auch hier ist a_n der Erfülltheitsgrad der Prämissen.

Der letzte Schritt hier ist die Berechnung der Ausgabe-Fuzzy-Mengen. Dies geschieht durch die Zusammenfügung der modifizierten $\mu_{Y^n}^*$ Zugehörigkeitsfunktionen. Dies geschieht im Normalfall mit dem Maximum [16] [17] [30].

$$\mu_{Y^n} = \max(\mu_{Y^n}^*)$$

DEFUZZIFIZIERUNG

Aus der Zugehörigkeitsfunktion der Ausgabe muss nun noch ein scharfer Wert erstellt werden. Dies erfolgt üblicherweise entweder mit Hilfe der Schwerpunktmethode oder der Singleton-Schwerpunktmethode.

SCHWERPUNKTMETHODE

Diese ist eine der numerisch aufwändigeren Methoden, erzielt aber ein relativ glattes Lösungsverhalten.

$$y_n = \frac{\int_Y y \mu_{Y^n}^*(y) dy}{\int_Y \mu_{Y^n}^*(y) dy}$$

SINGLETON-SCHWERPUNKTMETHODE

Hier wird die Berechnung deutlich reduziert indem man $\mu_{Y^n}^*$ zu einem Singleton vereinfacht [16] [17] [30].

$$y_n = \frac{\sum_{y_i \in Y} y_i \mu_{Y^n}^*(y_i)}{\sum_{y_i \in Y} \mu_{Y^n}^*(y_i)}$$

4. ADAPTIVE ONLINE FUZZY INFERENCE SYSTEM

Dieses Kapitel stellt eine Methode für eingebettete Intelligenz in einem AAL Kontext vor. Es beruht auf einer Technik zur Generation von interpretierbaren Fuzzy Zugehörigkeitsfunktionen, die mit Hilfe von zweifachem Clustering extrahiert werden [34]. Ziel ist es das Verhalten des Benutzers zu lernen und danach autonom und unaufdringlich Entscheidungen für diesen Benutzer zu treffen, wobei auf Änderungen reagiert werden kann und die Entscheidungen dementsprechend dynamisch angepasst werden [9].

Die Miniaturisierung von Computern schreitet immer weiter voran und ermöglicht es leistungsstarke autonome Geräte in intelligenten Umgebungen einzusetzen und somit Menschen unaufdringlich bei Entscheidungsfindungen zu unterstützen. Unaufdringlich bezieht sich hierbei nicht nur auf die physische Präsenz solcher Geräte sondern auch auf die Notwendigkeit zur Interaktion, welche wenn möglich weitgehend minimiert werden sollte. Es wird durch die Zunahme solcher intelligenten Geräte immer schwieriger jedes einzelne zu bedienen und dieses stellt ein Hindernis zur Allgegenwärtigkeit solcher Systeme dar [12]. Insofern scheint es sinnvoll Systeme zu entwickeln die nicht nur Daten austauschen können, sondern auch keinerlei Konfiguration vom Benutzer verlangen. Um dies zu bewerkstelligen muss ein solches System in der Lage sein verschiedene Verhaltensmuster des Benutzers zu erkennen, aber auch auf etwaige Änderungen ebensolcher zu reagieren und dem Benutzer die Möglichkeit erlauben das System zu übersteuern [9].

Das hier vorgestellte System ist einem Kontrollsysteem sehr ähnlich [7], wobei

die vom System gesteuerte Umgebung unter Berücksichtigung des Faktors Mensch, hochgradig nichtlinear ist. Um die unpräzisen Daten modellieren zu können wird auf ein *Fuzzy-System* zurückgegriffen. Es werden hierbei linguistische Informationen in mathematische Begriffe umgewandelt. Das *Adaptive Online Fuzzy Inference System* (AOFIS) besteht aus fünf Schritten, welche anschließend näher erläutert werden [8]:

- Sammeln von Daten
- Extraktion von Zugehörigkeitsfunktionen
- Extraktion von Fuzzy Regeln
- Anwenden der Fuzzy Regeln
- Adaptieren der Ausgabe

4.1. SAMMELN VON DATEN

Bevor das System die Steuerung übernehmen kann, müssen eine ausreichende Anzahl von Eingaben und Ausgaben gesammelt werden. Eingaben sind hier Sensordaten, Ausgaben Aktuatordaten. Immer wenn ein Aktuator betätigt wurde, wird ein Schnappschuss des gesamten Systems aufgezeichnet. Dies beinhaltet den Momentanen den Status aller Aktuatoren und die Werte aller Sensoren. Diese werden in Vektoren gespeichert, wobei $X = (x_1, x_2, \dots) \in \mathbb{R}^n$ die Eingabedaten und $Y = (y_1, y_2, \dots) \in \mathbb{R}^n$ die Ausgabedaten bezeichnen. Wir betrachten im Folgenden aber den aus Eingabedaten und Ausgabedaten zusammengesetzten Vektor $R = (x_1, \dots, x_n, y_1, \dots, y_m) \in \mathbb{R}^{n+m}$.

4.2. EXTRAKTION VON ZUGEHÖRIGKEITSFUNKTIONEN

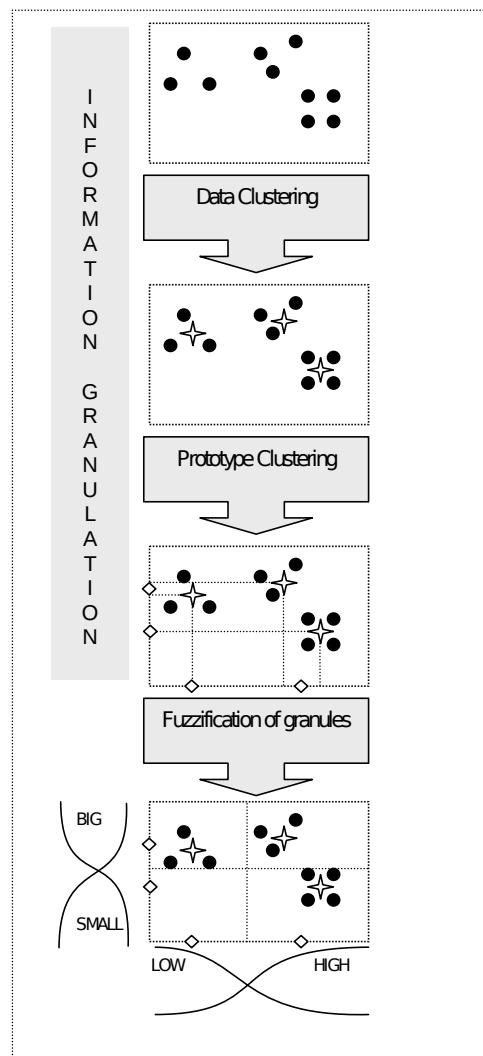


Abbildung 13: Extraktion von Zugehörigkeitsfunktionen

Um die scharfen Eingabe- und Ausgabedaten in linguistische Werte, wie kalt, warm umzuwandeln, wird eine zweifache Clusteringmethode angewandt. In der ersten Phase werden mit Hilfe der Fuzzy c-Means Clustering Methode eine vorgegebene-

ne Anzahl von k Clustern aus R mit den Zentren c_i extrahiert. Diese Zentren sind $n + m$ -Dimensionale Vektoren. Wenn wir von diesen Zentren die einzelnen Koordinaten betrachten, erhalten wir Skalare, die wir nochmalig Clustern können. Hier verwenden wir ein hierarchisches und agglomeratives Clusteringverfahren, mit folgendem Ablauf: Wenn c_i^j die j -te Koordinate des i -ten Vektors c_i bezeichne, betrachten wir die Menge

$$C_j = \{c_1^j, c_2^j, \dots\}$$

und sortieren diese so, dass $c_i^j \leq c_{i+1}^j$. Wir suchen nun das Element mit

$$d(c_i^j, c_{i+1}^j) \rightarrow \min$$

Also das Element mit kleinstem Abstand zu seinem Nachbarn. Wir bezeichnen es mit \tilde{c}_i^j und den Nachbarn mit \tilde{c}_{i+1}^j . Wir bilden nun eine neue Menge

$$C_j^{k-1} = \{c_1^j, \dots, c_{i-1}^j, \frac{(\tilde{c}_i^j + \tilde{c}_{i+1}^j)}{2}, c_{i+2}^j, \dots\}$$

Dies wird so oft wiederholt bis nur mehr die gewünschte Anzahl l_j an Cluster in der Menge C_j^l vorhanden sind.

Als nächsten Schritt müssen aus diesen Zentren Zugehörigkeitsfunktionen gebildet werden. In diesem Verfahren werden Gauß'sche Zugehörigkeitsfunktionen verwendet, es müssen also ein passendes μ und σ gefunden werden. Wir definieren hierzu eine Menge an Schnitten $T_j = \{t_1^j, t_2^j, \dots\}$:

$$t_i^j = \begin{cases} 2 \cdot \min(c_i^j \mid c_i^j \in C_j^l) - t_1^j & i = 0 \\ \frac{c_i^j + c_{i-1}^j}{2} & 0 < i < l_j \\ 2 \cdot \max(c_i^j \mid c_i^j \in C_j^l) - t_{k-1}^j & i = l_j \end{cases}$$

Der Mittelwert und die Standardabweichung der Zugehörigkeitsfunktion $N(\mu, \sigma)_i^j$ werden nun wie folgt definiert:

$$\mu_i^j = \frac{(t_{i-1}^j + t_i^j)}{2}$$

$$\sigma_i^j = \frac{t_i^j - t_{i-1}^j}{2\sqrt{-2 \ln \epsilon}}$$

Wobei ϵ den Überlappungsfaktor der Zugehörigkeitsfunktionen darstellt. Man kann nun den zu einer Koordinate gehörigen Zugehörigkeitsfunktionen auch linguistische Werte wie *hoch*, *mittel* oder *niedrig* zuweisen.

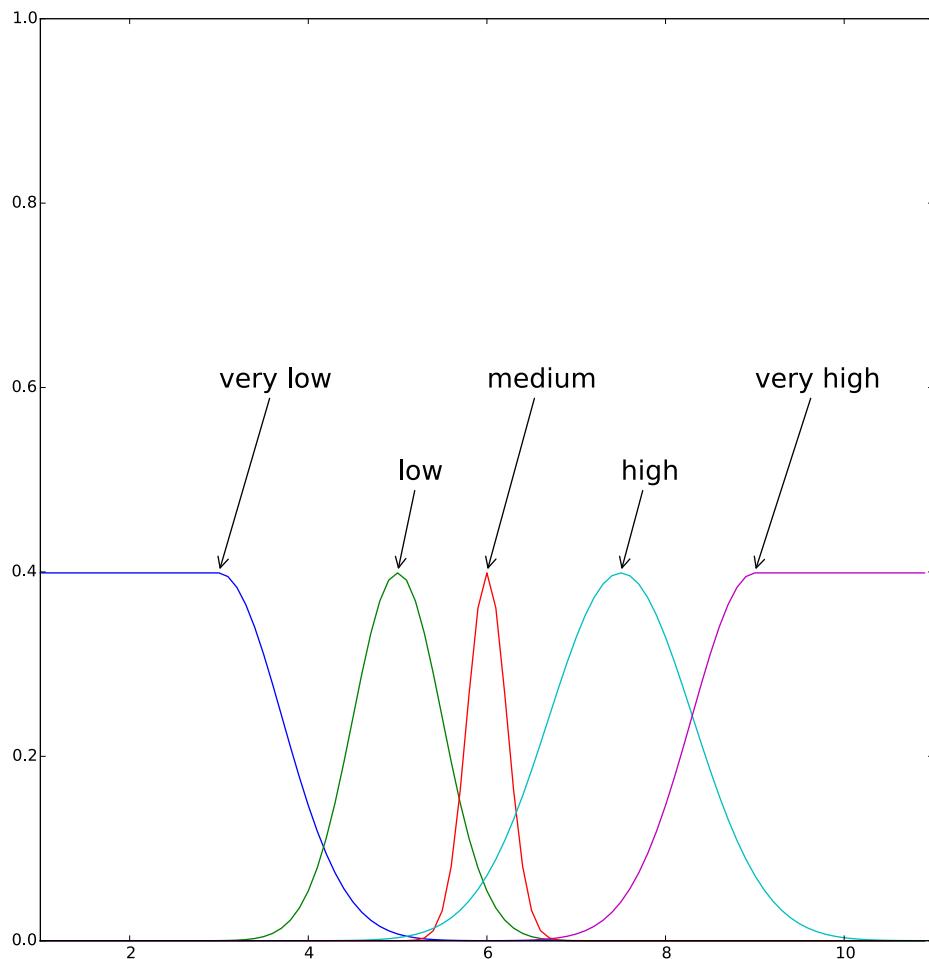


Abbildung 14: Zugehörigkeitsfunktionen mit linguistischer Zuordnung

4.3. EXTRAKTION VON FUZZY REGELN

Um nun das Verhalten des Benutzers abzubilden, werden die im vorigen Schritt extrahierten Zugehörigkeitsfunktionen verwendet um aus bestehenden Daten Fuzzy Regeln zu erstellen. Wenn x und y wieder wie im vorigen Abschnitt definiert sind, haben die Regeln folgende Form:

WENN x^1 ist A_i^1 **UND** ... **UND** x^n ist A_i^n

DANN y^1 ist B_i^1 **UND** ... **UND** y^m ist B_i^m

A_i^n und B_i^m sind hierbei die zu den Eingabedaten beziehungsweise zu den Ausgabedaten gehörigen Zugehörigkeitsfunktionen $N(\mu, \sigma)_i^j$.

Um die Vorgehensweise zu erläutern wird im Folgenden y als Eindimensionale Größe angenommen. Für jeden Vektor $R = \{x_1, \dots, x_n, y\}$ suchen wir die Zugehörigkeitsfunktionen A_i^j , die für jedes x_j den höchsten Wahrheitswert besitzt:

$$\mu_{A_i^j}(x_j) \geq \mu_{A_k^j}(x_j) \quad \forall k \leq l_j$$

Für jede EingabevARIABLE x_t können wir also eine erste Regel formulieren, die folgende Form besitzt:

WENN x_t^1 ist A_i^1 **UND** ... **UND** x_t^n ist A_i^n

DANN y ist zentriert bei y_t

Weiters definieren wir noch das Gewicht dieser Regel:

$$w_t = \prod_{j=1}^n \mu_{A_i^j}(x_t^j)$$

Diese Regeln haben oft widersprüchliche Antezedenzen, weswegen diese kombiniert werden müssen um eine konsistentes Regelsystem zu erhalten. Es werden

also alle k inkonsistenten Regeln mit der gleichen Konsequenz zusammengefasst und ein gewichtetes Mittel dieser Regeln als endgültige Regel formuliert.

$$\tilde{y} = \frac{\sum_{t_k=1}^k y_{t_k} w_{t_k}}{\sum_{t_k=1}^k y_{t_k}}$$

Die endgültige Regel lautet nun:

WENN x_t^1 ist A_i^1 **UND** ... **UND** x_t^n ist A_i^n

DANN y ist B_t

Wobei B_t die Zugehörigkeitsfunktion der Ausgabedaten ist, die größten Wahrheitswert zu \tilde{y} besitzt.

$$\mu_{B_t}(\tilde{y}) \geq \mu_{A_k}(\tilde{y}) \quad \forall k \leq l_j$$

Für mehrdimensionale Ausgabewerte, müssen nun für jede Dimension analog eine solche Regeln erstellt werden.

4.4. ANWENDEN DER FUZZY REGELN

Mit Hilfe der nun extrahierten Regeln kann das System dazu übergehen, die Ausgabe zu steuern und die menschlichen Verhalten zu imitieren. Als Fuzzy-System (*FLC*) betrachtet, besteht das System aus der Fuzzifizierung, einer Inferenzschemata, dem Regelsystem und der Defuzzifizierung. Wenn nun scharfe Eingabedaten x in das System geschickt werden, so werden diese in Fuzzy Werte umgewandelt und mit Hilfe des Regelsystems in Fuzzy Ausgabewerte umgewandelt. Schlussendlich werden diese wieder in scharfe Ausgabewerte y umgewandelt. In diesem System

werden Singletonfuzzifizierung, das Produktverfahren zur Komposition, die Max-Prod-Inferenz, sowie die Singleton-Schwerpunktmethode zur Defuzzifizierung angewandt. Als Formel zusammengefasst ergibt sich folgende Vorschrift:

$$y = \frac{\sum_{l=1}^M \bar{y}^l \prod_{j=1}^n \mu(x_j)}{\sum_{l=1}^M \prod_{j=1}^n \mu(x_j)}$$

Hier ist \bar{y}^l der Punkt mit dem höchsten Wahrheitswert in der Zugehörigkeitsfunktion B_t ist. Für mehrdimensionale y muss dies für alle Dimensionen wiederholt werden.

4.5. ADAPTIEREN DER AUSGABE

Oftmals ist es notwendig das Verhalten des Systems während der Laufzeit einem neuen Verhaltensmuster des Benutzers anzupassen. Im Kontext von *AAL* sollte dies auf nicht aufdringliche Weise passieren und außerdem dem Benutzer jederzeit die Möglichkeit geben das System zu korrigieren, wenn nicht gewünschtes Verhalten auftritt. Wenn nun der Benutzer eine Regel übersteuert, wird ein Schnappschuss der Eingaben und Ausgaben zu diesem Zeitpunkt erstellt und für alle x_j Eingabewerte das Gewicht der Antezedenzen aus den Regeln aus unserem Regel- system betrachtet. Wenn ein Gewicht, also das Produkt des Wahrheitswertes der Zugehörigkeitsfunktionen, mehr als ein vorgegebenes Signifikanzniveau von 0.05 erreicht wird, werden alle Regeln mit den neuen Ausgabewert angepasst, da sie ja einen Einfluss auf die Defuzzifizierung gehabt hätten. Die Konsequenz wird ersetzt durch die maximale Zugehörigkeitsfunktion des neuen Ausgabewertes y^* . Hierbei muss man allerdings Bedenken, dass ein solches Übersteuern der Regelung manchmal nur ein einmaliges Ereignis sein kann. Es muss also eine gewisse Schranke für

das Ansprechen der Adoptionsroutine geben. Man kann hierzu eine gewisse zeitliche Spanne definieren, die bei mehrmaligem Übersteuern innerhalb dieser, oder einfach bei wiederholtem übersteuern die Routine aufruft. Wenn nun eine Kombination von Regeln in der Prämisse noch nicht aufgetreten ist, wird dynamisch eine neue Regel nach obigem Verfahren erstellt. Diese neue Regel bildet nun ein neues Verhaltensmuster des Benutzers ab, das System lernt also stetig vom Benutzer und passt sein Verhalten immer besser dem des Benutzers an.

4.6. IMPLEMENTIERUNG

Die Implementierung des Programmes erfolgte in Java (*OpenJDK 8*). Es wurde versucht viele Methoden so allgemein wie möglich zu halten um ihre Wiederverwendbarkeit zu gewährleisten. Die *apache-commons-math* [2] Bibliothek stellt viele Grundlegende Klassen und Methoden bereit und wurde im gesamten Projekt verwendet. Die Clustering Algorithmen wurden mit Datensätzen der University of Eastern Finland getestet [22].

CLUSTERING

Die zugrundeliegende abstrakte Klasse der Clustering Algorithmen wurde aus dem *apache-commons-math* [2] Projekt übernommen. Es wurde die Verfahren aus 2.3 sowie aus 4.2 als `FuzzyCMeansClusterer` bzw. `AgglomerativeHierarchicalClusterer` implementiert und getestet.

Die Parameter m , k sind bei der Implementierung des Fuzzy c-Means Algorithmus frei wählbar, außerdem kann die Abbruchbedingung als maximale Anzahl der Iterationen oder durch ein ϵ , welches die Änderung zwischen zwei Iteratio-

nen beschreibt festgelegt werden. Auch die Distanzfunktion ist nicht im Vorhinein festgelegt.

Beim Agglomerativen Hierarchischen Clustering Algorithmus ist k sowie die Anzahl der Iterationen frei wählbar, außerdem ist die Distanzfunktion wieder frei wählbar.

Der Algorithmus selbst ist jeweils in der `cluster` Methode implementiert, die eine Liste von `CentroidCluster` (siehe [2]) zurückgibt.

GENERIERUNG DER ZUGEHÖRIGKEITSFUNKTIONEN

Um von Clusterzentren zu Zugehörigkeitsfunktionen zu gelangen wurde die `DoubleClusterToMSFGaussian` erstellt. Diese berechnet in der `generateMSF` Methode die Schnitte, welche in 4.2 definiert wurden und aus diesen den Mittelwert μ sowie die Standardabweichung σ . Hier wird eine List von `MembershipFunctionGaussian` Objekten zurückgegeben.

REGELBASIS

Nachdem die Daten doppelt geclustert und die Zugehörigkeitsfunktionen erstellt wurden, wird das Regelsystem mit Hilfe der `RuleGenerator` Klasse eine Regelbasis erstellt. Das Verfahren ist in 4.3 beschrieben und nutzt die `extractFinalRules` Methode. Diese ist etwas umfangreicher, da hier aus allen Daten zuerst die jeweiligen Regeln extrahiert, sowie zu einer finalen Regel kombiniert werden.

Die `adaptRules` ist diejenige Methode, die bei einem überschreiben der Regel durch den Benutzer angewandt wird (siehe 4.5). Hier wurde eine änderung bezüglich der Veröffentlichung von Faiyaz Doctor, Hani Hagras und Victor Cal-

laghan vorgenommen. Diese sind von kompakten Träger der Zugehörigkeitsfunktionen ausgegangen, in dieser Implementation wurde aber eine allgemeinere Zugehörigkeitsfunktion implementiert. Deswegen wurde ein Schwellenwert von 0.05 eingeführt, unter dem eine Regel als nicht signifikant eingestuft und deswegen nicht adaptiert wird.

Als letzte Methode existiert noch `getCrispOutputFromInput`, welche die Vorschrift aus 4.4 implementiert.

MAIN

DoubleClustering ist die Hauptklasse, in der die Daten eingelesen, verarbeitet sowie die Ergebnisse getestet werden.

Algorithm 4.1: AOFIS(*inputFile, outputFile*)

procedure GETIO(*inputFile, outputFile*)

input \leftarrow generateInputData(*inputFile*)

output \leftarrow generateOutputData(*outputFile*)

ioList \leftarrow combineIO(*input, output*)

return (*ioList*)

procedure GETRULES(*ioList*)

clusterList \leftarrow DoubleClusterer.cluster(*ioList*)

msfList \leftarrow ClusterToMSFGaussian.generateMSF(*clusterList*)

ruleList \leftarrow RuleGenerator.generateFinalRules(*msfList, ioList*)

return (*ruleList*)

procedure CHECKRULES(*IO, ruleList*)

RightRule \leftarrow RuleGenerator.getCrISPOutputFromInput(*IO*)

if *RightRule* == *IO.getOutput()*

then return (*RightRule*)

else RuleGenerator.adaptRules(*IO*)**return** (*IO.getOutput()*)

main

ioList \leftarrow GetIO(*inputFile, outputFile*)

ruleList \leftarrow GetRules(*ioList*)

if *newIO*

then CheckRules(*newIO, ruleList*)

4.7. ERGEBNISSE UND AUSBLICK

Um das System zu testen, wurden über mehrere Wochen 18 Sensoren aufgezeichnet und kategorisiert. Jede Belegung zu einem gewissen Zeitpunkt wurde einer von 11 Aktivitäten des täglichen Lebens zugeordnet.

1	NULL
2	Comb Hair
3	Wash Hands
4	Wash Face
5	Brush Teeth
6	Make Bed
7	Change Clothes
8	Put Blinds up/down
9	Prepare Food
10	Eat
11	Open/Close Window

Tabelle 1: Aktivitäten des täglichen Lebens

Aus diesen Datensatz von 9896 Zeitpunkten wurden 3000 zufällige Stichproben ohne Zurücklegen gezogen benutzt um das System zu trainieren und eine erste Regelbasis zu erstellen. Die übrigen Datensätze wurden dem System als Testdaten übergeben. Dies sind natürlich nicht optimale Testdaten, da nur eindimensionale Ausgangsdaten verwendet wurden, aber zum Testen des Systems verwendbar, da es sich um reale Sensordaten handelt.

Die Parameter für die Anzahl der Cluster wurden auf 55 erste “Cluster” und 9 Doppelclustern festgesetzt. Erste Ergebnisse zeigte eine Trefferquote von knapp 64%. Eine eingehende Analyse zeigte, dass der Großteil der Fehler durch Fälle eintraten, die durch die initiale Regelbasis nicht abgedeckt wurden. Von einer ersten Regelbasis von 2936 Regeln wurden 184 überschrieben, allerdings 2284 neue Regeln erkannt. Auch der Verlauf der Fehlerquote ließ erkennen, dass das System noch im Lernen war und die potentielle Regelbasis mit 18^9 verschiedenen Potentiellen Prämissen als zu fein festgesetzt wurde

	1	2	3	4	5	6	7	8	9	10	11
geordnet	638	87	74	73	233	297	90	233	589	139	38
ungeordnet	676	90	79	77	245	294	92	229	595	139	38

Tabelle 2: Anzahl der falsch zugeordneten Labels für 9 Doppelcluster

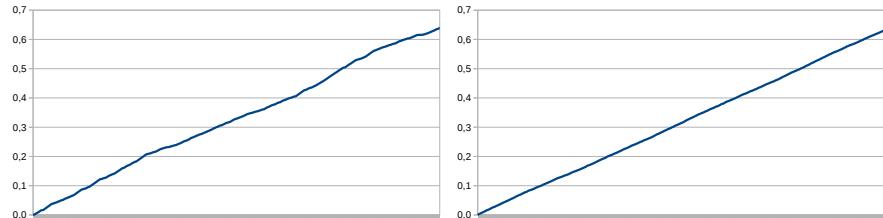


Abbildung 15: Entwicklung der Trefferrate für 9 Doppelcluster (geordnete Daten vs. ungeordnete Daten)

Ein weiterer Testlauf wurde wieder mit 55 erste Cluster und diesesmal nur 2 Doppelcluster gestartet. Hier zeigte sich dass zwei mögliche Cluster pro Dimension zu grob gewählt ist, da keine neuen Regeln erstellt wurden, sondern jede bestehende Regel überschrieben wurde. Die Trefferquote ist mit 86% höher als mit den vorigen

Paramenter, dies ist aber der Struktur der Daten geschuldet, da während dem Ablauf einer Aktivität immer mehrere Datensätze aufgezeichnet und nacheinander gespeichert wurden.

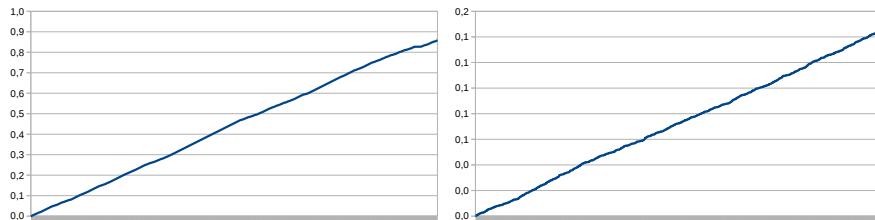


Abbildung 16: Entwicklung der Trefferrate für 2 Doppelcluster (geordnete Daten vs. ungeordnete Daten)

Nochmalige Testläufe mit in zufälliger Reihenfolge gezogenen Testdaten zeigte ein gänzlich anderes Bild mit einer Trefferquote von 15%. Zur Kontrolle wurde der ursprüngliche Ansatz mit 9 Doppelcluster mit zufällig gezogenen Daten wiederholt und es zeigte sich wie erwartet nur eine geringfügige Änderung der Trefferquote (von 64% zu 63%). Dies zeigt wie wichtig die Wahl der Parameter des Doppel-Clusterings auf die Qualität und Stabilität der Ergebnisse ist. Weitere Experimen-

	1	2	3	4	5	6	7	8	9	10	11
geordnet	306	79	32	38	64	150	94	133	29	12	38
ungeordnet	1393	282	120	137	1060	599	79	382	1116	630	38

Tabelle 3: Anzahl der falsch zugeordneten Labels für 2 Doppelcluster

te mit besseren Daten wurden von F. Doctor, H. Hagras und V. Callaghan an der University of Essex durchgeführt [9]. Dort wurden in einem intelligenten Dormitorium (*iDorm*) über einen Zeitraum von fünf Tagen Daten aufgenommen. Dieser

Datensatz wurde mit anderen Entscheidungsfindungssystemen wie *ANFIS* verglichen. Hierbei wurde eine höhere Trefferquote durch *AOFIS* festgestellt, wobei auch die benötigte Rechenleistung niedriger als bei *ANFIS* war.

Für zukünftige Verbesserungen ist vorgesehen dieses Verfahren in das *Home Event recognition system (HOMER)* zu integrieren. HOMER ist ein Platform, die mit Hilfe einer *finite state machine* versucht Geräte oder Aktuatoren zu steuern. Diese Regeln müssen händisch erstellt werden, was mit Hilfe dieses Systems auf einfache Weise automatisiert werden könnte. Außerdem kann untersucht werden, ob eine Dimensionsreduktion der Eingabedaten Auswirkungen auf die Trefferquote hat. Hierzu könnten Verschieden Methoden des Datamining angewandt werden.

5. EMPFEHLUNGSSYSTEME

Empfehlungssysteme *recommender systems* sind Softwaresysteme, die aus einer bestimmten Menge von Objekten solche auswählen, die für den Benutzer am relevantesten sind. Sie versuchen Vorschläge zu machen, was ein Benutzer beispielsweise kaufen oder welche Musik er hören soll. Mit Hilfe dieser Techniken wird versucht, Benutzern Entscheidungen zu erleichtern, auch wenn er in einem Gebiet kein Experte ist. Weiters ist es ein Ansatz, um aus großen Datenmengen relevante Daten zu extrahieren. Die vielfältigen Einsatzzwecke von Empfehlungssystemen sind ein Grund für das starke Interesse an solchen Systemen, obwohl es einen relativ jungen Bereich der Forschung darstellt. Es gibt verschiedene Ansätze um zu Vorschlägen zu kommen [23]:

Jedes Empfehlungssystem beruht auf Techniken des Datamining (*data mining*), im Rahmen dessen große Datenmengen aufbereitet, analysiert und in der Folge interpretiert werden. Im Rahmen des Empfehlungssystems können große Datenmengen anfallen, was die Wahl der spezifischen Algorithmen einschränkt [23].

5.1. COLLABORATIVE FILTERING

Diese Verfahren basieren auf der Bewertung von Objekten durch die Benutzer selbst. Die Annahme ist, dass ein Benutzer u_1 , der ähnliche Bewertungen wie ein Benutzer u_2 abgegeben hat, ein neues Objekt o_1 , das von u_1 noch nicht bewertet wurde, aber von u_2 bereits bewertet wurde, eine ähnliche Bewertung wie u_2 gibt. Es wird also die Gesamtheit der Benutzer zur Findung von Präferenzen herangezogen. Hier gibt es eine Vielzahl von Verfahren, die man noch weiter Zusammenfassen

kann in *nachbarschafts-basierte Methoden* und *Modell-basierte Methoden*. Diese sind simpel und intuitiv zu implementieren, stabil und effizient, funktionieren aber nur nach einer gewissen Anzahl von Trainingsdaten. Diese Abhangigkeit von einer gewissen Lernphase nennt man Kaltstart-Problem (*cold-start problem*) [28] [23].

5.2. CONTENT-BASED

Das Content-based System versucht an Hand von Eigenschaften eines Objektes die preferierten Objekte fur den Benutzer herauszufinden. Wenn ein Benutzer u_1 also ein Objekt o_1 hoch bewertet, werden ihm weitere Objekte vorgeschlagen, die eine ahnliche Struktur wie o_1 besitzen. Diese Struktur konnen beispielsweise Genres bei Musik oder Filmen, aber auch Autoren fur Bucher sein. Diese Strukturen konnen automatisiert durch *Content analyzer* erhoben, oder auch manuell eingegeben werden.

Die Vorteile dieses Ansatzes sind, dass sie vom Benutzer selbst unabhangig sind, die Transparenz des Verfahrens, sowie der Umgehung des sogenannten Kaltstart-Problems. Das Empfehlungssystem kann sofort eingesetzt werden und muss nicht vorher trainiert werden.

Die Nachteile betreffen die Analyse des Inhaltes. Automatisierte Verfahren konnen gewisse Eigenschaften von Objekten verlieren und eine manuelle Eingabe ist oft aufwandig. Weiters kann es passieren, dass durch die Einschrankung auf ahnliche Objekte, dem Benutzer keine neuen Objekte vorgeschlagen werden, er gewissermaen in einem Genre gefangen bleiben kann [23].

5.3. HYBRIDE VERFAHREN

Diese Verfahren versuchen die Vorteile der beiden anderen Verfahren zu vereinen sowie die Nachteile jeweils mit den Vorteilen des anderen auszugleichen. So kann beispielsweise das Kaltstart-Problem umgehen, indem man zu Beginn des Verfahrens die Daten des dazugehörigen Content-based Systems übernimmt [23].

5.4. TECHNIKEN ZUM DATAMINING

Es ist oft Notwendig, echte Daten aufzubereiten um zufriedenstellende Ergebnisse zu erzielen. Um dies zu erreichen können verschiedene Techniken angewandt werden um die Datenmenge zu reduzieren, oder zumindest die Dimension der Daten zu verringern.

STICHPROBEN

Stichprobennahme (*sampling*) ist eine der am Häufigsten eingesetzte Techniken um die Anzahl der Daten zu reduzieren. Das Problem hierbei ist, eine Untermenge zu finden die Representativ für die Gesamtdaten ist. Üblich ist es Stichproben zu ohne Zurücklegen zu ziehen, aber auch mit Zurücklegen wird vereinzelt angewandt. Hierbei ist eine 80 – 20 Teilung zwischen 20% Testdaten und 80% Trainingsdaten eine verbreitete Methode. In fortgeschrittenen Techniken wird auch eine gewisse Tendenz (*bias*) im Ziehen der Stichproben inkorporiert um beispielsweise neuere Daten häufiger zu ziehen [23].

DIMENSIONSREDUKTION

Die Art der Daten, die von einem Empfehlungssystem verarbeitet werden sind oft von hoher Dimension, aber dünnbesetzt. Dies sind allerdings sehr negative Eigenschaften zur Berechnung von Dichten oder von Ähnlichkeit von Objekten und wird als Fluch der Dimensionalität (*Curse of dimensionality*) bezeichnet. Um nun die Dimensionen zu Verringern, aber dennoch die essentiellen Informationen zu erhalten werden verschiedene Techniken eingesetzt [23] [28].

HAUPTKOMPONENTENANALYSE

Ein etabliertes Standardverfahren zur Dimensionsreduktion ist die Hauptkomponentenanalyse. Ausgangspunkt ist hierbei die Berücksichtigung einer Gewichtung der Merkmale eines Objekts, im Gegensatz zum Mittelwert. Diese Gewichtung ist hier abhängig von der Varianz. Es werden dabei in der Folge alle Komponenten mit einer kleinen Varianz weggelassen um eine Dimensionsreduktion zu erzielen.

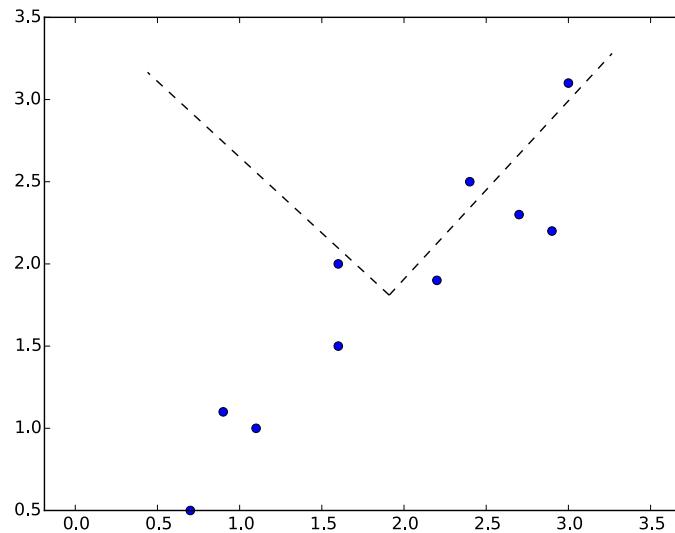


Abbildung 17: Hauptkomponentenanalyse

$X = (X_1, \dots, X_n)^T$ bezeichnet hier ein Objekt, während o_i dessen Merkmale sind.

Um nun die Hauptkomponentenanalyse durchzuführen, suchen wir $\delta = (\delta_1, \dots)$

$$\delta^T X = \sum_{i=1}^n \delta_i X_i \text{ sodass } \sum_{i=1}^n \delta_i^2 = 1$$

mit der Nebenbedingung

$$\max_{\delta: \|\delta\|=1} \text{Var}(\delta^T X) = \max_{\delta: \|\delta\|=1} \delta^T \text{Var}(X) \delta$$

Die Hauptkomponententransformation ist somit definiert als

$$Y = \Gamma^T (X - \mu)$$

wobei Γ definiert ist als $\text{Var}(X) = \Gamma \Lambda \Gamma^T$ und $\mu = \text{EX}$ [23] [13].

SINGULÄRWERTZERLEGUNG

Auch die Singulärwertzerlegung kann für Dimensionsreduktion eingesetzt werden.

Sie basiert auf der Zerlegung der Matrix M in die unitären Matrizen U und V sowie der reellen Matrix Σ , sodass M dargestellt werden kann als

$$M = U \Sigma V^*$$

V^* ist hier die zu V adjungierte Matrix. Σ hat hierbei folgende Form:

$$\Sigma = \left(\begin{array}{ccc|ccc} \sigma_1 & & & & \vdots & & \\ & \ddots & & & \dots & 0 & \dots \\ & & \sigma_r & & & \vdots & \\ \hline & \vdots & & & \vdots & & \\ \dots & 0 & \dots & \dots & 0 & \dots & \\ & \vdots & & & \vdots & & \end{array} \right)$$

Ausgehend von einer Objekt×Eigenschaften Matrix M , wird U bei dieser Methode als “Objekt-zu-Konzept” Ähnlichkeitsmatrix aufgefasst, während V als “Eigenschaft-zu-Konzept” Matrix interpretiert wird. Um nun eine Dimensionsreduktion durchzuführen werden diese Matrizen mit verkürzten Matrizen approximiert. Dies zeigt gute Resultate für die Genauigkeit der Approximation, ist aber aufwändig zu berechnen wenn sich die Matrix M ändert [23] [13] [28].

KLASSIFIKATION

Klassifikation bezeichnet Methoden, die Objekte in bestimmte Kategorien einteilen. Diese können im Vorhinein bekannt sein, oder aus den Objekten selbst generiert werden. Erstere nennt man *supervised*, letztere *unsupervised* Klassifikation.

NÄCHSTE-NACHBARN-KLASSIFIKATION

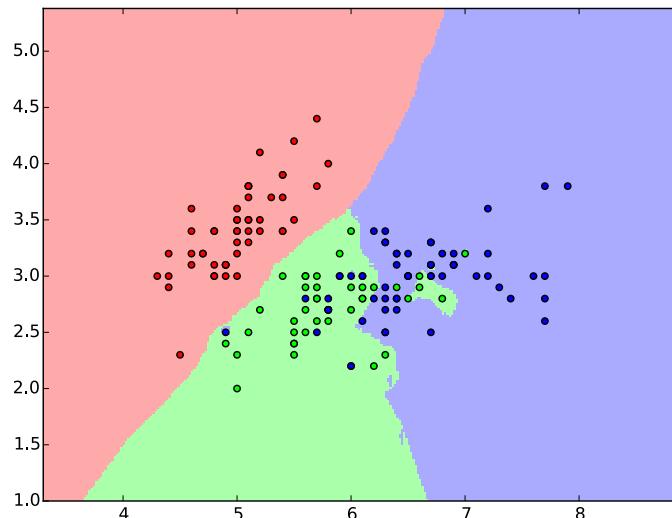


Abbildung 18: Nächste-Nachbarn-Klassifikation

Die Nächste-Nachbarn-Klassifikation (*nearest neighbors classification*) versucht einem Punkt eine gewisse Klasse zuzuweisen. Die Idee hierbei ist, einem neuen Ob-

jekt die Klassen zuzuweisen, die dominant in seiner Nähe sind. Ein bekannter Algorithmus ist hier bei der *k-Nearest-Neighbor (kNN)* Algorithmus. Dazu muss allerdings ein geeignetes Abstandsmaß vorher festgelegt werden. Meist wird dabei der euklidische Abstand oder die Manhattan-Metrik verwendet. Obwohl es eine sehr simple Herangehensweise ist, liefert sie gute Resultate [23] [18].

REGELBASIERTE KLASSIFIKATION

Regelbasierte Klassifikationsverfahren arbeiten mit if-then-Regeln. Dies ist allerdings kein sehr populäres Verfahren, da man die Regeln im Vorhinein aufstellen muss, was aber Expertenwissen impliziert [23].

NAIVE BAYES-KLASSIFIKATION

Bei einem Bayes-Klassifikator (*bayesian classifier*) wird jede Eigenschaft als (unabhängige) Zufallsvariable angesehen. Wenn man nun ein Objekt mit n Eigenschaften $A = \{A_1, \dots, A_n\}$ hat, ist das Ziel die Klasse C_k zu finden, sodass $\max(P(C_k|A))$. Wendet man das Bayessche Theorem an, so erhält man $P(C_k|A) \propto P(A|C_k)P(C_k)$. Da man von der Unabhängigkeit der A_i ausgegangen ist, kommt man schlussendlich auf $P(A|C_k) = P(A_1|C_k) \cdot P(A_2|C_k) \cdot \dots \cdot P(A_n|C_k)$ [23] [18].

ANDERE KLASSIFIKATIONSVERFAHREN

Diese Liste deckt bei weitem nicht alle wichtigen Verfahren ab. *Bayes'sche Klassifikationsverfahren*, *Support-Vector-Machines* oder *Künstliche Neuronale Netze* sind andere populäre Klassifikationsverfahren, die gute Ergebnisse im jeweiligen Anwendungsbereich liefern [23] [18].

CLUSTERING

Clustering wird einerseits eingesetzt um die Kandidaten für die Nächste-Nachbarn-Klassifikation zu reduzieren, aber auch in content-based Empfehlungssystemen um dort ähnliche Objekte aus der Grundmenge zu filtern. Eine umfangreichere Diskussion über Clustering Algorithmen und den dazu benötigten Distanzmaßen, ist in einem früheren Kapitel zu finden 2.

6. EMPFEHLUNGSSYSTEM FÜR *YouDo it - we help!*

6.1. *YouDo it - we help!*

Das *YouDo* Projekt ist eine Informations- und Trainingsplattform, deren Hauptaugenmerk auf einfacher Bedienung liegt. Es ist nicht nur für ältere Personen, oder Personen mit besonderen Bedürfnissen ausgelegt, sondern soll auch die Interaktion zwischen diesen mit ihren betreuenden Personen ermöglichen [27]. Wichtig für die Akzeptanz eines solchen Systems ist die Präsentation, die im Rahmen dieses Projektes nicht nur auf den PC beschränkt sein soll, sondern mit Hilfe des *AALuis Transformation Layer* für andere Geräteklassen (TV, Tablets, Smartphones) ermöglicht wird [26].

6.2. USER ADAPTION MODULE

Um den Benutzer mit relevanten Inhalten zu versorgen, wurde für das *User Adaption Module* ein Empfehlungssystem erstellt, das den Benutzern auf einfachste Weise erlaubt neue Inhalte zu entdecken. Dabei sollte nicht nur der Inhalt, sondern auch das Profil des Benutzers neue Empfehlungen ermöglichen. Dies wurde für diese Implementation auf Bewertungen von Filmen durch die einzelnen Benutzer reduziert [27].

Das hier vorgestellte Verfahren ist ein hybrides Empfehlungssystem, das Versucht mit Hilfe von Clustering einerseits die Benutzer aber auch die Inhalte zu

kategorisieren und somit Empfehlungen abgeben.

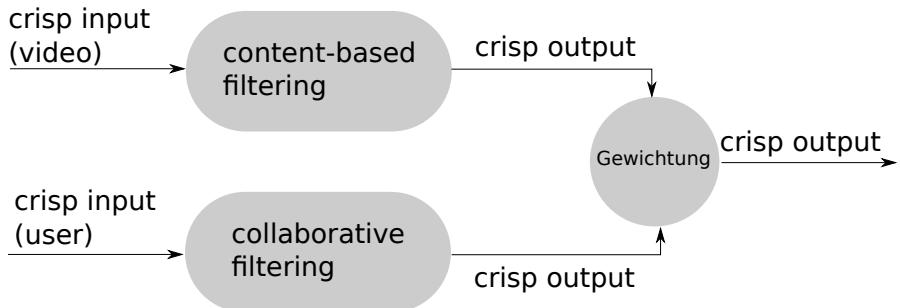


Abbildung 19: Recommender System

Ursprünglich war vorgesehen zweifaches Clustering mit Bestimmung von Zugehörigkeitsfunktionen zu implementieren, dies wurde aber nach Tests mit einfachen Testdaten wieder verworfen. Einerseits ist der Rechenaufwand nicht unerheblich, die Daten zweifach zu Clustern, andererseits ist ein solches Regelsystem nicht zielführend. Die Zuordnung von linguistischen Variablen zu Zugehörigkeitsfunktionen beim content-based Teil des Empfehlungssystem erhöht die Komplexität, aber nicht die Interpretierbarkeit [27].

6.3. DATEN

Die Daten der Filme sind einfache Vektoren K , die die Zughörigkeit zu einer bestimmten Kategorie (als Analogon zu Genres von Filmen) wiedergeben, $K = (k_1, \dots, k_n)$, $k_i \in \{0, 1\}$, während die Benutzerdaten als Matrix R gespeichert werden, die Zeilenweise die Bewertungen der Benutzer für die einzelnen Filme enthält. Die Bewertungen sind in diesem Falle die natürlichen Zahlen zwischen 1 und 5, dies ist aber eine willkürliche Limitation, die einzige wirkliche Einschränkung wäre die Maschinengenauigkeit. Nicht bewerteten Elementen wird 0 zugeordnet. Wir

definieren die Bewertungsmatrix:

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{pmatrix}$$

Wobei r_{ij} die Bewertung des i. Benutzer für den j. Film bedeutet.

6.4. CONTENT BASED FILTERING

Der Content Based Filtering Teil des Empfehlungssystems basiert auf Clustering mit Hilfe des Fuzzy c-Means Algorithmus. Da die Zugehörigkeit zu den jeweiligen Kategorien als Vektor mit Elementen aus $\{0, 1\}$ dargestellt wird, ist der euklidische Abstand nicht gut geeignet um Ähnlichkeit zwischen den einzelnen Objekten zu beschreiben. Deswegen wurde die *Kosinus Ähnlichkeit* aus 2.1 gewählt. Mit Hilfe dieser können wir nun die Funktion $d_{cos} : X \rightarrow [0, 1]$ konstruieren:

$$d_{cos}(x, y) = \frac{\arccos\left(\frac{\langle x, y \rangle}{\|x\|\|y\|}\right)}{\pi}$$

Die Symmetrie ist leicht ersichtlich und folgt aus der Vertauschbarkeit des Skalarprodukts, sowie von $\|\cdot\|$. Außerdem ist sie positiv definit, denn $\arccos(x)$ nimmt für Werte aus $[-1, 1]$ nur Werte in $[0, \pi]$ an. Die Dreiecksungleichung ist mit Hilfe der Cauchy-Schwarzschen Ungleichung für die euklidische Norm leicht ersichtlich, allerdings verletzt es die Voraussetzungen für eine Norm. Dies ist auch leicht zu veranschaulichen, da wir mit $\cos(\theta)$ den Winkel zwischen zwei Vektoren betrachten, der von der Länge unabhängig ist. Auch wenn dadurch nicht gesichert ist, dass wir wirklich ein globales Minimum erhalten [5], kann man eine Distanzfunktion, die nicht von einer Norm bzw. einem Skalarprodukt induziert ist, im Fuzzy c-Means Algorithmus verwendet werden.

Bei der Abfrage nach ähnlichen Filmen, werden nur solche betrachtet, die im selben Cluster wie der vorgegebene Film sind. Die Zugehörigkeit zum Cluster gibt hierbei die Höhe der Bewertung an und wird zwischen 1 und 5 normiert um ein zum collaborativen Filtering vergleichbaren Wert zu erhalten.

6.5. COLLABORATIVE FILTERING

Um Benutzer zu Gruppieren werden die Bewertungen der Benutzer in Vektoren mit Elementen aus $[0, 5]$ geschrieben, wobei 0 bedeutet, dass dieser Film nicht bewertet wurde. Danach wird der Fuzzy c-Means Algorithmus darauf angewendet. Hierbei wird als Distanzfunktion der euklidische Abstand verwendet, da die Kosinusähnlichkeit nur den Unterschied in der Orientierung aber nicht den Größenunterschied zwischen zwei Vektoren beschreibt, dies aber ein essentieller Teil in dieser Klassifizierung ist.

Definition 17 (Euklidischer Abstand). *Sei $X \subset \mathbb{R}^n$. Wir bezeichnen d als (euklidischen) Abstand zwischen den Vektoren $x, y \in X$, wenn*

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (= \|x - y\|_2)$$

Wie man leicht überprüft ist $d(x, y)$ sogar eine Norm, welche von einem Skalarprodukt induziert wurde. Deswegen kann in diesem Fall beim Fuzzy c-Means Algorithmus ein globales Minimum erreicht werden.

Bei Abfrage nach den passendsten Filmen werden nur solche betrachtet, die von mindestens einem anderen Benutzer aus dem selben Cluster bewertet wurden. Von diesen wird entweder die Bewertung des Benutzers selbst übernommen, oder der

Durchschnitt aller Bewertungen der anderen Benutzer in diesem Cluster gebildet und als prognostizierte Bewertung angenommen.

6.6. HYBRIDES VERFAHREN

Um nun zu einer endgültigen Empfehlung zu gelangen werden die jeweiligen Ergebnisse kombiniert und das Mittel genommen. Wenn $S = \{s_1, s_2, \dots\}$ die Menge der ähnlichen Filme bezeichnet, und $U = \{u_1, u_2, \dots\}$ die Menge der von ähnlichen Benutzern bewerteten Filme, so bezeichnen wir die Menge der Empfehlung $E = S \cup U$. Die Bewertungsfunktion der Filme bezüglich des jeweiligen Verfahrens bezeichnen wir mit r_{cb} für das content based Verfahren und r_{cf} für das collaborative Verfahren. Für die endgültige Bewertungsfunktion r_f wird das Mittel der zwei Werte genommen, wobei eine Bewertung von 0 für alle Elemente angenommen wird, die in S oder U nicht vorkommen. Also

$$r_f^i = \frac{r_{cb}(s_i) + r_{cf}(u_i)}{2}$$

Alle so gefundenen Filme werden nach Bewertung sortiert und an das System zurückgegeben. Da man allerdings bei einer großen Anzahl an Daten, auch viele Empfehlungen bekommt, ist es sinnvoll die Anzahl der empfohlenen Filme zu begrenzen.

6.7. IMPLEMENTIERUNG

Bei der Implementation des Fuzzy Empfehlungssystems wurden möglichst viele Teile der Decision Making Engine aus 4.6 wiederverwendet. Details zur **FuzzyCMeans-Clusterer** Klasse sind in 4.6 zu finden. Die Testdaten wurden von kommasepa-

rierten Textdateien eingelesen, wobei aber eine Datenbankanbindung vorgesehen ist.

ÄHNLICHKEITSFUNKTIONEN

Um die Ähnlichkeit der Benutzer zu messen, wurde die normale euklidische Norm aus [2] verwendet, für die Ähnlichkeit der Videos wurde allerdings die *Kosinus Ähnlichkeit* verwendet die als `SimilarityCosine` Klasse implementiert wurde.

COLLABORATIVE SYSTEM

In der `CollaborativeFilteringEngine` Klasse werden die Benutzerdaten geclustert sowie Empfehlungen in der `calculateRecommendations` Methode berechnet.

CONTENT-BASED SYSTEM

Auch die `ContentBasedFilteringEngine` arbeitet ähnlich wie die `CollaborativeFilteringEngine`, allerdings mit Hilfe der *Kosinus Ähnlichkeit* beim Clustering. Da die *Kosinus Ähnlichkeit* nur Werte aus $[0, 1]$ berechnet, werden vor der Rückgabe `calculateRecommendations` die die Bewertungen der Filme auf das Intervall $[0, 5]$ transformiert.

HYBRIDES SYSTEM.

Die `RecommendationEngine` Klasse implementiert den hybriden Teil des Empfehlungssystems. Es kombiniert in der `getRecommendations` die Ergebnisse der Teilsysteme wie in 6.6 beschrieben.

ABFRAGEN

Die Verbindung des Systems an weitere Komponenten der *YouDo* Platform wird durch einen *jetty* Webserver [10] hergestellt. Dieser wird über *http* kontaktiert und sendet die Empfehlungen als *json*-codierte Strings zurück.

HAUPTKLASSE

Die **Advisor** Hauptklasse ist für das Einlesen der Daten, sowie der Weiterleitung der berechneten Empfehlungen an den Webserver zuständig.

Algorithm 6.1: FUZZYRECOMMENDER(*userList, movieList*)

```
procedure GETRECOMMENDATIONS(uId, mId)
  recommList ← RecommendationEngine.getRecommendations(mId, uId)
  return (recommList)
main
  RecommEngine ← newRecommendationEngine(userList, movieList)
  if newquery
    then GetRecommendations(userId, movieId)
```

6.8. ERGEBNISSE UND AUSBLICK

Da sich das *YouDo it - we help!* System erst in Entwicklung befindet, ist eine Evaluation durch die Benutzer noch nicht möglich. Mit einfachen Testdaten scheint dies allerdings ein robuster erster Ansatz zu sein, der aber in vielen Faktoren ausbaufähig ist.

Beim content-basierten Teil des Systems ist ein interessanter Ansatz, alternative Clustering-Algorithmen zu verwenden, sowie andere Distanzfunktionen zu untersuchen. Außerdem ist es ein *cluster-only*-Ansatz, der in Hinblick mit alternativen Kategorisierungsverfahren, leicht mit solchen kombinierbar ist. Die Information des Zugehörigkeitsgrades der einzelnen Objekte zu einem Cluster wird im Moment noch nicht wirklich ausgenutzt. Hier könnte man untersuchen ob ein daraus abgeleiteter Regelbasierter Ansatz bessere Ergebnisse bringt.

Beim collaborativen Teil wären Methoden zur Dimensionsreduktion eine Möglichkeit das System auszubauen. Da dies ein Echtzeitsystem sein soll (das heisst die Ergebnisse müssen möglichst zeitnah an den Benutzer weitergeleitet werden), muss auf die Laufzeit und Effizienz des Systems besonderen Wert gelegt werden. Auch hier stellt sich die Frage nach der Wahl des richtigen Algorithmus. Welches Clustering Verfahren für diese Art der Daten die besten Ergebnisse liefert ist nicht von vornherein klar. Auch scheint die euklidische Distanz, gerade in Hinblick auf die hohe Dimension der Daten, nicht die beste Wahl zu sein (*Curse of dimensionality*) [28].

LITERATUR

- [1] AIT Austrian Institute of Technology GmbH - Biomedical Systems. HOMER - Home Event Recognition System. <http://homer.aaloa.org/>, March 2011.
- [2] Apache. Commons math. <https://commons.apache.org/proper/commons-math/>, 2015.
- [3] Statistik Austria. Bevölkerung. http://www.statistik.at/web_de/statistiken/bevoelkerung/.
- [4] U. Bankhofer and J. Vogel. *Datenanalyse und Statistik*. Gabler, 2008.
- [5] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer, 1981.
- [6] U. Brandes and Erlebach T., editors. *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*. Lecture Notes in Computer Science. Springer, 2005.
- [7] K. Cai and L Zhang. Fuzzy reasoning as a control problem. *Fuzzy Systems, IEEE Transactions on*, pages 600–614, 2008.
- [8] G. Castellano, A. M. Fanelli, and C. Mencar. Generation of interpretable fuzzy granules by a double-clustering technique. *Archives of Control Sciences*, pages 397–410, 2002.
- [9] F. Doctor, H. Hagras, and V Callaghan. An intelligent fuzzy agent approach for realising ambient intelligence in intelligent inhabited environments. *Sys-*

- tems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35:55 – 65, 2005.
- [10] Eclipse. Jetty. <http://www.eclipse.org/jetty/>, 2015.
- [11] N. Good, J.B. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J Riedl. Combining collaborative filtering with personal agents for better recommendations. *Conference of the American Association of Artificial Intelligence (AAAI-99)*, pages 439–446, 1999.
- [12] A. Holmes, H. Duman, and A. Pounds-Cornish. The idorm: Gateway to heterogeneous networking environments. *International ITEA Workshop on Virtual Home Environments*, 2002.
- [13] W. Härdle and L. Simar. *Applied Multivariate Statistical Analysis*. Springer, second edition edition, 2007.
- [14] J. Jang. Anfis: Adaptive-network-based fuzzy inference system. *Systems Man and Cybernetics, IEEE Transactions on*, pages 665 – 685, 1993.
- [15] A. Klahold. *Empfehlungssysteme*. Vieweg+Teubner, 2009.
- [16] O. Kramer. *Computational Intelligence*. Springer, 2008.
- [17] W. Lippe. *Soft-Computing*. Springer, 2006.
- [18] O. Maimon and L. Rokach, editors. *Data Mining and Knowledge Discovery Handbook*. Springer, 2009.
- [19] E. H. Mamdani. Applications of fuzzy algorithms for simple dynamic plant. *Proceedings of the IEEE*, pages 1585–1588, 1974.

- [20] M. Munstermann. *Technisch unterstützte Pflege von morgen*. Springer, 2015.
- [21] Chris D. Nugent, Jit Biswas, and Jesse Hoey. *Activity Recognition in Pervasive Intelligent Environments*. Springer, 2011.
- [22] University of Finland. Clustering datasets. <http://cs.joensuu.fi/sipu/datasets/>, 2015.
- [23] F Ricci, L. Rokach, B. Shapira, and P. Kantor, editors. *Recommender Systems Handbook*. Springer, 2010.
- [24] D. Schröder. *Intelligente Verfahren*. Springer, 2010.
- [25] Shokri Z. Selim and M. A. Ismail. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(1):81–87, January 1984.
- [26] M. Sili, M. Gira, C. Mayer, M. Morandell, and M. Petzold. A framework for the automatic adaptation of user interfaces. *Assistive Technology Research Series, AAATE 2013*, pages 1298 – 1303, 2013.
- [27] Sili, M. and Bollinger, D. and Morak, J. and Gira, M. and Wessig, K. and Brunmeir, D. and Kropf, J. and Tellioğlu, H. YouDo - we help! - An Open Information and Training Platform for Informal Caregivers. *Studies in Health Technology and Informatics*, pages 873 – 877, 2015.
- [28] D. Simivici and C. Djeraba. *Mathematical Tools for Data Mining*. Springer, second edition edition, 2008.
- [29] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, us ed edition, May 2005.

- [30] H. Unbehauen. *Regelungstechnik I*. Vieweg, 14. auflage edition, 2007.
- [31] G. Van Den Broek, F. Cavallo, and C. Wehrmann. *AALIANC Ambient Assisted Living Roadmap*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2010.
- [32] L. Zadeh. Fuzzy sets. *Information and Control*, 8, 1965.
- [33] L. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *Systems, Man and Cybernetics, IEEE Transactions on*, 1973.
- [34] L. Zadeh. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, pages 111–127, 1997.