

Schriftenreihe des Instituts für
Angewandte Informatik / Automatisierungstechnik
Universität Karlsruhe (TH)
Band 6

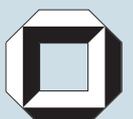
R. Mikut, M. Reischl (Hrsg.)
Institut für Angewandte Informatik /
Automatisierungstechnik



Proceedings

14. Workshop Fuzzy-Systeme und Computational Intelligence

Dortmund,
10.-12. November 2004



Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe

www.uvka.de

© Universitätsverlag Karlsruhe 2004
Print on Demand

ISSN 1614-5267
ISBN 3-937300-20-1

VORWORT

Dieser Tagungsband enthält die Beiträge des 14. Workshops „Fuzzy Systeme und Computational Intelligence“ des Fachausschusses 5.22 „Fuzzy Control“ der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und der Fachgruppe „Fuzzy-Systeme und Soft-Computing“ der Gesellschaft für Informatik (GI), der vom 10.-12. November 2004 im Haus Bommerholz bei Dortmund stattfindet.

Dieser jährliche Workshop bietet ein Forum zur Diskussion neuer methodischer Ansätze und industrieller Anwendungen auf den Gebieten Fuzzy-Systeme, Künstliche Neuronale Netze, Evolutionäre Algorithmen und Data Mining. Besondere Schwerpunkte sind automatisierungstechnische Anwendungen, z.B. in der Verfahrenstechnik, Energietechnik, Kfz-Technik, Robotik und Medizintechnik, aber auch Lösungen in anderen Problemgebieten (z.B. Data Mining für technische und nichttechnische Anwendungen) sind von Interesse.

Die Ergebnisse werden von den ca. 50 Mitgliedern und Gästen aus Hochschulen, Forschungseinrichtungen und der Industrie präsentiert und in Klausuratsmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

Nähere Informationen zum GMA-Fachausschuss erhalten Sie unter

<http://www.iai.fzk.de/medtech/biosignal/gma/index.html>.

Die Herausgeber bedanken sich an dieser Stelle bei allen Autoren und Rednern sowie bei Prof. Dr. Klawonn (Fachhochschule Braunschweig/Wolfenbüttel), Herrn Dr. Runkler (Siemens AG), Herrn Dr. Jäkel (Forschungszentrum Karlsruhe) und Herrn Dr. Kroll (ABB Heidelberg), die maßgeblich an der Vorbereitung des Workshops beteiligt waren.

Ralf Mikut und Markus Reischl

INHALTSVERZEICHNIS

F. Hoffmann <i>Universität Dortmund:</i> Lernen von Roboterverhalten durch Demonstration	1
F. Rehm, F. Klawonn, R. Kruse <i>DLR Braunschweig, FH Braunschweig/Wolfenbüttel, Universität Magdeburg:</i> Ausreißerererkennung mit Fuzzy-Clustering-Methoden	3
J. Jäkel, M. Nöllenburg <i>Forschungszentrum Karlsruhe GmbH:</i> Validation in the Cluster Analysis of Gene Expression Data	13
K. Villforth, S. Schabel <i>TU Darmstadt:</i> Clustermethoden für die Analyse von Siebmustern in der Papierfabrikation	33
R. Palm <i>Universität Örebro, Schweden:</i> Multi-step-ahead prediction in control systems using Gaussian Processes and TS-Fuzzy Models	42
M. Sterneke, P. Krause, H. Kiendl <i>Universität Dortmund, NuTech Solutions GmbH:</i> Erweiterung des Fuzzy-ROSA-Verfahrens für die datenbasierte Regelgenerierung bei kleinen und stark gestörten Datensätzen	55
A. Seeliger, M. Wagenknecht, R. Hampel <i>Hochschule Zittau/Görlitz (FH):</i> Benchmarkproblem: Takagi-Sugeno Fuzzy Modell zur Bestimmung von Parametern in einem Druckbehälter	69
S. Beck, R. Mikut <i>Forschungszentrum Karlsruhe GmbH:</i> Konzepte zur entscheidungstheoretischen Bewertung von Fuzzy- Systemen	70
T. Schmidt, D. Henrich <i>Universität Bayreuth:</i> Temporal erweiterte Prädikate der Fuzzy-Logik zur Überwachung und Wartung	89

X. Wang, D. Nauck, M. Spott, R. Kruse <i>Universität Magdeburg, BT, Research and Venturing, Intelligent Systems Research Centre:</i> Fuzzy Decision Trees – A New CI-Method for the Automatic Data Analysis Platform SPIDA	109
N. Bauersfeld, K.-D. Kramer <i>Hochschule Harz:</i> Gewinnung von Merkmalen für die Clusteranalyse anhand eines selektiven Gassensors für Hochtemperaturanwendungen bis 900°C	124
H. Köpcke <i>Universität Dortmund:</i> Optimierung von Aufstellungsentwürfen von Chemieanlagen durch Constraint Programming	139
E. Pawelka, M. Kuhn, Ch. Gierend <i>MVA Flötzersteig, Wien, IPG GmbH, Grünstadt:</i> Optimierung NH ₄ OH-Eindüsung einer SCR-Anlage mittels Fuzzy Control am Beispiel der MVA Flötzersteig, Wien	140
T. Runkler <i>Siemens AG, München:</i> The fuzzy component assignment problem	149
J. Krone, C. Douven, H. Brinck, U. Lehmann <i>Fachhochschule Südwestfalen, Iserlohn, Fachhochschule Gelsenkirchen, Recklinghausen:</i> Ein genetischer Algorithmus zur Konturerkennung sternförmiger Gebiete	163
M. Reischl, R. Mikut <i>Forschungszentrum Karlsruhe GmbH:</i> Validierung hierarchischer Klassifikatoren für Mehrklassenprobleme	175
W. Kästner, T. Förster, R. Hampel <i>Hochschule Zittau/Görlitz (FH):</i> Bewertung der Netzgüte eines MLP mit Hilfe der Gewichtsanalyse – Potential und Grenzen	189

Lernen von Roboterverhalten durch Demonstration

Frank Hoffmann

Lehrstuhl für Elektrische Steuerung und Regelung
Fakultät für Elektrotechnik und Informationstechnik
Universität Dortmund
44221 Dortmund
E-Mail: hoffmann@esr.e-technik.uni-dortmund.de

Dieser Beitrag präsentiert einen auf Fuzzy Logik basierenden Ansatz zum Erlernen von Roboterverhalten anhand von demonstrierten Fahrten. Dabei werden die einzelnen Roboterverhalten mit den während der Demonstration aufgezeichneten Benutzerkommandos verglichen und an das durch die Trainingsbeispiele demonstrierte Verhalten adaptiert.

In der verhaltensbasierten Robotik wird die Steuerung des Roboters durch Zerlegung in einzelne Verhalten modularisiert. Isoliert betrachtet dienen einzelne Verhalten der Erreichung oder Aufrechterhaltung von Zielen, wie beispielsweise der Kollisionsvermeidung. Jedes Verhalten lässt sich als ein dynamischer Teilregler interpretieren, welcher durch die Sensorik wahrgenommene Eingangsgrößen auf die Ausgangsgrößen der Aktuatoren abbildet. Die Leistungsfähigkeit und Flexibilität verhaltensbasierter Systeme erwächst aus der Kooperation und Koordination einzelner Verhalten zur Bewältigung komplexer Aufgaben. Das Problem der Verhaltenskoordination oder Handlungsauswahl besteht darin, aus den vorgeschlagenen Handlungen der gleichzeitig aktiven Verhalten die vom Roboter letztendlich auszuführende Handlung zu bestimmen. Diese Auswahl basiert neben den vorgeschlagenen Handlungen selber zusätzlich auf der statischen Priorität konkurrierender Verhalten oder dem kontextabhängigen Aktivierungsgrad der Verhalten. Bei der kooperativen Verhaltensfusion ergibt sich die resultierende Handlung aus der mit ihrer Stärke gewichteten Mittelung der Handlungsvorschläge einzelner Verhalten.

Im allgemeinen schlägt jedes Verhalten für den durch die Sensoren wahrgenommenen Kontext eine durch eine scharfe Stellgröße repräsentierte gewünschte Handlung vor. Diese Darstellung besitzt den Nachteil, dass mögliche Alternativen, wie zum Beispiel ein in Fahrtrichtung befindliches Hindernis links oder rechts zu passieren, nicht adäquat repräsentiert werden können. Dementsprechend werden im aktuellen mögliche oder gar erwünschte alternative Aktionen eines Verhaltens bei der Befehlsfusion mit anderen Verhalten nicht berücksichtigt.

Aufgrund dieser Problematik bietet eine auf Fuzzy-Mengen basierende Repräsentation von gewünschten Handlungen deutliche Vorteile. Dabei repräsentiert jedes Roboterverhalten seine Präferenzen durch eine von der aktuellen Wahrnehmung bestimmte Fuzzy-Menge über dem Raum möglicher Handlungen. Die Befehlsfusion ergibt sich aus der anhand der Priorität und Stärke gewichteten Überlagerung der Fuzzy-Präferenzen der Einzelverhalten und anschließender Defuzzifizierung in eine scharfe Ausgangsgröße.

Wie sich zeigt, ermöglicht es erst die Repräsentation gewünschter Handlungen durch Fuzzy-Präferenzen, während der Demonstration aufgezeichnete Kommandos des Bedieners sinnvoll mit den Handlungen einzelner Verhalten zu vergleichen. Diese Ei-

genschaft eröffnet die Möglichkeit, das vorhandene Repertoire an Verhalten an ein durch Vorführung demonstriertes Gesamtverhalten des Roboters anzupassen.

Ein Fuzzy-Verhalten ist charakterisiert durch die parametrisierte Abbildung von wahrgenommenen Sensorinformationen auf Handlungspräferenzen, seinen kontextabhängigen Aktivierungsgrad und seine statische Priorität in der Verhaltenshierarchie. Die Lernaufgabe besteht darin, die Parameter der einzelnen Verhalten und ihre Aktivierungsfunktionen an die aufgezeichneten Trainingsdaten anzupassen. Eine besondere Schwierigkeit ergibt sich aus der Tatsache, dass demonstrierte Handlungen nicht immer eindeutig sind, das heißt, die benutzergesteuerten Fahrten enthalten für gleich oder ähnlich wahrgenommene Zustände unterschiedliche Handlungen.

Die Adaption der einzelnen Verhalten kann nicht mit herkömmlichen überwachten Lernverfahren erfolgen, da diese je nach Aktivierungsgrad und Priorität mit unterschiedlichem Gewicht zur aus der Fusion resultierenden Gesamthandlung beitragen. Stattdessen werden die Fuzzy-Regeln der Einzelverhalten im Kontext aller gleichzeitig aktiven Verhalten beurteilt. Als Gütemaße dienen die Plausibilität und die Spezifität eines Verhaltens. Plausibilität bedeutet anschaulich, dass für in den Trainingsbeispielen beobachtete Handlungen aktive Verhalten eine hohe Präferenz aufweisen. Spezifität bedeutet umgekehrt, dass ein aktives Verhalten für nicht beobachtete Handlungen eine niedrige Präferenz besitzt. Beide Gütekriterien lassen sich in dem Satz "Präferenzen für Handlungen sollen so allgemein wie nötig, aber so spezifisch wie möglich sein" zusammenfassen.

Der Beitrag beschreibt zunächst die Grundlagen zur präferenzbasierten Repräsentation von Verhalten und deren prioritäts- und kontextabhängigen Koordination. Die grundlegende Methodik wird am Beispiel eines Hindernisvermeidungsverhaltens illustriert. Der nächste Abschnitt beschäftigt sich mit der Adaptation von Verhalten und ihrer Koordination mittels Lernen durch Demonstration. Er behandelt die Definition geeigneter Gütemaße und deren Optimierung mit Hilfe einer biologisch inspirierten Partikelschwarmmethode. Der letzte Abschnitt präsentiert auf simulierten und realen Fahrten des Roboters basierende Ergebnisse und Analysen der Praxistauglichkeit der vorgeschlagenen Methode.

Ausreißerererkennung mit Fuzzy-Clustering-Methoden

Frank Rehm

DLR Braunschweig
Institut für Flugführung
Lilienthalplatz 7
38108 Braunschweig
Tel.: (0531) 295 2577
Fax: (0531) 295 2899
E-Mail: frank.rehm@dlr.de

Frank Klawonn

FH Braunschweig/Wolfenbüttel
Fachbereich Informatik
Salzdahlumer Str. 46/48
38302 Wolfenbüttel
Tel.: (05331) 939 6111
Fax: (05331) 939 6002
E-Mail: f.klawonn@fh-wolfenbuettel.de

Rudolf Kruse

Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2
39106 Magdeburg
Tel.: (0391) 67 18706
Fax: (0391) 67 12018
E-Mail: kruse@iws.cs.uni-magdeburg.de

1 Einleitung

In vielen Bereichen der Wissensentdeckung in Datenbanken ist das Finden von Ausreißern, Werten die stark vom Rest der Daten abweichen, von großer Bedeutung. Selbst wenige Ausreißer können die Ergebnisse statistischer Methoden wie die Mittelwertschätzung aber auch Clustering-Verfahren wie *k-means* oder *fuzzy c-means* verzerren [6, 12]. Neben einigen auf Distanz- und Dichtemaßen basierenden Ansätzen [3, 14] zur Erkennung von Ausreißern wurden auch Methoden entwickelt, die für prototypbasierte Clustering-Verfahren geeignet sind [4, 5, 10, 13, 15]. In dieser Arbeit geben wir einen Überblick über verschiedene prototypbasierte Verfahren zur Ausreißerererkennung und präsentieren Ergebnisse zweier neuerer Ansätze [13, 16]. Im folgenden Abschnitt beschreiben wir kurz den Fuzzy c-Means-Algorithmus [2], auf dem die Verfahren zur Ausreißerbehandlung dieser Arbeit basieren. Abschnitt 3 befaßt sich mit einem possibilistischem Fuzzy-Clustering-Ansatz [15], der sich zur Ausreißerererkennung eignet. In Abschnitt 4 zeigen wir einen Ansatz der beim Noise-Clustering [4, 5], einer Fuzzy-Clustering-Methode, die Ausreißer während des Clustering-Prozesses berücksichtigt, gute Ergebnisse erzielt. In Abschnitt 5 beschreiben wir eine Methode, bei der ein robustes statistisches Verfahren zur Ausreißerererkennung [8] auf prototypbasierte Clustering-Verfahren angewendet wird.

2 Fuzzy c-Means

Fuzzy-Clustering eignet sich zum Finden von Strukturen in Daten. Datensätze werden dabei in eine Menge von Clustern eingeteilt. Im Gegensatz zum Hard-Clustering, wird ein Datum beim Fuzzy-Clustering allen Clustern zu einem bestimmten Grad zugeordnet. Ein gängiger Vertreter, der Fuzzy c-Means-Algorithmus (FCM), minimiert die Zielfunktion J , welche die Summe der gewichteten Distanzen d_{ij} zwischen den c Prototypvektoren v_i und den n Merkmalsvektoren x_j im Merkmalsraum \mathfrak{R}^p beschreibt:

$$J = \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m d_{ij}. \quad (1)$$

Der Parameter u_{ij} gibt den Zugehörigkeitsgrad eines Merkmalsvektors x_j zu einem Prototypvektor v_i an. Mit dem Unschärfeparameter (*fuzzyfier*) $m \in [1, \infty]$ kann man bestimmen, wie stark die Überlappung der Cluster ist. Während große Werte für m zu stark überlappenden Clustern führen, erzielt man mit kleinen Werten nahe 1 eher scharfe Clustereinteilungen.

Um die triviale Lösung zu vermeiden, keinen Merkmalsvektor einem Cluster zuzuweisen indem alle u_{ij} gleich 0 gesetzt werden, gelten folgende Randbedingungen:

$$u_{ij} \in [0, 1] \quad 1 \leq i \leq c, \quad 1 \leq j \leq n \quad (2)$$

$$\sum_{i=1}^c u_{ij} = 1 \quad 1 \leq j \leq n \quad (3)$$

$$0 < \sum_{j=1}^n u_{ij} < n \quad 1 \leq i \leq c. \quad (4)$$

Häufig wird die Euklidische Norm als Distanzmaß zwischen Prototypvektoren v_i und Merkmalsvektoren x_j verwendet:

$$d_{ij} = d^2(v_i, x_j) = (x_j - v_i)^T (x_j - v_i).$$

Modifikationen des FCM, bei denen ein anderes Distanzmaß, z.B. die Mahalanobis-Distanz, verwendet wird, ermöglichen es, unterschiedliche Clusterformen zu erkennen [7].

Die Minimierung der Zielfunktion in Gleichung (1) repräsentiert ein nichtlineares Optimierungsproblem, das gewöhnlich mit Hilfe von Lagrange Multiplikatoren gelöst wird, indem ein alternierendes Optimierungsschema angewendet wird [1]. Dabei wird abwechselnd eine der gesuchten Parametermengen, entweder die Zugehörigkeitswerte

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{1}{m-1}}} \quad (5)$$

oder die Prototypvektoren

$$v_i = \frac{\sum_{j=1}^n (u_{ij})^m x_j}{\sum_{j=1}^n (u_{ij})^m} \quad (6)$$

festgesetzt, während der andere Parameter optimiert wird, solange bis der Algorithmus letztlich konvergiert. Mit der Euklidischen Distanz findet der FCM hyperkugelförmige Cluster annähernd gleicher Größe.

3 Possibilistische Clusteranalyse

Beim FCM werden probabilistische Zugehörigkeitswerte verwendet. Für jedes Datum ist die Summe der Zugehörigkeitswerte zu allen Clustern gleich 1. Nachteilig bei probabilistischen Fuzzy-Clustering-Verfahren ist, daß sich die Zugehörigkeitswerte nicht immer als Wert interpretieren lassen, der angibt, wie typisch ein Datum für ein bestimmtes Cluster ist. Abbildung 1 veranschaulicht dieses Problem. Der Datensatz enthält zwei Cluster. Bei der Clusteranalyse mit dem probabilistischen FCM werden x_1 und x_2 beiden Clustern jeweils mit Zugehörigkeitswert 0.5 zugeordnet, obwohl x_2 deutlich weiter von den Clustern entfernt ist. Je nach fuzzifizier erhält man für x_3 hohe Zugehörigkeitswerte von über 0.95 zum linken Cluster. Objektiv betrachtet handelt es sich bei diesem Wert eher um einen Ausreißer.

Beim possibilistischen Clustern (PCM) läßt man die Nebenbedingung (3) fallen. Dadurch kann die Summe der Zugehörigkeitswerte zu überlappenden Clustern größer als 1 und bei Ausreißern kleiner als 1 sein. Um die triviale Lösung des Optimierungsproblems zu vermeiden, bei der alle Zugehörigkeitswerte gleich 0 gesetzt werden, modifiziert man die Zielfunktion zu

$$J = \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m d_{ij} + \sum_{i=1}^c \eta_i \sum_{j=1}^n (1 - u_{ij})^m. \quad (7)$$

Der erste Term bewirkt, daß die Abstände der Merkmalsvektoren zu den Prototypvektoren minimiert werden und entspricht der Optimierungsfunktion des probabilistischen Clusters. Der zweite Term bewirkt, daß u_{ij} so groß wie möglich ausfällt. Mit η_i spezifiziert man den Abstand, bei dem der Zugehörigkeitsgrad zum Cluster v_i 0.5 betragen soll und gibt damit die geschätzte Größe des Clusters an. Für die Zielfunktion (7) ergibt sich folgende Berechnungsvorschrift für die Zugehörigkeitswerte

$$u_{ij} = \frac{1}{1 + \left(\frac{d_{ij}}{\eta_i}\right)^{\frac{1}{m-1}}}. \quad (8)$$

Die Berechnung der Prototypvektoren erfolgt wie bei dem FCM. In [15] wird vorgeschlagen, η_i durch

$$\eta_i = K \frac{\sum_{j=1}^n u_{ij}^m d_{ij}}{\sum_{j=1}^n u_{ij}} \quad (9)$$

zu schätzen. K wird üblicherweise auf 1 gesetzt. Die Schätzung von η_i kann auf einer zuvor durchgeführten probabilistischen Fuzzy-Clusteranalyse mit dem FCM basieren. Im Gegensatz zum FCM, bei dem die partitionierende Eigenschaft bei der Berechnung der Zugehörigkeitswerte deutlich wird, beschreiben die Zugehörigkeitswerte beim PCM, wie typisch ein Merkmalsvektor für das jeweilige Cluster ist. Der PCM ist weitgehend von der Initialisierung abhängig. Verwendet man den FCM zur Initialisierung der Prototypen und der Zugehörigkeitswerte, können nach Anwendung des PCM je nach gewählter Anzahl der Prototypen mehrere Cluster identisch sein. Man kann den Wert von η_i für alle Iterationen fixieren oder aber in jeder Iteration neu schätzen, was jedoch zu Instabilitäten führen kann. Ein häufig verwendeter

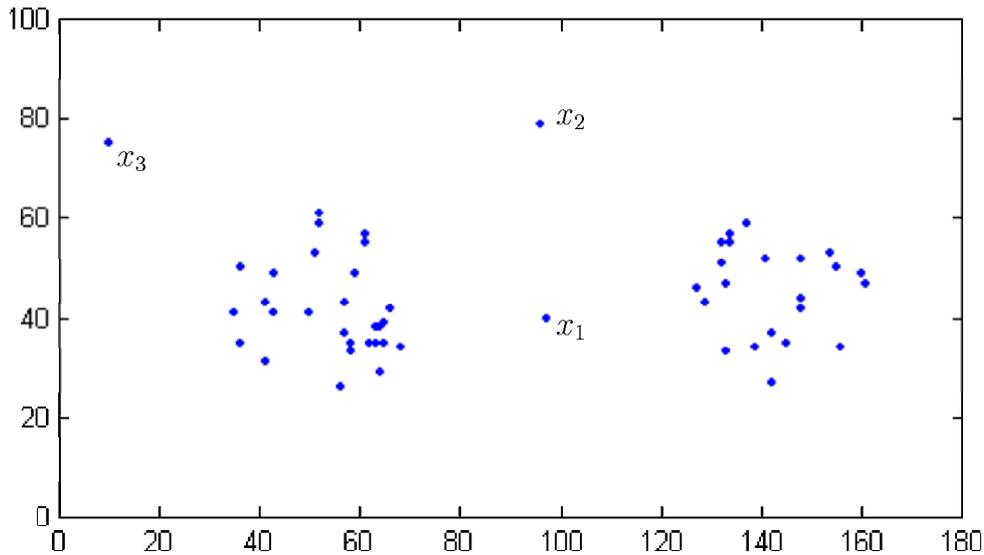


Abbildung 1: Datensatz mit zwei Clustern und drei Ausreißern

Ansatz ist, den PCM mit den initialen Werten aus der probabilistischen Fuzzypartition durchzuführen, anschließend η_i neu zu berechnen, und einen nachfolgenden Lauf mit den korrigierten Werten durchzuführen.

Die Clusteranalyse mit dem PCM liefert für die kritischen Punkte in Abbildung 1 insgesamt niedrige Zugehörigkeitswerte und erkennt diese damit als Ausreißer. Die Zugehörigkeitswerte von x_2 sind noch einmal um die Hälfte kleiner als die von x_1 und spiegeln dadurch die größere Entfernung zu beiden Clustern wider. x_3 erhält zu beiden Clustern sehr niedrige Zugehörigkeitswerte von < 0.1 in der Summe. Alle Merkmalsvektoren, die man auch intuitiv einem der beiden Clustern zuordnen würde, erhalten große Zugehörigkeitswerte zum jeweiligen Cluster und sehr kleine Werte zum jeweils anderen Cluster.

Der PCM eignet sich in einigen Fällen sehr gut zur Ausreißerererkennung, besonders wenn klar abgrenzbare Strukturen im Datensatz vorhanden sind und eine geeignete Initialisierung getroffen wurde. Häufig werden jedoch kompakte Cluster nicht mehr richtig erkannt und zu viele Merkmalsvektoren als Ausreißer klassifiziert, wenn η_i unterschätzt oder mit zu wenigen Prototypen initialisiert wurde.

4 Noise-Clustering

Ein weiterer Ansatz, Ausreißer zu klassifizieren, ist die Verwendung eines sogenannten Noise-Clusters, dem alle Ausreißer zugeordnet werden. Bei diesem Ansatz, der als *Noise-Clustering* (NC) bezeichnet wird und auf dem FCM basiert, wird für jeden Merkmalsvektor neben den Zugehörigkeitswerten zu den Prototypvektoren noch ein weiterer Zugehörigkeitswert zu einem virtuellen Cluster berechnet. Dieses Noise-Cluster, repräsentiert durch Prototyp v_c , hat per Definition einen festen Abstand zu allen Daten

$$d_{cj} = \delta, \quad \forall j.$$

Somit erhalten Merkmalsvektoren, deren Distanz zu den regulären Clustern größer als diese *Noise-Distanz* δ ist, große Zugehörigkeitswerte zum Noise-Cluster.

Die Zielfunktion für das Noise-Clustering gleicht der von FCM und schließt das Noise-Cluster mit ein:

$$J_N = \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m d_{ij} \quad (10)$$

wobei die Abstände zu den regulären Clustern definiert sind durch

$$d_{ij} = d^2(v_i, x_j) = (x_j - v_i)^T (x_j - v_i), \quad i = 1, \dots, c-1$$

und zum Noise-Cluster durch

$$d_{ij} = d^2(v_i, x_j) = \delta, \quad i = c.$$

Die Berechnung der Zugehörigkeitswerte u_{ij} und der Prototypvektoren v_i erfolgt analog zu der beim FCM. Da Merkmalsvektoren mit großer Distanz zu den Clustern beim Noise-Clustering geringere Zugehörigkeitswerte erhalten, beeinflussen diese die Positionierung der Prototypvektoren weniger. Mit Hilfe der Zugehörigkeitswerte zum Noise-Cluster können die Ausreißer klassifiziert werden.

Ein wichtiges Problem bei diesem Verfahren ist die Bestimmung der Noise-Distanz δ , die meist nicht im voraus bekannt ist. Wählt man δ zu groß, hat dieser Ansatz keine bedeutsame Auswirkung auf die Clustereinteilung. Ein zu klein gewähltes δ führt dazu, daß zu viele Daten dem Noise-Cluster zugeordnet werden und dadurch nicht mehr bei der Positionierung der Prototypvektoren berücksichtigt werden. In [4] wird vorgeschlagen, die Noise-Distanz mit dem mittleren Abstand aller Merkmalsvektoren zu den Clustern zu schätzen

$$\delta = \lambda \frac{\sum_{i=1}^{c-1} \sum_{j=1}^n d_{ij}}{n(c-1)}. \quad (11)$$

Im Vergleich zum PCM, bei dem für jedes Cluster ein η_i die zulässige Ausdehnung des Clusters i bestimmt, gibt es beim Noise-Clustering nur einen globalen Wert δ . Mit dem probabilistischen Basismodell findet der NC-Algorithmus stabile Clustereinteilungen. Die Schätzung der Noise-Distanz mit Gleichung (11) führt jedoch dazu, daß mit wachsender Anzahl Prototypen, die Noise-Distanz überschätzt wird und in der Konsequenz nur sehr wenig Ausreißer gefunden werden.

4.1 Schätzung der Noise-Distanz

Die Schätzung der Noise-Distanz ist von zentraler Bedeutung beim Noise-Clustering. Wie bereits erwähnt, kann eine ungünstig geschätzte Noise-Distanz zur fehlerhaften Erkennung von Ausreißern führen. Schätzt man die Noise-Distanz mit Gleichung (11), ist die Erkennung von Ausreißern stark von der verwendeten Anzahl von Prototypen abhängig. Da der Noise-Clustering-Algorithmus auf dem FCM basiert, bewirkt die Verwendung vieler Prototypen für das Clustering eine stärkere Partitionierung des Datensatzes. In der Folge werden auch einzelne kleine Cluster durch Prototypen

approximiert. Da aber in Gleichung (11) der mittlere Abstand aller Merkmalsvektoren zu allen Clustern berücksichtigt wird, wird die Noise-Distanz in solchen Fällen überschätzt.

In [16] haben wir einen Ansatz zur Schätzung der Noise-Distanz vorgestellt, der auf der Erhaltung des Hypervolumens des Merkmalsraumes basiert. Ausgehend von der Annahme, daß bei einem Merkmalsraum, in dem die Daten gleichverteilt sind, die Summe der Volumen der einzelnen Cluster gleich dem Volumen des gesamten Merkmalsraumes ist, wird hier die Noise-Distanz gleich dem Radius eines Clusters gesetzt. Die Beispiele in Abbildung 2 veranschaulichen diesen Ansatz.



Abbildung 2: Auswirkungen der Noise-Distanz bei verschiedenen Datenverteilungen

Abbildung 2.a zeigt einen gleichverteilten 1-dimensionalen Datensatz, der mit zwei Prototypen geclustert wurde. Hier sind alle Merkmalsvektoren weniger als die Noise-Distanz vom Clusterzentrum entfernt. Kein Datum wird hier als Ausreißer gekennzeichnet. In dem Beispiel, das Abbildung 2.b zeigt, enthält der Datensatz zwei Regionen hoher Dichte und einen Punkt, der sich relativ weit entfernt von den restlichen Daten befindet. Da der Merkmalsraum die gleiche Ausdehnung hat, wie der im linken Beispiel, führt die gleiche Noise-Distanz hier dazu, daß dieser Punkt, der weiter als die Noise-Distanz von beiden Clusterzentren entfernt ist, einen großen Zugehörigkeitswert zum Noise-Cluster erhält und dadurch als Ausreißer klassifiziert wird. Für hochdimensionale Datensätze kann dieser Ansatz analog angewendet werden.

Ein weiteres Beispiel demonstriert diesen Ansatz an einem 2-dimensionalen Datensatz. Abbildung 3 zeigt ein gebogenes Cluster, das von einigen Stördaten umgeben ist. Der Merkmalsraum wurde mit vier Prototypen approximiert. Diese sind jeweils in der Abbildung mit einem Viereck gekennzeichnet. Beide Abbildungen zeigen die klassifizierten Ausreißer mit einem Kreuz gekennzeichnet. In Tabelle 1 sind die Ergebnisse der Ausreißerererkennung mit unterschiedlicher Anzahl von Prototypen aufgelistet.

#Prototypen	NC mit konventioneller Noise-Distanz		NC mit Hypervolumen erhaltender Noise-Distanz	
	$\sqrt{\delta}$	#Ausreißer	$\sqrt{\delta}$	#Ausreißer
2	80	36	79	37
3	84	11	65	29
4	85	5	56	30
6	86	0	46	28

Tabelle 1: Noise-Clustering mit unterschiedlicher Anzahl von Prototypen

Wie aus der Tabelle ersichtlich ist, sinkt die Anzahl der gefundenen Ausreißer bei wachsender Anzahl von Prototypen, wenn die Noise-Distanz mit Gleichung (11) geschätzt wird. Im Gegensatz dazu liefert der Ansatz, die Noise-Distanz bei Erhaltung des Volumens zu schätzen, auch bei unterschiedlicher Prototypanzahl vergleichbare

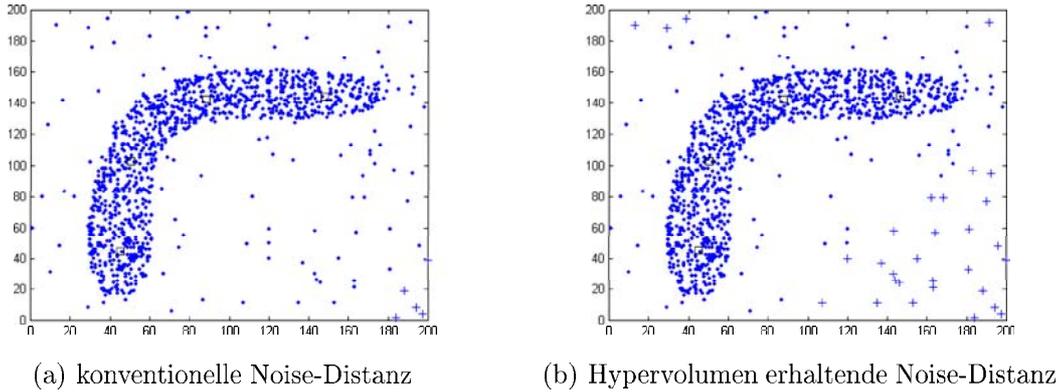


Abbildung 3: Noise-Clustering mit unterschiedlichem δ

Ergebnisse. Da sich die Prototypen in den Regionen hoher Datendichte positionieren, sind die Merkmalsvektoren in den dünnbesetzten Regionen relativ weit entfernt von den Clusterzentren. Sie erhalten entsprechend große Zugehörigkeitswerte zum Noise-Cluster und werden dadurch als Ausreißer klassifiziert.

5 Robuste Ausreißerererkennung mit Clustering-Methoden

In diesem Abschnitt beschreiben wir kurz eine Methode zur Ausreißerererkennung, die wir bereits in [13] vorgestellt haben. Bei diesem Ansatz, der eine Adaption des Verfahrens von Grubbs [8] und Modifikation eines in [17] vorgestellten Verfahrens darstellt, werden zwei Schritte nacheinander durchgeführt; ein Clustering-Algorithmus zum Partitionieren des Datensatzes und eine nachfolgende Ausreißerererkennung für jedes einzelne Cluster. Die Partitionierung des Datensatzes kann beispielsweise mit dem FCM, k-means oder einem anderen prototypbasierten Clustering-Verfahren durchgeführt werden. Die Merkmalsvektoren werden dem Prototypvektor zugeordnet, zu dem die größten Zugehörigkeitswerte ermittelt worden. Nach der Partitionierung der Daten werden für die Ausreißerererkennung nur noch die Merkmalsvektoren betrachtet, die zu dem jeweiligen Cluster gehören. Für jedes Attribut a aller n_i Merkmalsvektoren des korrespondierenden Clusters v_i werden Mittelwert

$$v_i^{(a)} = \frac{1}{n_i} \sum_{x_j \in v_i} x_j^{(a)}$$

und Standardabweichung

$$s_i^{(a)} = \sqrt{\frac{1}{n_i - 1} \sum_{x_j \in v_i} (x_j^{(a)} - v_i^{(a)})^2}$$

berechnet. Für den Merkmalsvektor mit der größten Distanz zum Clusterzentrum, also dem Datum, das als möglicher Ausreißer betrachtet wird, wird der Wert der z-Transformation einer jeden Komponente des Vektors mit einem, für jedes Attribut zuvor spezifizierten, kritischen Wert verglichen. Der kritische Wert für jedes Attribut ist abhängig von der Definition eines Ausreißers. Ein typisches Kriterium ist die maximale Anzahl der Ausreißer relativ zur Gesamtdatenmenge [11]. Letztlich führen große kritische Werte zu einer geringen Anzahl von erkannten Ausreißern und

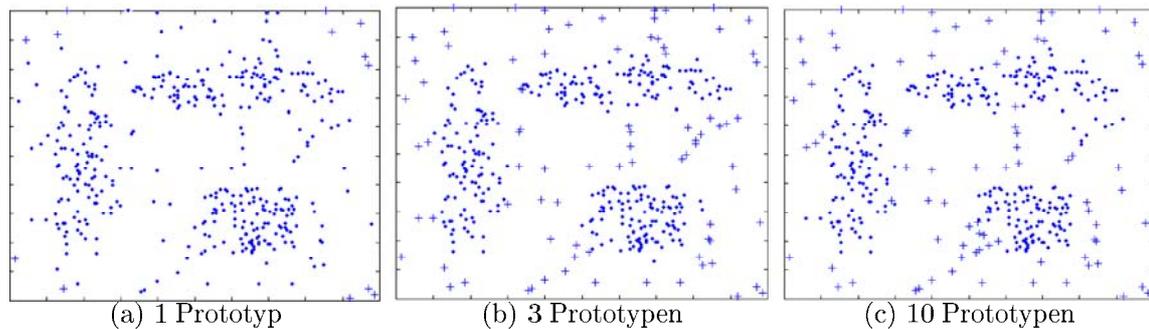


Abbildung 4: Ausreißerererkennung mit unterschiedlicher Anzahl von Prototypen

kleine kritische Werte zu besonders kompakten Clustern. Bemerkt sei, daß für jedes Attribut ein separater kritischer Wert festgelegt werden kann. Durch die attributweise Überprüfung der Daten erhält das Verfahren eine achsen-parallele Sichtweise auf die Daten, was in Fällen von achsen-parallelen Clustern, wie sie etwa durch eine Hauptachsentransformation begünstigt werden, zu einer besseren Ausreißerererkennung führt, als die Sicht auf (hyper)kugelförmige Cluster.

Wenn ein Ausreißer erkannt wurde, wird der Merkmalsvektor von dem Datensatz entfernt. Mit der neuen Datenmenge werden Clusterzentrum und Standardabweichung für alle Attribute in diesem Cluster neu bestimmt. Mit dem Merkmalsvektor, der die größte Distanz zu dem neuen Zentrumsvektor hat, wird der Ausreißertest erneut durchgeführt. Diese Prozedur wird solange wiederholt, bis kein Ausreißer mehr gefunden wird. Die Ausreißerbehandlung wird für die anderen Cluster in gleicher Weise durchgeführt.

Abbildung 4 zeigt die Ergebnisse dieser Methode. Die gefundenen Ausreißer sind in den Abbildungen mit einem Kreuz gekennzeichnet. Erwartungsgemäß werden nur wenige Merkmalsvektoren als Ausreißer klassifiziert, wenn der Merkmalsraum nur mit einem Prototyp approximiert wird. In diesem Fall befindet sich der Prototyp im Schwerpunkt aller Daten. Demzufolge werden nur Punkte am Rand des Merkmalsraumes als Ausreißer erkannt. Vergleicht man die Ergebnisse der Partitionen mit drei und zehn Prototypen, kann man feststellen, daß diese beiden Lösungen nahezu identisch sind.

Dieser Ansatz ist weitgehend unabhängig von der Anzahl der verwendeten Prototypen. Dadurch können auch komplizierte Clusterstrukturen, die meist nur mit einer großen Anzahl von Prototypen approximiert werden können, von Ausreißern unterschieden werden. Bei hochdimensionalen Daten sind dementsprechend viele Parameter zu setzen. In den meisten Fällen ist jedoch kein a-priori Wissen über die einzelnen Attribute verfügbar. Dann kann ein einheitlicher kritischer Wert für alle Attribute definiert werden. Verwendet man einen Fuzzy-Algorithmus wie den FCM zum Partitionieren der Daten, kann man auch einen Ausreißertest für das Cluster durchführen, zu dem ein Merkmalsvektor den zweithöchsten Zugehörigkeitswert aufweist. Der betreffende Merkmalsvektor kann unter Umständen in einem der beiden Cluster als Ausreißer klassifiziert werden und bezüglich des anderen nicht. Da der FCM vorzugsweise (hyper)kugelförmige Cluster findet, kann ein solches Vorgehen zu robusteren Ergebnissen führen.

6 Zusammenfassung

In dieser Arbeit haben wir einige Ansätze zur Ausreißerererkennung mit prototypbasierten Fuzzy-Clustering-Verfahren diskutiert. Anhand von einigen einfachen Beispielen konnten wir kurz die Stärken und Schwächen der vorgestellten Verfahren beleuchten. Für die Noise-Clustering-Technik haben wir einen Ansatz vorgestellt, mit dem die Noise-Distanz geschätzt werden kann und konnten zeigen, daß dadurch stabile Ergebnisse erzielt werden können. Einige Probleme haben jedoch alle hier vorgestellten Verfahren gemeinsam. Wenn die Anzahl der Cluster in einem Datensatz unbekannt ist, muß erhöhter Rechenaufwand betrieben werden, um die optimalen Einstellungen für die diversen Parameter zu finden. Abschließend läßt sich feststellen, daß eine Reihe von Verfahren zur Ausreißerererkennung zur Verfügung steht. Bei der Auswahl der entsprechenden Methode muß in jedem Fall das Datenmaterial und die anwendungsspezifische Definition von Ausreißern berücksichtigt werden.

Literatur

- [1] Bezdek, J.C.: A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, S. 1-8, 1980.
- [2] Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.
- [3] Breunig, M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD 2000)*, Dallas, S. 93-104, 2000.
- [4] Dave, R.N.: Characterization and detection of noise in clustering. *Pattern Recognition Letters*, Vol. 12, S. 657-664, 1991.
- [5] Dave, R.N., Krishnapuram, R.: Robust Clustering Methods: A Unified View. *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 2, S. 270-293, 1997.
- [6] Estivill-Castro, V., Yang, J.: Fast and Robust General Purpose Clustering Algorithms. *Data Mining and Knowledge Discovery*, 8, S. 127-150, Kluwer Academic Publishers, Netherlands, 2004.
- [7] Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, S. 773-781, 1989.
- [8] Grubbs, F.: Procedures for Detecting Outlying Observations in Samples. *Technometrics*, Vol. 11, No. 1, S. 1-21, 1969.
- [9] Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. John Wiley & Sons, Chichester, 1999.
- [10] Keller, A.: Fuzzy Clustering with outliers. In T. Whalen (editor), *PeachFuzz 2000, 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, S. 143-147, Atlanta, 2000.

- [11] Klawonn, F.: Noise Clustering with a Fixed Fraction of Noise. In: A. Lotfi, J.M. Garibaldi: *Applications and Science in Soft Computing*, S. 133-138, Springer, Berlin, 2004.
- [12] Klawonn, F., Georgieva, O.: Identifying Single Clusters in Large Data Sets. In Wang, J. (editor) *The Encyclopedia of Data Warehousing and Mining*, Montclair State University, 2005.
- [13] Klawonn, F., Rehm, F.: Clustering techniques for outlier detection. In Wang, J. (editor) *The Encyclopedia of Data Warehousing and Mining*, Montclair State University, 2005.
- [14] Knorr, E.M., Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. *Proceedings 24th International Conference Very Large Data Bases (VLDB)*, New York, 1998.
- [15] Krishnapuram, R., Keller, J. M.: A Possibilistic Approach to Clustering. *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 2, S. 98-110, 1993.
- [16] Rehm, F., Klawonn, F., Kruse, R.: New Approches to Noise Clustering for Detecting Outliers. (angenommen bei) *27th German Conference on Artificial Intelligence*, Ulm, 2004.
- [17] Santos-Pereira, C. M., Pires, A. M.: Detection of Outliers in Multivariate Data: A Method Based on Clustering and Robust Estimators. In W. Härdle & B. Rönz (editors) *Proceedings in Computational Statistics: 15th Symposium held in Berlin*, S. 291-296, Physica-Verlag, Heidelberg, 2002.

Validation in the Cluster Analysis of Gene Expression Data

Jens Jäkel, Martin Nöllenburg

Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik
Postfach 3640, D-76021 Karlsruhe
Tel.: (07247) 825726, Fax: (07247) 825786, E-Mail: jens.jaekel@iai.fzk.de

1 Introduction

In the early 1990s, a new technology called *DNA microarrays* was developed that allows for a simultaneous measurement of the expression levels, i. e. activity levels, of several thousand genes at specific conditions. During the last years many genome-scale experiments for various species and conditions were conducted. Most of the resulting data is publicly available in internet databases.

There are several questions that can be tackled by analysing data from microarray experiments. In the research field of functional genomics one goal is to annotate genes with their respective functions. This can be approached by gene expression analysis using the hypothesis that co-expressed genes are often co-regulated and thus might share a common function or are at least involved in the same biological process. So identifying groups of co-expressed genes over a number of different experiments can give hints for the putative functions of genes whose function were previously unknown by looking at the known functions of genes in the same group. Another example is the inference of regulatory networks and the identification of regulatory motifs in the sequences from groups of co-expressed genes.

For all these questions it is necessary to identify genes with similar expression patterns, which is usually done using different clustering techniques known from the area of unsupervised machine learning.

After the publication of the first large scale cluster analysis by Eisen et al. [8] many different approaches for the clustering of gene expression data have been made and proven to be successful in their respective situations. Nevertheless, no single clustering algorithm, similarity measure or validation criterion has yet become accepted as being the optimal choice for clustering genes based on microarray data. Former general results on clustering in literature cannot be taken over in any case because many of the theoretical studies assume well separable data. This is not the case for microarray data since clusters often overlap and cannot be easily identified. Some of the problems in clustering gene expression data are discussed in [4]. Only few works systematically evaluate and compare different clustering methods and results. In [7] six clustering algorithms are compared, but the choice of the number of clusters or the dissimilarity measure is not addressed. [3] discusses three validation indices and evaluates them on Kohonen's Self Organizing Maps algorithm. [9] presents a framework for validation of clusterings using external biological information. In [22] an overview of several clustering algorithms and dissimilarities for microarray data is given.

In this article we show on an example how to select good clusterings step-by-step based on several validation criteria. This includes choice of the algorithm, the dissimilarity measure and the number of clusters. In Section 2 some basic background on microarray technology and the pre-processing of microarray data is given. Next, in Section 3 different dissimilarity measures for gene expression profiles are introduced followed by the description of two popular clustering algorithms and the discussion of several validation techniques. Section 4 finally gives the results of applying these clustering algorithms

on a sample data set. The resulting clusterings are systematically compared and several candidates are selected for further visual and external validation using biological information about the clustered genes. Conclusions are presented in Section 5.

2 Measuring gene expression with microarrays

Gene expression is the process by which a gene's information is converted into a functional protein of a cell. It involves two main steps according to the central dogma of molecular biology: The section of the DNA corresponding to the gene is first transcribed into a single-stranded complementary messenger RNA (mRNA) molecule. Thereafter the mRNA is translated into a protein.

It is widely believed that regulation of gene expression is largely controlled at the transcriptional level [29], so studying the abundance of the mRNA can give insight into how the corresponding genes control the function of the cell. Microarrays are a new techniques to measure the mRNA abundance for several thousand genes in parallel.

Microarrays are based on the fact that two complementary DNA (or RNA) molecules can hybridize, i. e. they can bind together. When the sequence of the genome (entirety of all genes) of an organism is known, it is possible to synthesize millions of copies of DNA fragments of each gene. These are then able to hybridize with the corresponding complementary molecules.

Microarrays are small glass slides with thousands of spots printed on it in a grid-like fashion. Each spot corresponds to one gene and consists of thousands of identical and gene-specific single-stranded DNA sequences fixed to the glass surface. The mRNA abundance in a sample is measured indirectly by reverse transcribing the mRNA into cDNA¹. Then the cDNA molecules are able to stick to the spots on the microarray that correspond to their respective genes. Although measuring transcript abundance is not exactly the same as measuring gene expression, it is very common to use both terms synonymously.

Usually the abundance of mRNA transcripts is measured relatively to a control sample. To this end the cDNA prepared from the mRNA of the experiment sample is labeled with a fluorescent dye (usually red Cy5) and the control cDNA is labeled with green-fluorescent dye (Cy3) or vice versa. When both samples are mixed at equal amounts and washed over the glass slide the target cDNA will hybridize on the spot with its complementary sequences (called probes).

Each dye can emit light at a specific wave length and thus, using a laser scanner, the intensity of fluorescence is measured for both dyes.

The construction of a microarray, sample preparation and scanning of the slides is illustrated schematically in Figure 1.

The resulting images can be overlaid and show whether a gene is over- or underexpressed relative to the control sample. Further, from these images an intensity value for each spot and both color channels, denoted by R_g and G_g , $g = 1, \dots, n$, n the number of genes, can be extracted. Their log-ratio $M_g := \log_2(R_g/G_g)$ is related directly to the *fold change*, a common measure of differential expression. In case $R_g \geq G_g$ the fold change is simply R_g/G_g and otherwise the fold change is defined as $-G_g/R_g$. In that sense a fold change of 2 means that the corresponding gene is overexpressed by a factor 2 and a fold change of -2 means it is underexpressed by a factor 2.

In order to identify differentially expressed genes it is necessary to get rid off the systematic error which is present in microarray data. A well-known source of error is e. g. the different labelling efficiency of Cy3 and Cy5.

¹Complementary DNA (abbreviated cDNA) denotes single-stranded DNA molecules that are complementary to their mRNA templates. cDNA is assembled by the enzyme reverse transcriptase.

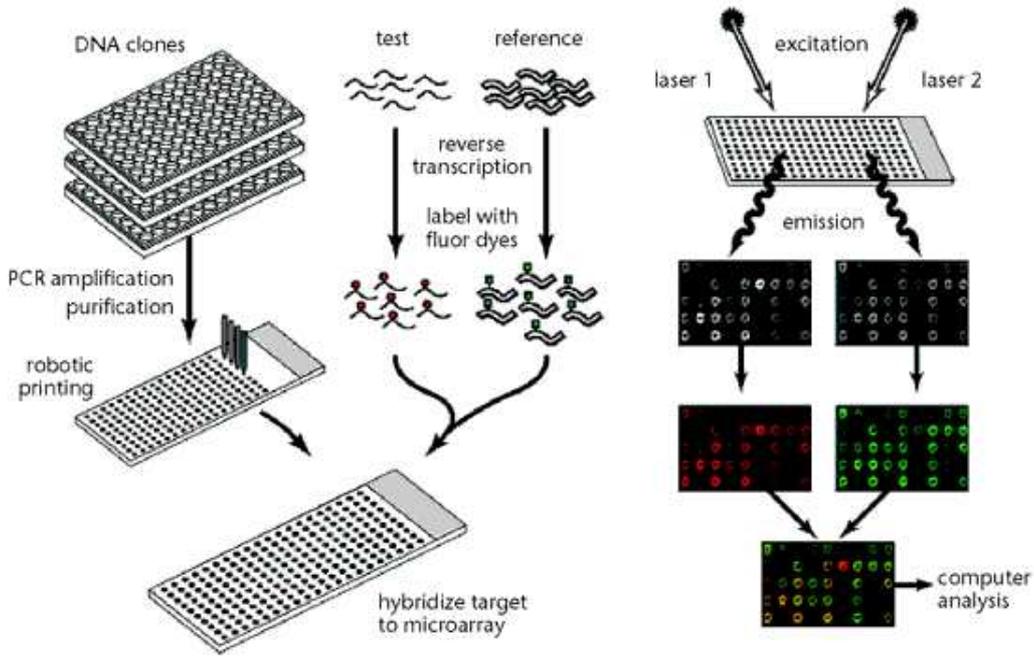


Figure 1: Schematic overview of a microarray experiment. Figure taken from [1].

One method to minimize the systematic variation in the data is by applying a global intensity-dependent normalization [27, 28] using local regression (e.g. performed by `loess` [18, 6]):

$$M_g^{norm} = M_g - c(A_g), \quad A_g := \log_2 \sqrt{R_g G_g}.$$

This regression method performs a robust locally linear fit and is not affected by a small number of outliers due to differentially expressed genes. Hence, it is applicable when the majority of the genes is assumed to be not differentially expressed.

3 Background on cluster analysis

Cluster analysis is a data mining technique concerned with grouping a set of objects having certain attributes into subsets called *clusters*. The objective is to arrange the groups such that all objects within the same cluster are, based on their attribute values, similar to each other and objects assigned to different clusters are less similar. So clustering is about revealing a hidden structure in a given set of data, usually without any external knowledge about the objects.

Not always well-separated groups are present in a data set. In this case clustering is sometimes referred to as segmentation [10]. For such problems it is considerably more difficult to assess the results of cluster analysis and decide which is the correct number of clusters.

The following notation is used throughout the paper: Let $S := \{s_1, \dots, s_n\}$ be a set of objects s_i . For each s_i there are p attributes A_1, \dots, A_p observable. Here we can assume all attributes to be real-valued. The object-attribute matrix $M := (m_{ij})$ contains the values m_{ij} of attribute A_j for object s_i , where $i = 1, \dots, n$, $j = 1, \dots, p$. A clustering C of size k of the set S is then a partition of S into pairwise disjoint non-empty clusters C_1, \dots, C_k .

3.1 Dissimilarity measures

There are many different dissimilarities for cluster analysis listed in the literature (see for example [13]). The choice of a dissimilarity measure is highly dependent on the data that is to be analysed. In the context of gene expression profiles we will only consider Euclidean distance and a dissimilarity based on Pearson's correlation coefficient.

3.1.1 Euclidean distance

Probably the most widely used dissimilarity measure for numeric attributes is the Euclidean distance. The objects s_i are considered as points in the p -dimensional space and the standard Euclidean metric is used:

$$d_{ij} := d(s_i, s_j) := \sqrt{\sum_{l=1}^p (m_{il} - m_{jl})^2}$$

In case of missing values all incomplete attribute pairs are discarded in the above sum and the sum is scaled by $\frac{p}{p_c}$ where p_c is the number of complete attribute pairs.

3.1.2 Pearson Correlation

Another common dissimilarity measure, especially for gene expression data, based on the Pearson correlation, is

$$d_{ij} := d(s_i, s_j) := 1 - \left| \frac{\sum_{l=1}^p (m_{il} - \bar{m}_i)(m_{jl} - \bar{m}_j)}{\sqrt{\sum_{l=1}^p (m_{il} - \bar{m}_i)^2 \sum_{l=1}^p (m_{jl} - \bar{m}_j)^2}} \right|,$$

where $\bar{m}_i := \frac{m_{i1} + \dots + m_{ip}}{p}$ is the arithmetic mean of s_i 's attributes and \bar{m}_j for s_j respectively. Note that this measure treats positively and negatively correlated objects equally.

Attribute pairs with at least one missing value are discarded from the calculation of the correlation coefficient.

3.2 Characteristics of clusterings

Given a clustering $C = (C_1, \dots, C_k)$ of S and the underlying dissimilarity measure d , two characteristic values can be defined as in [10]:

$$W(C) := \frac{1}{2} \sum_{l=1}^k \sum_{i, j \in C_l} d(s_i, s_j),$$

the so-called (total) *within cluster point scatter* and the (total) *between cluster point scatter*

$$B(C) := \frac{1}{2} \sum_{l=1}^k \sum_{\substack{i \in C_l \\ j \notin C_l}} d(s_i, s_j).$$

$W(C)$ characterizes the internal cohesion as it measures the pairwise dissimilarities within each cluster, whereas $B(C)$ characterizes external isolation of clusters.

They both are related through $T = W(C) + B(C)$. T , which is nothing else but the sum of all pairwise dissimilarities, is called *total point scatter* and is constant for all clusterings of S given d . Because the natural aims of clustering are to produce well-isolated and internally similar clusters, this task can be seen as minimizing $W(C)$ or maximizing $B(C)$.

Therefore W and B will play a role in assessing the quality of different clusterings in Section 3.4.

3.3 Clustering methods

There is a large number of clustering algorithms known in pattern recognition and data mining. For this work we restricted ourselves to two widely used combinatorial algorithms. Combinatorial is used here in the sense of [10], i. e. each observation s_i is uniquely assigned to a cluster C_j based solely on the data without making any assumption about an underlying probabilistic model.

Combinatorial clustering algorithms are often divided into partitioning and hierarchical methods [14]. The former construct k clusters for a given parameter k whereas the latter construct a hierarchy covering all possible values for k at the same time. In each group of methods we picked one algorithm, namely partitioning around medoids (PAM) and agglomerative clustering using Ward's method.

3.3.1 Partitioning around medoids

Partitioning methods cluster the data into k groups, where k is a user-specified parameter. It is important to know that a partitioning method will find k groups in the data for any k provided as a parameter, regardless of whether there is a "natural" clustering with k clusters or not. This leads to different criteria for choosing an optimal k . Some are discussed in Section 3.4.

Partitioning around medoids (PAM), also called k-medoids clustering², is an algorithm described by Kaufman and Rousseeuw [14] and is implemented in the R package `cluster` [18]. Its objective, seen as an optimization problem, is to minimize the within cluster point scatter $W(C)$. The resulting clustering of S is usually only a local minimum of $W(C)$.

The idea of PAM is to select k representative objects, or medoids, among S and assign the remaining objects to the group identified by the nearest medoid. Initially, in the medoids can be chosen arbitrarily, although the R implementation of PAM distributes them in a way that S is well covered. Then, all objects $s \in S$ are assigned to the nearest medoid. In an iterative loop as a first step a new medoid is determined for each cluster by finding the object with minimum total dissimilarity to all other cluster elements. Next, all $s \in S$ are reassigned to their clusters according to the new set of medoids. This loop repeats until no more changes of the clustering appear.

Because the R implementation of PAM assigns the initial clustering deterministically the results of PAM will always be identical and repeated runs to cope with random effects are not necessary.

3.3.2 Agglomerative clustering

Agglomerative methods are very popular in microarray data analysis. For example the first genome-wide microarray clustering study [8] used agglomerative hierarchical clustering.

Hierarchical clustering methods do not partition the set S into a fixed number k of clusters but construct a tree-like hierarchy that encodes implicitly all possible values of k . At each level $j \in \{1, \dots, n\}$ there are j clusters encoded. The lowest level consists of the n singleton clusters and at level one there is just one cluster containing all objects. However, hierarchical clustering imposes a nested tree-like cluster structure on the data regardless of whether the data really have this property. Therefore one has to be careful when drawing conclusions from hierarchical clustering.

²There is a close relationship to the popular k-means clustering algorithm. The advantage of k-medoids is that it can be used with any dissimilarity and not only with Euclidean distance.

In agglomerative clustering, at each level j the two “closest” clusters are merged to form level $j - 1$ with one less cluster. Agglomerative methods vary only in terms of the dissimilarity measure between clusters. In any case the dissimilarity $d : S \times S \rightarrow \mathbb{R}_{\geq 0}$ must be extended to $D : \mathcal{P}(S) \times \mathcal{P}(S) \rightarrow \mathbb{R}_{\geq 0}$, such that $D(\{s\}, \{t\}) = d(s, t)$ for $s, t \in S$. Note that the dissimilarity matrix for D would be of exponential size, but only few values are actually needed. Therefore, in practice the values of D will be computed on-demand. The general agglomerative hierarchical clustering algorithm proceeds as follows.

Initially, the set A is the trivial partition of S into singleton sets. Then in an iteration from $j = n$ down to 1 the current partition A is assigned to the clustering $\mathcal{C}^{(j)}$. Further, the two clusters with the smallest dissimilarity are determined, merged and replaced in A by their union.

There are several dissimilarity measure between clusters, e. g.

- single linkage, leading mainly to large, elongated clusters,
- complete linkage, yielding rather compact clusters,
- average linkage, being a compromise between these two extremes, or
- the dissimilarity measure used by Ward’s method [25].

The latter method uses the increment in the within cluster point scatter, which would result from merging two clusters, as the dissimilarity between them. When defining the within cluster point scatter

$$W(G) := \frac{1}{2} \sum_{i, j \in G} d(s_i, s_j)$$

for a single cluster G analogously to the total within cluster point scatter (see Section 3.2), the dissimilarity is given as

$$D(G, H) = \frac{2}{n_G + n_H} W(G \cup H) - \frac{2}{n_G} W(G) - \frac{2}{n_H} W(H).$$

Since the within cluster point scatter is a measure for the internal cohesion of clusters, Ward’s method tends to create compact clusters with very similar objects. In the case of squared Euclidean distance this method is also known as incremental sum of squares. Merging clusters that minimize D is equivalent to minimizing the within cluster variance.

A problem with Ward’s method is that it minimizes the objective function locally so that decisions taken at lower levels of the hierarchy do not necessarily mean optimality at higher levels. The other agglomerative methods suffer from this fact as well. After two clusters have been merged on a certain level there is no way of reversing this decision at a later step of the algorithm although it might be favorable. Especially when rather few clusters are sought this might be disadvantageous because many previous merging decisions are influencing the shape of higher level clusters.

When comparing the results of different agglomerative methods using the function `hclust` from the R package `mva` for our gene expression data, Ward’s method is performing best. Several comparative studies, mentioned in [12], also suggest that Ward outperforms other hierarchical clustering methods. Thus for our experiments in Section 4 we chose to compare the PAM algorithm and agglomerative clustering using Ward’s method.

3.4 Cluster validation

In most applications of clustering techniques it is impossible to speak of *the* correct clustering and therefore it is necessary to use some validation criteria to assess the quality of

the results of cluster analysis. These criteria may then be used to compare the adequacy of certain algorithms and dissimilarity measures or to choose the best number k of clusters. This is especially important when the correct number of clusters is unknown a-priori as it is the case in this study. When using PAM, the algorithm is run with different values for the parameter k and when using agglomerative clustering, this refers to finding the optimal level in the hierarchy.

Following [12], validation measures are grouped into *internal*, *relative* and *external criteria*. Internal criteria assess the quality of a given clustering based solely on the data themselves or on the dissimilarity used. Four internal criteria are introduced in Section 3.4.1.

Relative criteria are used to directly compare the agreement between two clusterings, for example to examine how similar two k -clusterings resulting from different algorithms or dissimilarities are. Section 3.4.2 describes two relative criteria.

External criteria are measuring the quality of a clustering by bringing in some kind of external information such as a-priori class labels when available. Here we denote criteria based on visualization of the cluster data as external criteria too. Visualization of clusters can help the user to assess the adequacy of a given clustering. Despite this is less objective than an internal criterion, visualization is often preferred in the final evaluation of the clustering results. Further, gene clusters based on expression profiles can be evaluated by looking at the functional annotations for their constituent genes. If certain properties are shared by many genes this might be a sign for a “good” cluster. Because external criteria are more domain specific they are described in Section 4 when the data and experiments are covered.

3.4.1 Internal validation

Silhouettes Rousseeuw [20] proposed the *silhouette* statistic which assigns to each object a value describing how well it fits into its cluster. Let $a(s_i)$ denote the average dissimilarity of s_i to all points in its own cluster and let $b(s_i)$ denote the minimum of all average dissimilarities to the other clusters, i. e. the average dissimilarity to the second best cluster for s_i . Then

$$sil(s_i) = \frac{b(s_i) - a(s_i)}{\max(a(s_i), b(s_i))}$$

is the silhouette value for s_i . Object s_i matches its cluster well if $sil(s_i)$ is close to one and poorly matches it if $sil(s_i)$ is close to zero or even negative. Negative values only occur when an object is not assigned to the best fitting cluster.

A natural measure for the quality of the whole clustering is

$$sil(C) = \frac{1}{n} \sum_{s_i \in S} sil(s_i), \quad (1)$$

the average silhouette for all objects in S . According to this criterion choose the number of clusters \hat{k} as the value maximizing the average silhouette.

Measure of Calinski and Harabasz In [17] the authors compare 28 validation criteria and found that in their experiments the measure by Calinski and Harabasz [5] performed best. It assesses the quality of a clustering with k clusters via the index

$$CH(k) = \frac{BSS(k)/(k-1)}{WSS(k)/(n-k)} \quad (2)$$

where $WSS(k)$ and $BSS(k)$ are the within and between cluster sums of squares defined analogously to W and B in Section 3.2 but using squared dissimilarities. The optimal value \hat{k} for the number of clusters is again the value k maximizing the criterion.

The idea is to choose clusterings with well isolated and coherent clusters but at the same time keeping the number of clusters as small as possible.

Originally this index was meant for squared Euclidean distance. Because our optimization criterion is not based on squared dissimilarities we used W and B instead of WSS and BSS in the definition of CH . This still follows the same motivation as using squared dissimilarities and is more robust against outliers.

Measure of Krzanowski and Lai Krzanowski and Lai [15] defined an index based on the decrease of the within cluster sums of squares. First, they defined

$$DIFF(k) = (k-1)^{2/p} WSS(k-1) - k^{2/p} WSS(k)$$

and then the index

$$KL(k) = \left| \frac{DIFF(k)}{DIFF(k+1)} \right| \quad (3)$$

which should be maximized again.

Let g be the correct number of groups in the data. Then the idea of KL is based on the assumption that $WSS(k)$ decreases rapidly for $k \leq g$ and it decreases only slightly for $k > g$. This is justified by the hypothesis that for $k \leq g$ at every successive step large clusters that do not belong together are separated resulting in a strong decrease of WSS . Conversely for $k > g$ good clusters are split resulting in a very small decrease of the within cluster sum of squares. Thus one can expect that $DIFF(k)$ is small for all k but $k = g$ and consequently $KL(k)$ is largest for the optimal k .

The same intuition holds when replacing WSS by W as in the case of Calinski and Harabasz' measure. Thus we used W in the definition of $DIFF$ for our experiments because again the measure was originally designed for use with squared Euclidean distance.

Prediction strength A more recent approach to assessing the number of clusters is the measure of *prediction strength* proposed by Tibshirani et al. [24]. It uses cross-validation of the clustering process and determines how well the clusters formed in the training set agree with the clusters in the test set. More precisely, the set S is divided into a test set S_{te} and a training set S_{tr} . Then both sets are clustered individually into k clusters yielding two clusterings C_{te} and C_{tr} . For each cluster a suitable representative element is chosen, e.g. the cluster medoid. Finally the elements in S_{te} are assigned to the training set clusters by minimizing the dissimilarity to the representative elements of the clusters in C_{tr} .

Then for each pair of elements belonging to the same cluster in C_{te} it is checked whether they fall into the same cluster again when using the training clusters. If this is the case for most pairs the prediction strength should be high otherwise it should be lower. So formally the prediction strength is defined as

$$ps(k) = \frac{1}{k} \sum_{j=1}^k \frac{1}{n_{C_j}(n_{C_j} - 1)} \sum_{\substack{i, i' \in C_j \\ i \neq i'}} D[C_{tr}, S_{te}]_{ii'}, \quad (4)$$

where $D[C_{tr}, S_{te}]$ is a matrix with the (i, i') th entry equal to one if the elements s_i and $s_{i'}$ fall into the same cluster when assigned to the training clusters as described above and zero otherwise. Thus, $ps(k)$ is the average proportion of test pairs correctly classified by the training clusters.

[24] shows that 2-fold cross-validation has no disadvantages compared to higher order cross-validation. Therefore we used 2-fold cross-validation to measure the quality of a clustering given the parameter k .

3.4.2 Relative validation

Rand index The Rand index [19] for comparing two clusterings \mathcal{C} and \mathcal{C}' of the same data S is based on four counts of the pairs (s_i, s_j) of objects in S .

- N_{11} number of pairs that are in the same cluster both in \mathcal{C} and in \mathcal{C}'
- N_{00} number of pairs that are in different clusters both in \mathcal{C} and in \mathcal{C}'
- N_{10} number of pairs that are in the same cluster in \mathcal{C} but not in \mathcal{C}'
- N_{01} number of pairs that are in the same cluster in \mathcal{C}' but not in \mathcal{C}

The Rand index is defined as the relative proportion of identically classified pairs

$$R(\mathcal{C}, \mathcal{C}') = \frac{N_{11} + N_{00}}{N_{11} + N_{00} + N_{10} + N_{01}}$$

and lies between zero and one.

It has the disadvantage that its expected value is not equal to zero for two random partitions. Therefore [11] introduced the adjusted Rand index

$$R'(\mathcal{C}, \mathcal{C}') = \frac{R(\mathcal{C}, \mathcal{C}') - E(R(\mathcal{C}, \mathcal{C}'))}{1 - E(R(\mathcal{C}, \mathcal{C}'))} \quad (5)$$

as a normalized form of Rand's criterion. It is explicitly given in [12]. The maximum value is still one and reached only if the two clusterings are identical.

Variation of information A recent work by [16] introduces an information theoretical criterion called *variation of information*. It is defined as

$$VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C}, \mathcal{C}'), \quad (6)$$

where H is the entropy of a clustering and I is the mutual information of two clusterings. The probabilities needed for these terms are estimated by the relative cluster sizes. [16] shows that VI is a metric on the space of all clusterings and gives upper bounds for its value. $VI(\mathcal{C}, \mathcal{C}')$ can be seen as a measure for the uncertainty about the cluster in \mathcal{C} of an element s knowing its cluster in \mathcal{C}' .

4 Experiments and results

4.1 Yeast cell-cycle data

A very popular data source for comparative gene expression studies is the cell cycle data set for the yeast *Saccharomyces cerevisiae* published by Spellman et al. [21]. *S. cerevisiae* is the most studied eukaryotic model organism and has about 6300 potential genes. It has the advantage that many details about its individual genes are available helping to verify the results of a cluster analysis using this external information.

Spellman et al. conducted three different time series microarray experiments. We selected the *cdc15* series consisting of 24 measurements of the expression levels for 6283 open reading frames³ (ORFs). The yeast strain was grown and then arrested at a certain

³An open reading frame is a DNA sequence that has the potential to encode a protein or polypeptide. It does not necessarily correspond to a gene.

state of the cell cycle by incubating it at 37°C. The synchronized sample was then released from the arrest and measurements were taken every 10 or 20 minutes over a total period of 300 minutes corresponding to about three cell cycles. Part of the same original culture was grown unsynchronized and served as the control sample in the microarray experiments.

The raw data for our analyses were retrieved from the Stanford Microarray Database⁴ and consist of absolute intensity values for both color channels as well as a quality flag for each gene on the array. Then the data were transformed into log-ratios and normalized as described in Section 2.

A subset of 238 genes was preselected for further processing based on several criteria. First, all genes having three or more missing values were excluded. Next, genes showing not enough differential expression over time were discarded. We chose to set a threshold of at least two time points showing an absolute fold change⁵ of two or higher. Finally, we only considered genes that exhibit a periodic behavior and thus are likely to be cell cycle dependent. This was done using a statistical method introduced in [26] and implemented in the R package *GeneTS*.

4.2 Clustering and internal validation

The general procedure is to compute the pairwise Euclidean distances (see Section 3.1.1) and the correlation based dissimilarities (see Section 3.1.2). Note that the Euclidean distances are computed from standardized profiles⁶. Then we apply PAM clustering and agglomerative clustering using Ward's method.

When clustering gene expression data, in most cases the number of clusters k is unknown in advance. Thus determining the number of clusters is a very important task. Evaluating the internal validation measures discussed in Section 3.4.1 for a range of possible k -values we can choose a couple of good candidate clusterings. By visualizing the clusterings, the homogeneity and separation of the clusters can be assessed. Differences between the candidate clusterings are quantified using the relative validation measures of Section 3.4.2. Only the best clusterings from these candidates are kept and evaluated further with external methods, which is discussed in Section 4.3.

The subset to be clustered contained 238 genes as given in Section 4.1. Therefore we chose to compare clusterings with 2 to 40 clusters. Further increasing the number of clusters would only result in a growing number of singleton clusters which is not desirable.

We used both Euclidean distance and the correlation based dissimilarity measure. The main difference between the two is that two expression profiles that are strongly negatively correlated have a high Euclidean distance but a very low correlation based dissimilarity. Hence, the genes will almost never be in the same cluster using Euclidean distance and they are very likely to end up in the same cluster using the correlation dissimilarity. The biological justification for using the correlation based measure is that genes that regulate biological processes can either be activating or repressing. So for the initiation of such a process the activator genes must be highly expressed whereas the expression of the repressor genes must be scaled back. Nevertheless it makes sense to put both groups of genes into the same cluster because they launch the same process.

Figure 2 shows the internal validation measures applied to the four different types of clusterings. The general tendency is that prediction strength and silhouette propose rather

⁴<http://genome-www5.stanford.edu>

⁵See definition in Section 2.

⁶Standardization means that for each profile the profile's mean value is subtracted and then it is divided by its standard deviation. Thus the standardized profiles have zero mean and unity standard deviation.

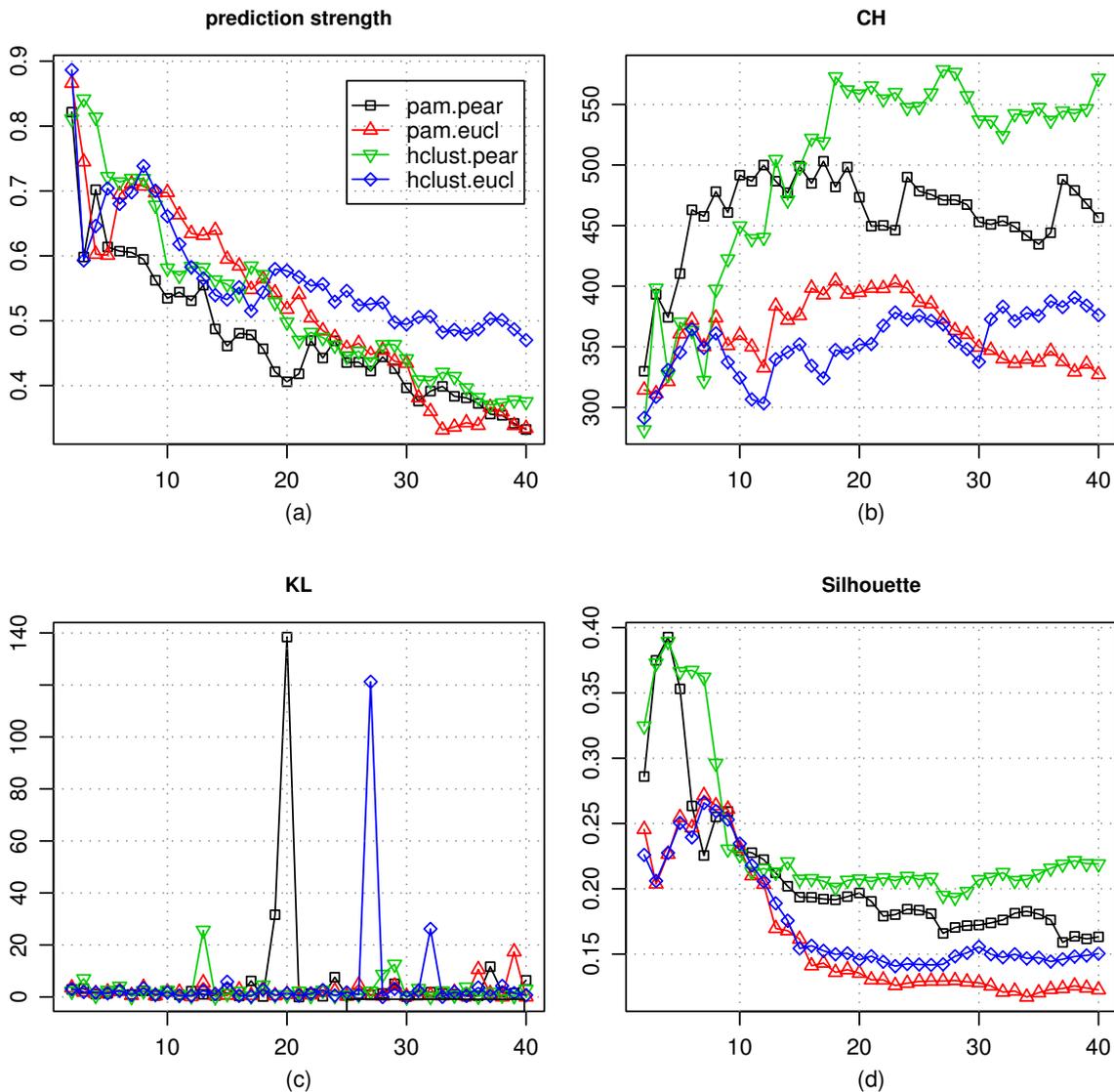


Figure 2: Plots of the internal validation measures for clusterings with $k = 2, \dots, 40$ clusters. Each plot shows the respective values for one of four validation measures applied to PAM and hierarchical clustering (hclust) using both Euclidean and Pearson correlation based dissimilarities. (a) prediction strength, (b) measure of Calinski and Harabasz, (c) measure of Krzanowski and Lai and (d) silhouette.

small cluster numbers whereas the CH criterion prefers clusterings with more clusters. The plot (c) shows strong peaks at certain positions and has a much smaller range for the rest of the values. When looking closer at how these peaks originate from eq. (3) one can see that they don't always reflect a good clustering. For example the peak at $k = 20$ with a KL value of about 140 arises from dividing $DIFF(20) = 1.636$ by $DIFF(21) = 0.012$. This is of course a *relatively* high decrease in the $DIFF$ -values. But in comparison to $DIFF$ -values in the order of 10^4 for the first $k = 2, 3, 4, 5$ considered here, these small $DIFF$ -values for larger k do not mean any true improvement in the clustering at all. This suggests that the KL index is not very useful for poorly separated data such as gene expression profiles.

For agglomerative clustering the dendrograms also give hints for good clusterings. In Figure 3 the dendrograms for both dissimilarities are shown. In plot (a) up to five clusters can be easily distinguished and about 13 clusters still have a reasonable dissimilarity when

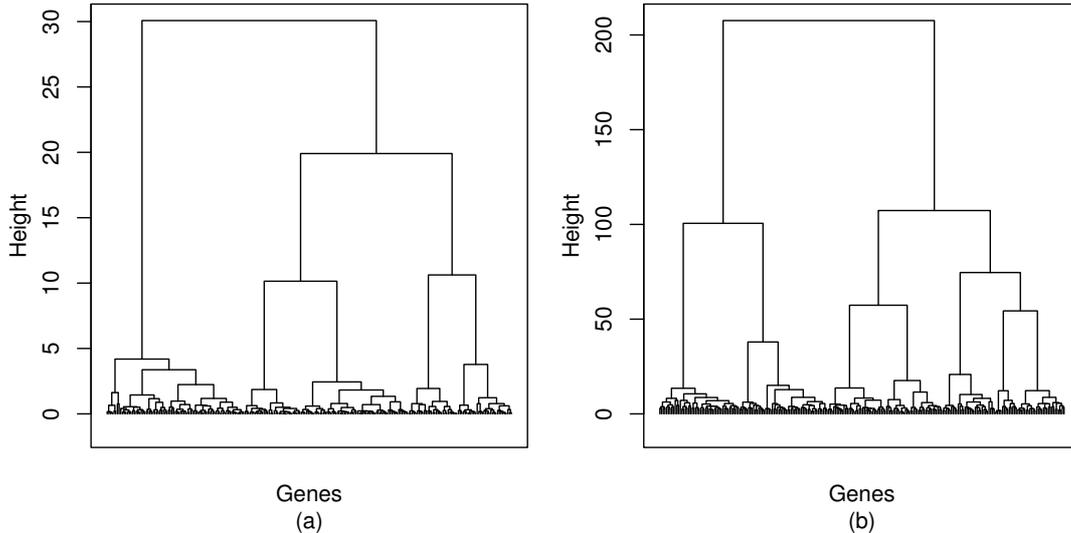


Figure 3: Dendrograms from agglomerative clustering using Ward’s method and (a) correlation based dissimilarities or (b) Euclidean distances.

they are merged, suggesting that they really represent different groups of genes. Plot (b), showing the dendrogram for Euclidean dissimilarity, indicates slightly more clusters than plot (a) when using similar arguments. This follows the motivation for the correlation based dissimilarity given at the beginning of this section: One can expect that a cluster combining positively and negatively correlated expression profiles in (a) corresponds to at least two clusters in (b).

Taking into account all four validation measures and their local behavior we identified for each combination of algorithm and dissimilarity a set of candidate values for further investigation. The selected values are given in Table 1. As an example $k = 8$ for agglomerative clustering with Euclidean dissimilarity was chosen because in Figure 2(a) the prediction strength at $k = 8$ forms a local maximum with a good value of 0.73. Further CH in part (b) of the figure has a local maximum with a high value at $k = 8$ and the silhouette in Figure 2(d) is still in a stable range before falling for $k \geq 10$. For these candidate values we displayed the clusterings as *Eisenplots* and cluster profile plots. Here only two examples are given in figures 4 and 5.

algorithm	dissimilarity	candidate values for k
PAM	correlation based	6, 8, 10 , 12, 15, 17, 20
PAM	Euclidean	6, 8, 10 , 13, 18
agglomerative	correlation based	5, 6, 10 , 13
agglomerative	Euclidean	6, 8, 15

Table 1: Candidate values for the number of clusters. The bold values are selected as “good” by visual examination.

Eisenplots (see Figure 4(a)) are named after M. B. Eisen who introduced this type of display in [8]. The expression data contained in the matrix M (see Section 2) are plotted as a table where row i and column j encodes the expression value for gene g_i at time point t_j by a color similar to the original color of its spot on the microarray. This means that high expression is coded as red and low expression as green. If the value is zero, meaning no differential expression, it is displayed in black. Note that in case of correlation based dissimilarity, profiles negatively correlated in respect to the medoid profile are multiplied by -1 because otherwise the plots would become messy. The rows are ordered such that

the elements of each cluster appear next to each other. To help identifying the cluster boundaries we included an additional column on the right hand side of the plot where alternating black and white bars mark the individual clusters.

The cluster profile plots show for each cluster the profile of the cluster medoid together with a vertical bar giving the standard deviation within the cluster at each time point. Further, in light grey all profiles of genes in the cluster are plotted and the size of the cluster is given. Note that again in case of correlation based dissimilarity, profiles negatively correlated with respect to the medoid are inverted before calculating the standard deviations. However, these profiles are plotted without change in a very light grey in the background. This enables the viewer to distinguish between positively and negatively correlated profiles.

Cluster visualization allows the viewer to assess the adequacy of the selected clusterings. One can make statements about the compactness and the isolation of the given clusters by verifying how similar the plotted profiles within each cluster and how dissimilar the medoid profiles are. If multiple clusters have very similar medoids the clustering parameter k might be chosen too large. In contrast, if clusters show a large within cluster variance it might be better to increase k in order to split those clusters.

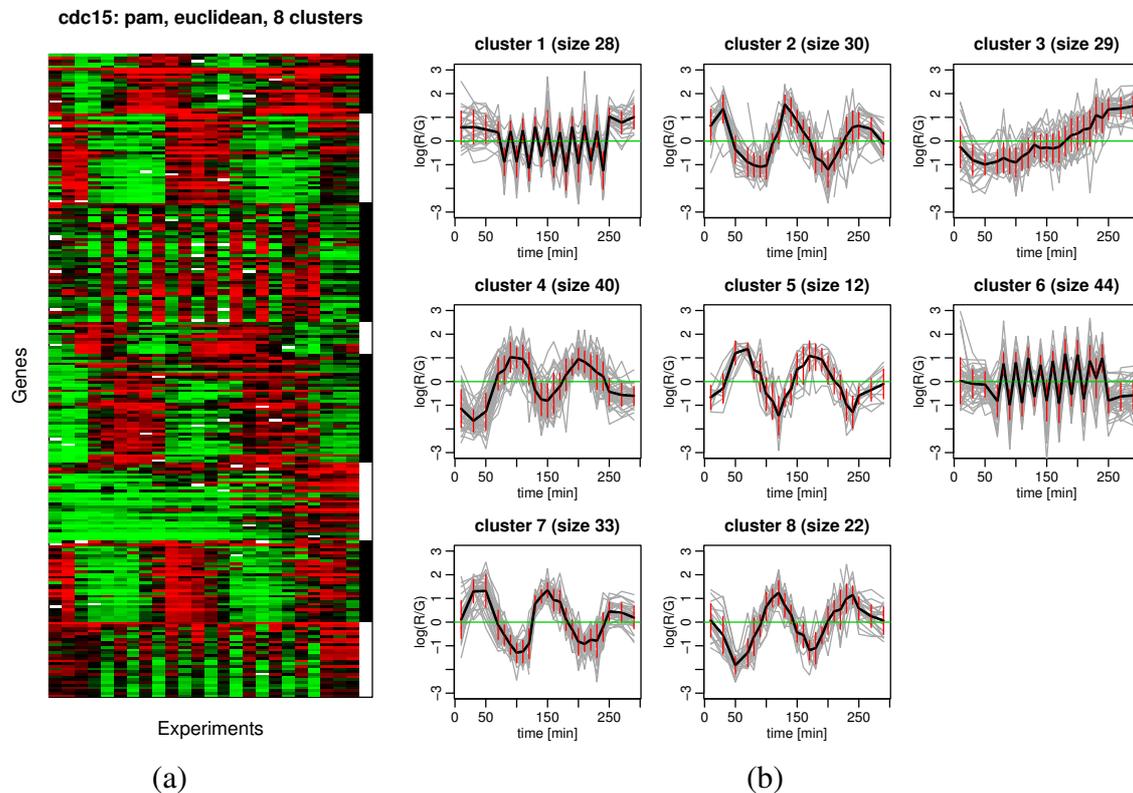


Figure 4: Visualization of PAM clustering with Euclidean dissimilarity and eight clusters. In the Eisenplot in (a) the individual clusters are marked on the right side of the plot. Cluster 1 is on the bottom and Cluster 8 on the top of the plot. In (b) the corresponding cluster profile plots are shown.

It is impossible to discuss the visualization for all candidate clusterings, so we just give two illustrative examples. In Figure 4 the result of PAM clustering with Euclidean dissimilarity and eight clusters is displayed. Three different types of clusters can be identified in the Eisenplot in (a) and the cluster profiles in (b):

Clusters 1 and 6 group together profiles that oscillate with a period of 20 minutes. Note that the first four time points and the last three time points are 20 minutes apart

cdc15: hclust, pearson, 6 clusters

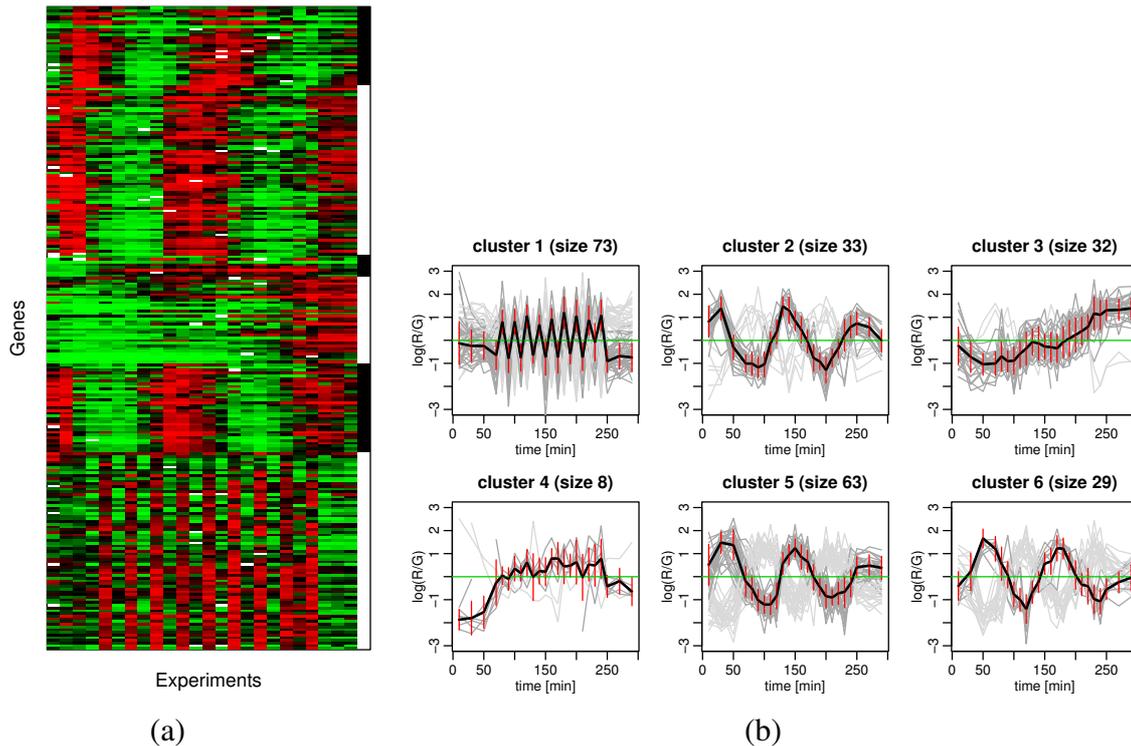


Figure 5: Visualization of agglomerative clustering with correlation based dissimilarity and 6 clusters. In the Eisenplot in (a) the individual clusters are marked on the right side of the plot. Cluster 1 is on the bottom and Cluster 6 on the top of the plot. In (b) the corresponding cluster profile plots are shown.

instead of 10 minutes for the rest of the time points. Thus the profiles look different in the beginning and in the end. The clusters are not merged because the profiles are strongly negatively correlated between both clusters resulting in large Euclidean distances.

Cluster 3 is distinct from the others by grouping profiles that have a slowly increasing behavior over the full range of measurements. Possibly these profiles have been identified as periodic with a period longer than 300 minutes.

Finally clusters 2, 4, 5, 7 and 8 show a cyclic behavior matching the approximately 2.5 cell cycles studied in the experiment. They can be subdivided into clusters 4 and 8, which consist of profiles that start off with low expression. The peaks in Cluster 4 appear about 30 minutes before those in Cluster 8 and therefore it seems reasonable not to merge them. The second group consists of clusters 2, 5 and 7, which have profiles increasing first. Here the profiles in Cluster 2 take their maxima about 20 minutes earlier than those in Cluster 7. Cluster 7 in turn is left-shifted in comparison to Cluster 5 by about 20 minutes.

The second example is given in Figure 5. To cover both algorithms and dissimilarities this figure shows agglomerative clustering with correlation based dissimilarity and 6 clusters. In contrast to the previous example, profiles that can be transferred into each other by mirroring them on the x-axis are likely to end up in the same cluster because they are strongly negatively correlated and thus have a small dissimilarity now.

As expected, Cluster 1 represents all the “zigzag” profiles that were in different clusters beforehand. Cluster 3 contains the slowly increasing profiles as before whereas Cluster 4, the smallest one with only eight elements, cannot be found in the previous example.

Clusters 2, 5 and 6 group the cell cycle dependant profiles. They can again be distinguished by the positions of the peaks over time. But here the clusters group together both

positively and negatively correlated genes, which is shown by the grey-shaded curves in the background of the plots in Figure 5(b).

The eight clusterings that we finally selected after visual examination are given in bold face in Table 1. The next step is to apply the adjusted Rand index and the variation of information criterion given in Section 3.4.2.

The goal of applying these relative validation measures is to identify the clustering that is most similar to all other clusterings. This clustering is a good candidate for further external validation discussed in the next section. Moreover the clustering being least similar to the others can be used for external validation as well. This second clustering should have discovered a structure in the data that is very different from the one discovered with the first clustering. Therefore it will be interesting to see how these two extremes perform in external validation.

clusterings			PAM				agglomerative			
			correlation		Euclidean		correlation		Euclidean	
			6	10	8	10	6	10	8	15
PAM	correlation	6	0.000	0.226	0.246	0.278	0.195	0.261	0.259	0.331
		10	0.226	0.000	0.308	0.292	0.228	0.249	0.303	0.307
	Euclidean	8	0.246	0.308	0.000	0.075	0.197	0.265	0.097	0.190
		10	0.278	0.292	0.075	0.000	0.216	0.257	0.115	0.174
aggl.	correlation	6	0.195	0.228	0.197	0.216	0.000	0.102	0.191	0.263
		10	0.261	0.249	0.265	0.257	0.102	0.000	0.263	0.280
	Euclidean	8	0.259	0.303	0.097	0.115	0.191	0.263	0.000	0.108
		15	0.331	0.307	0.190	0.174	0.263	0.280	0.108	0.000
cumulative			1.796	1.913	1.378	1.407	1.392	1.677	1.336	1.653

Table 2: Values of the variation of information criterion for all pairs of the eight selected clusterings. The smaller the value the more similar are the two clusterings with the minimum 0 reached only if two clusterings are equal. The bottom row gives the cumulative values over all columns. The smallest and largest value are given in bold face.

Both criteria, adjusted Rand index and variation of information, suggest to use the same two clusterings for the final external validation. Therefore only the variation of information is shown in Table 2. The last row contains the column sums. The smallest value, indicating the most central clustering, is found in the column of agglomerative Euclidean clustering with eight clusters. Further the largest value is the one for correlation based PAM with 10 clusters.

4.3 External validation

External validation is the final step in evaluating clusterings of gene expression data. It involves information about the genes that have not been used in the cluster analysis itself and aims at evaluating the biological relevance of the clusters. As mentioned briefly in Chapter 3.4 one possibility of assessing the biological meaning of a cluster is by looking at the functional annotations of its constituent genes. Based on the assumption that genes with similar functions or genes involved in the same biological processes are also expressed similarly, we expect meaningful clusters to group exactly these genes. In other words an optimal cluster would reflect all those and only those genes in a data set having the same function or participating in the same biological process.

One way to check this property is by scanning the Gene Ontology terms associated with the genes in a cluster. Gene Ontology (GO) [23] is a widely accepted approach for a unified vocabulary to describe molecular functions, biological processes and cellular

components of genes or gene products. The terms used for the description of genes and gene products are organized as a directed acyclic graph (DAG) and become more precise at the lower levels of the graph. It is possible that a single gene has multiple functions, takes part in different processes and appears in several components of the cell. Further, each term in the ontology can have several (less specialized) parent terms and the terms themselves follow the “true path rule”. This means that a gene product described by a child term is also described by all parent terms.

Many tools exist for accessing the Gene Ontology. For our purpose of evaluating the genes in a cluster relative to a reference set FatiGO [2] is suitable. It is a web-based application (<http://fatigo.bioinfo.cnio.es>) that extracts the GO terms for a query and a reference group of genes and further computes several statistics for the query group. We used FatiGO to access the biological process annotations for each cluster C_i in the clusterings selected in the last section. As reference set we used the union of the corresponding complementary clusters, i. e. all of the clustered genes that do not fall into this cluster C_i . The GO level to be used in the analysis has to be fixed in advance between 2 and 5. We used level 5 because most of the genes under study actually have annotations at this level⁷. As a consequence only the subset of a cluster consisting of genes that have level 5 annotations can be evaluated this way.

We used two criteria for validating clusters externally. First the cluster selectivity is assessed. This means that the proportion of genes with a certain annotation in the cluster relative to all genes in the data having this annotation is determined. A high selectivity thus indicates that the clustering algorithm is able to distinguish these genes well, based on their expression profiles, among all genes.

The second criterion is the cluster sensitivity, the proportion of genes with a certain annotation relative to all genes within the same cluster. If the sensitivity of a cluster is high then most genes in the cluster have the same annotation, in this case they participate in the same process. This is important for annotating previously unknown genes. The putative biological process for an unknown gene found in a very sensitive cluster can be given with a higher confidence compared to unknown genes in a cluster representing genes from many different processes.

For the cell cycle data we have selected the two clusterings “agglomerative with Euclidean distance and eight clusters” and “PAM with correlation dissimilarity and 10 clusters”. It is not possible to give the validation results for each cluster. Rather we give only selected results which have some interesting properties.

It must be stated that most clusters are neither very selective nor very sensitive. This may be caused on the one hand by using GO annotations from a too high level. When the level is too high, the categories are too coarse so that genes participating in subprocesses with rather different expression properties still have the same annotation from the common ancestor node in the GO tree. Of course this results in a rather low selectivity because the clustering algorithm will not group genes from these subprocesses together due to their different expression profiles. On the other hand when the level is too low, meaning that the annotations are very specific, only few genes actually have a annotation at this level and therefore only a few can have a common annotation. In this case the sensitivity of a cluster is generally low unless the cluster sizes are very small and consequently the number of clusters is undesirably large. This shows that there is a trade-off between cluster selectivity, sensitivity and the number of clusters.

Figure 6 shows the results of FatiGO⁸ for Cluster 3 of the hierarchical Euclidean

⁷Actually almost all genes not being annotated at level 5 have the annotation “molecular function unknown” at level 2.

⁸Note that the three p-values given in the figure are computed by FatiGO to assess the significance of the differences between query and reference set. The first value is the unadjusted p-value, the second and

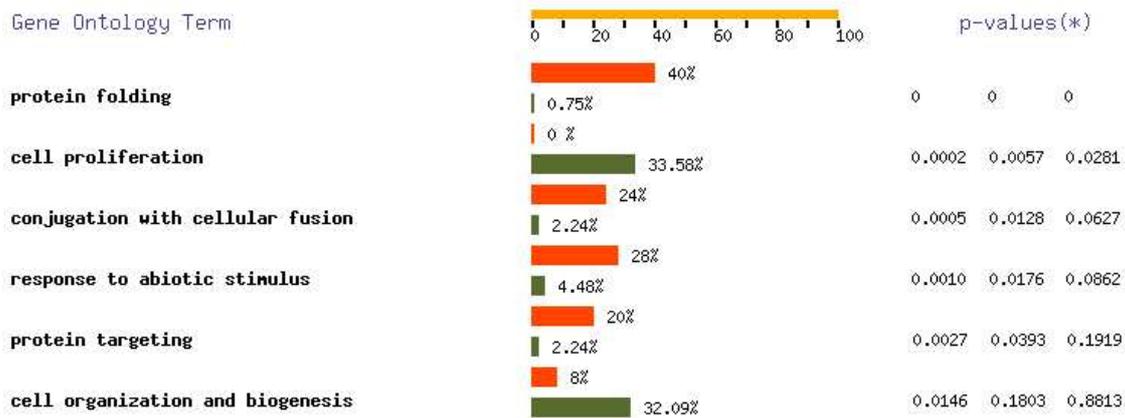


Figure 6: Part of the output of FatiGO for Cluster 3 of hierarchical Euclidean clustering with eight clusters. The six most significant differences between Cluster 3 and the reference set are given. For each GO term the upper bar gives the percentage of genes in the query cluster and the lower bar in the reference set.

clustering with eight clusters. Cluster 3 contains 25 annotated genes and the reference set has 134 annotated genes. The figure shows that this cluster is very selective for genes involved in protein folding. When looking at the absolute numbers it groups 10 out of 11 genes having this annotation. However, it is not very sensitive for protein folding, since 60 percent of the cluster is constituted by genes not using this term.

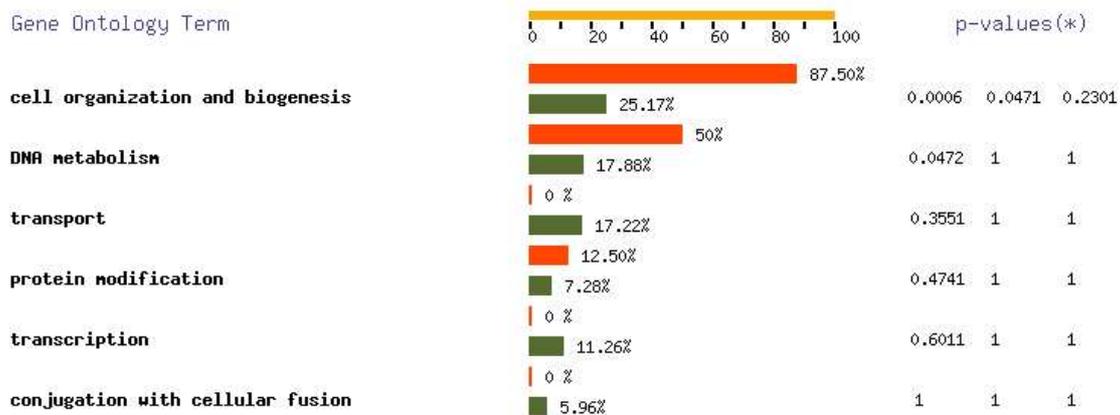


Figure 7: The six most significant differences between Cluster 5 and the reference set for the same clustering as used in Figure 6.

In Figure 7 Cluster 5, a very sensitive cluster, is shown. Seven of the eight annotated genes are labeled with cell organization and biosynthesis. However, it does not have a high selectivity for this feature because only 7 out of 45 genes involved in this process have been selected.

When considering the second clustering, PAM with correlation dissimilarity and 10 clusters, the clusters are generally less selective and sensitive, probably caused by the different dissimilarity measure and its properties. Nevertheless for example Cluster 10 shown in Figure 8 is both selective and sensitive for protein folding. As in the example of Figure 6, it contains 10 out of 11 protein folding genes. But since it is made up by

third value are computed using the *false discovery rate* adjustment procedure by Benjamini and Hochberg assuming independence and arbitrary dependence between GO terms respectively.

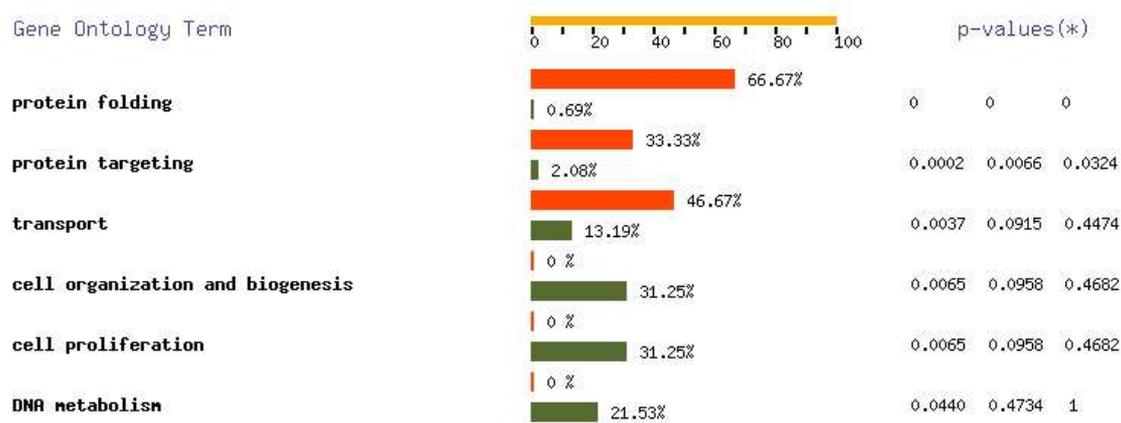


Figure 8: The six most significant differences between Cluster 10 and the reference set, now for PAM clustering with correlation based dissimilarity and 10 clusters.

only 15 annotated genes it also has a good sensitivity of 66 percent for protein folding. Another property of this cluster is that it is negatively selective for GO terms like “cell proliferation” and others. Although there are a total of 45 genes with this term none of them falls in Cluster 10.

5 Conclusions

The results presented in the previous section show once again that it is impossible to recommend a single algorithm or dissimilarity. All the clusterings evaluated – whether they came from hierarchical or PAM clustering and whether they used Euclidean or correlation based dissimilarities – have some clusters of good quality.

But still many clusters are neither very selective nor very sensitive for certain biological processes. This may have many different reasons. First, the clustering quality is highly dependent on the experimental design underlying the data. Not all experiments have the same power to extract certain groups of genes by analysing their expression profiles. Genes whose expression does not depend on the conditions tested in the experiment will most likely not show specific behaviors since they will behave asynchronously and thus not much differential expression can be expected. This means that genes with the same biological annotation do not necessarily have similar expression profiles and hence the selectivity for these genes will be low. Further, often the assumption that genes with similar functions or biological processes actually have similar expression patterns does not hold, in particular for higher and more unspecific levels in the GO tree. But this assumption is crucial for example to reveal the function of unknown genes by gene expression analysis.

In contrast, this is not a problem when the goal is to find regulatory motifs in the sequences of co-expressed genes or to discover regulatory networks. Here the relationship between co-regulation and co-expression is much closer and external validation must follow different approaches from what is described in Chapter 4. However, in this case it is more important that the dissimilarity measures take into account for instance that time-shifted profiles might still belong to the same group if the temporal distance is not too large. This is just one example for the importance of the choice of a dissimilarity measure. How similar should genes be when their profiles are scaled by a positive factor or shifted horizontally or vertically? What about scaling with negative factors resulting in negatively correlated genes? These questions have to be answered before clustering.

A second reason for clusters of low quality is that it is especially difficult to cluster

microarray data because the expression profiles tend to fill the feature space in a way that the data points are not well separated. This leads to the absence of “natural” clusters and clustering becomes segmentation. The known difficulties with the measuring precision of microarrays are partly overcome by sophisticated normalization methods. Still, the precision could be greatly improved by repeated measurements at the same conditions and a better temporal resolution of time series experiments.

The bottom line is that clustering gene expression data from microarrays is a powerful tool in bioinformatics and can reveal biologically relevant information. But it is important to compare multiple clusterings and not run one algorithm with one parameter setting and then take the results as the true structure of the data. Only the comparison of carefully chosen clusterings can result in reliable conclusions drawn from cluster analysis.

In this article we have shown how to choose and compare clusterings from different algorithms and parameter settings. After selecting meaningful dissimilarity measures a set of candidate clusterings can be chosen by evaluating several internal validation criteria. This includes specifying the number of clusters. Next, relative validation indices can help to determine the differences between clusterings and identify relatively stable clusters that appear in several clusterings. These clusters are likely to be more reliable as their structure is extracted from the data by several algorithms or dissimilarity measures. Finally, if possible suitable external biological information should be used to assess the quality of the clusterings. The Gene Ontology annotations used in this work are of course just one example of a source of external information.

However, even conclusions drawn from “good” clusters can only be seen as indications of biological meaning. The power of cluster analysis of gene expression data is that it can greatly reduce the search space and thus can lead biologists towards promising presumptions which are worth further biological examination. The verification of these presumptions by biological experiments is not replaceable.

References

- [1] Chipping Forecast. *Nature Genetics Supplement* (1999).
- [2] Al-Shahrour, F.; Diaz-Uriarte, R.; Dopazo, J.: FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics* 20 (2004) 4, pp. 578–580.
- [3] Bolshakova, N.; Azuaje, F.: Cluster validation techniques for genome expression data. *Signal Processing* 83 (2003) 4, pp. 825–833.
- [4] Bryan, J.: Problems in gene clustering based on gene expression data. *J. Multivariate Analysis* 90 (2004) 1, pp. 44–66.
- [5] Calinski, R. B.; Harabasz, J.: A dendrite method for cluster analysis. *Communications in statistics* 3 (1974), pp. 1–27.
- [6] Cleveland, W. S.; Grosse, E.; Shyu, W. M.: *Statistical Models in S*, chap. 8. Wadsworth & Brooks/Cole. 1992.
- [7] Datta, S.; Datta, S.: Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics* 19 (2003) 4, pp. 459–466.
- [8] Eisen, M. B.; Spellman, P. T.; Brown, P. O.; Botstein, D.: Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95 (1998), pp. 14863 – 14868.
- [9] Gat-Viks, I.; Sharan, R.; Shamir, R.: Scoring clustering solutions by their biological relevance. *Bioinformatics* 19 (2003) 18, pp. 2381–2389.
- [10] Hastie, T.; Tibshirani, R.; Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer. 2001.

- [11] Hubert, L.; Arabie, P.: Comparing partitions. *Journal of Classification* 2 (1985), pp. 193–218.
- [12] Jain, A. K.; Dubes, R. C.: *Algorithms for clustering data*. Prentice-Hall, Inc. 1988.
- [13] Janowitz, M. F.: *Short Course: A Combinatorial Introduction to Cluster Analysis*. Classification Society of North America. URL <http://www.pitt.edu/~csna/reports/janowitz.pdf>. 2002.
- [14] Kaufman, L.; Rousseeuw, P. J.: *Finding groups in data*. John Wiley & Sons. 1990.
- [15] Krzanowski, W. J.; Lai, Y. T.: A criterion for determining the number of groups in a data set using sum of squares clustering. *Biometrics* 44 (1985), pp. 23–44.
- [16] Meilä, M.: Comparing Clusterings. Tech. Rep. 418, University of Washington. 2002.
- [17] Milligan, G. W.; Cooper, M. C.: An examination of procedures for determining the number of clusters in a data set. *Psychometrika* 50 (1985), pp. 159–179.
- [18] R Development Core Team: *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org>. 2003.
- [19] Rand, W. M.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66 (1971), pp. 846–850.
- [20] Rousseeuw, P. J.: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.
- [21] Spellman, P. T.; Sherlock, G.; Zhang, M. Q.; Iyer, V. R.; Anders, K.; Eisen, M. B.; Brown, P. O.; Botstein, D.; Futcher, B.: Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell* 9 (1998), pp. 3273–3297.
- [22] Sturn, A.: *Cluster analysis for large scale gene expression studies*. Master’s thesis, Technische Universität Graz and The Institute for Genomic Research (TIGR) Rockville. 2001.
- [23] The Gene Ontology Consortium: Gene ontology: tool for the unification of biology. *Nature Genetics* 25 (2000), pp. 25–29.
- [24] Tibshirani, R.; Walther, G.; Botstein, D.; Brown, P.: Cluster validation by prediction strength. Tech. rep., Stanford University. 2001.
- [25] Ward, J. H.: Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc.* 58 (1963), pp. 236–244.
- [26] Wichert, S.; Fokianos, K.; Strimmer, K.: Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics* 20 (2004) 1, pp. 5–20.
- [27] Yang, Y. H.: *Statistical methods in the design and analysis of gene expression data from cDNA microarray experiments*. Ph.D. thesis, University of California, Berkeley. 2002.
- [28] Yang, Y. H.; Dudoit, S.; Luu, P.; Lin, D. M.; Peng, V.; Ngai, J.; Speed, T. P.: Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucl. Acids. Res.* 30 (2002) 4, pp. e15–.
- [29] Zhang, M. Q.: Large-Scale Gene Expression Data Analysis: A New Challenge to Computational Biologists. *Genome Research* 9 (1999) 8, pp. 681–688.

Clustermethoden für die Analyse von Siebmustern in der Papierfabrikation

K. Villforth, S. Schabel

Fachgebiet Papierfabrikation und Mechanische Verfahrenstechnik
Technische Universität Darmstadt
Alexanderstr. 8, 64283 Darmstadt
Tel. (06151) 16-2154
Fax (06151) 16-2454
E-Mail: villforth@papier.tu-darmstadt.de

1 Einleitung

Seit der Erfindung der Papiermaschine durch Nicolas-Louis Robert im Jahre 1798 werden umlaufende Siebgewebe zur Herstellung von Papier eingesetzt. Die heutigen Gewebe mit ihrem funktionellen Aufbau und den komplexen Webmustern sind ganz auf die schnelllaufenden Papiermaschinen ausgerichtet. **Abbildung 1** zeigt eine typische Siebstruktur im Querschnitt.

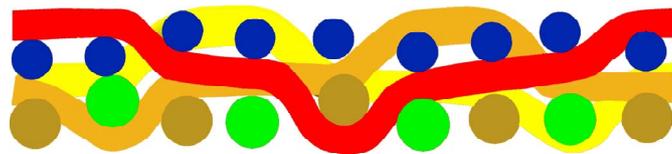


Abb. 1: Beispiel einer 2-lagigen Gewebestruktur

Bei der Entwässerung der Faserstoffsuspension entsteht auf der feiner gewobenen Sieboberseite das Papierblatt. Die Sieboberfläche prägt sich als deterministisches Muster in das stochastisch verteilte Fasergefüge ein. **Abbildung 2** veranschaulicht die Bildung des Faservlieses auf dem Sieb.

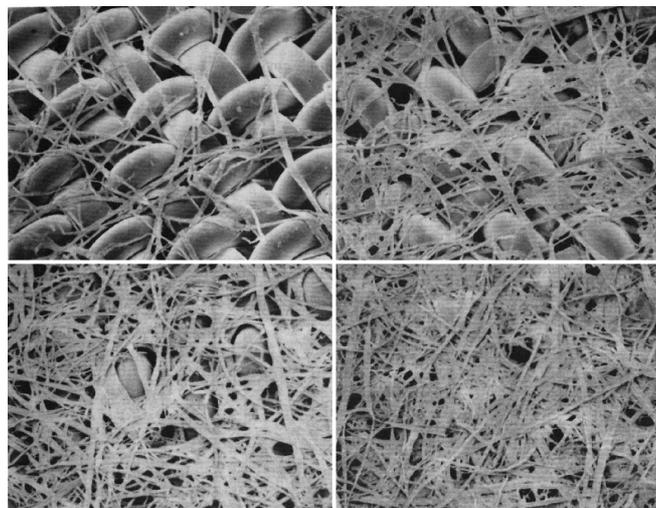


Abb. 2: Fasermatten von 0,2 g/m², 4 g/m², 7 g/m² und 10 g/m² auf einem Sieb [1]

Die lokalen Strömungsverhältnisse bei der Blattbildung beeinflussen die Verteilung von feinen Bestandteilen im Faservlies. Diese hydraulischen Effekte ergeben sich aus den unterschiedlichen Webmustern der feinen Papierseite und der größeren Tragschicht. **Abbildung 3** zeigt Siebmuster mit unterschiedlichen Drahtstärken und Strukturen, die auf eine geringe Markierungsneigung bei optimierter Wasserdurchlässigkeit und großer mechanischer Festigkeit zielen.

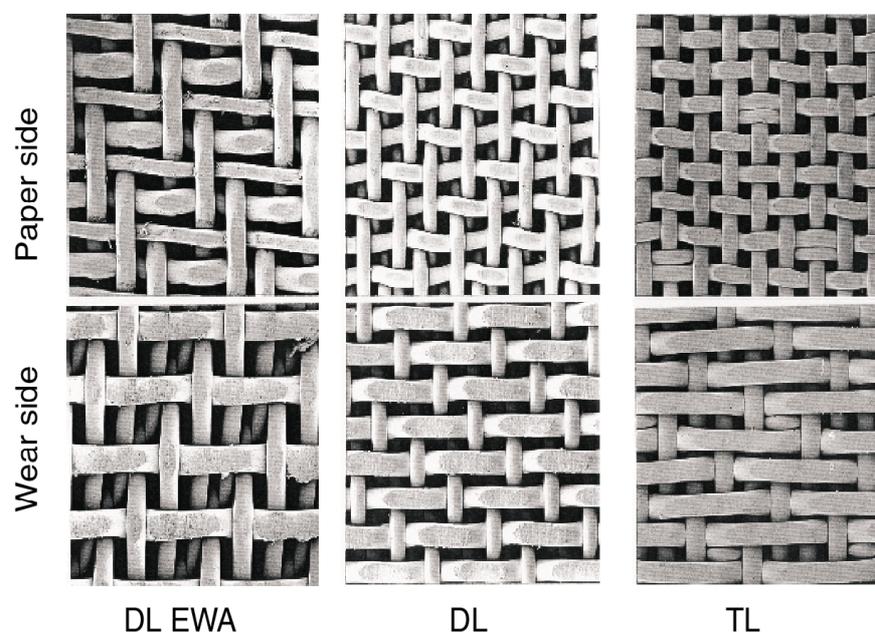


Abb. 3: Zwei- und dreilagige Siebmuster [2]

Die meisten modernen Papiermaschinen für graphische Papiere entwässern bei der Blattbildung von beiden Seiten, so dass das Faservlies zwischen zwei Sieben entsteht. Diese als Ober- und Untersieb bezeichneten Siebe können unterschiedlich aufgebaut sein. Im Papierblatt kommt es zu einer Überlagerung von deterministischen und stochastischen Signalen, die sich in der Qualität des Papiers z. B. der Bedruckbarkeit niederschlagen. In der Durchsicht lässt sich die Wolkigkeit oder auch Formation genannt, als stochastischer Anteil mit bloßem Auge erkennen.

2 Siebmarkierungsanalyse

Ziel der Siebmarkierungsanalyse ist es, deterministische Signale im Papier zu identifizieren und zu quantifizieren. Mit Hilfe der klassischen Bildanalyse lassen sich Grauwertbilder soweit aufbereiten, dass Siebmarkierungen als Amplitudenspitzen im Frequenzbereich sichtbar werden. Die Herausforderung liegt in der Identifizierung und der Klassifizierung dieser Peaks. Für diese Aufgabe eignet sich das Fuzzy-Clustering, jedoch müssen die Clustermethoden an die periodischen Strukturen angepasst werden.

2.1 Bildanalytische Vorverarbeitung

Zur Detektion von Siebmarkierungen eignen sich Aufnahmen im Durchlicht. Mit der hochauflösenden Kamera Teli CS 3910 (1.300 Pixel * 1.030 Pixel) werden die im Durchlicht beleuchteten Papierproben mit einem Bildausschnitt von 102 mm * 102 mm aufgenommen. Die Ortsauflösung beträgt 100 µm, so dass sich Gewebemarkierungen

bis 50 Drahte pro cm ohne Aliasingeffekte erfassen lassen. Typische Markierungen der papierzugewandten Siebseite liegen blicherweise zwischen 20 cm^{-1} und 30 cm^{-1} . In Einzelfallen knnen hydraulische Markierungen mehrlagiger Siebe im Bereich bis 60 cm^{-1} und hher auftreten. Dies ist bei der Identifikation von Markierungen zu bercksichtigen. Fr eine bessere Grauwertaufslung wurde in der Vorverarbeitung mit der Bildanalysesoftware Heurisko das Signal-Rauschverhaltnis durch Mittelung ber 16 Einzelbilder um 12 dB auf 70 dB erhht. Dieser Schritt ist erforderlich, da die Proben nur einen geringen Grauwertumfang aufweisen.

Die weitere Verarbeitung der Bildinformation erfolgt aufgrund des besseren Funktionsumfangs mit Matlab. **Abbildung 4** zeigt das Originalbild und dessen Filterung im Ortsbereich (von Hann). Dieser Schritt macht das Bild fr die nachfolgende Fourier-Transformation periodisch, so dass Peaks scharfer abgebildet werden.

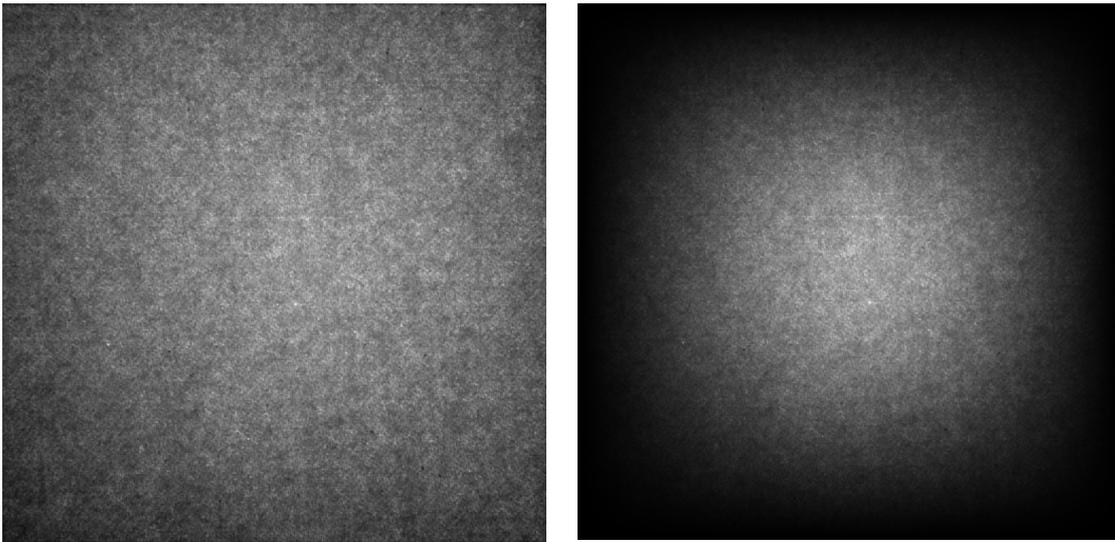


Abb. 4: Original und „von Hann“-gefiltertes Durchlichtbild im Ortsbereich

Das in **Abbildung 5** dargestellte Amplitudenspektrum muss mit einem Hochpass gefiltert werden, um die deterministischen Anteile erkennen zu knnen.

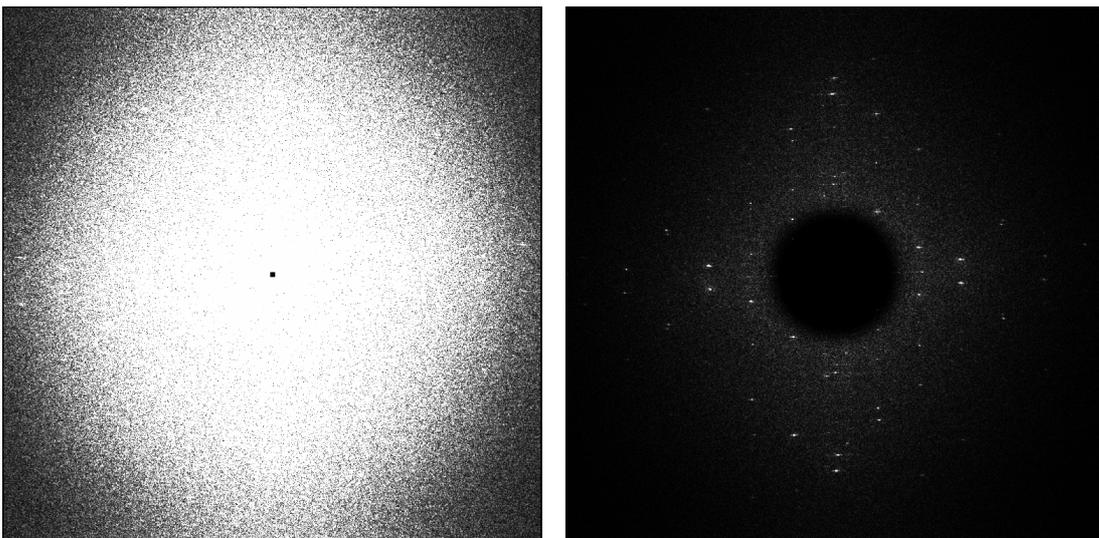


Abb. 5: Ungefiltertes und mit einem Hochpassfilter aufbereitetes Amplitudenspektrum

Der Filter ist auf gangige Markierungsmuster abgestimmt. Das Programm andert iterativ den Schwellenwert bis die zur Markierungsanalyse notwendige Anzahl an Peaks er-

reicht ist. Ihr Schwerpunkt und ihre Intensität werden aus dem unverfälschten Leistungsspektrum ermittelt, um ungefilterte Werte zu erhalten. Die Aufgabe der Bildanalyse ist mit der Ausgabe einer Liste mit den Koordinaten der Peaks und deren Anteil an der Gesamtvarianz abgeschlossen.

2.2 Siebmarkierungsgeometrie

Betrachtet man Siebmuster im Ortsbereich, so lassen unter verschiedenen Winkelperiodische Symmetrielinien erkennen, wie in **Abbildung 6** skizziert.

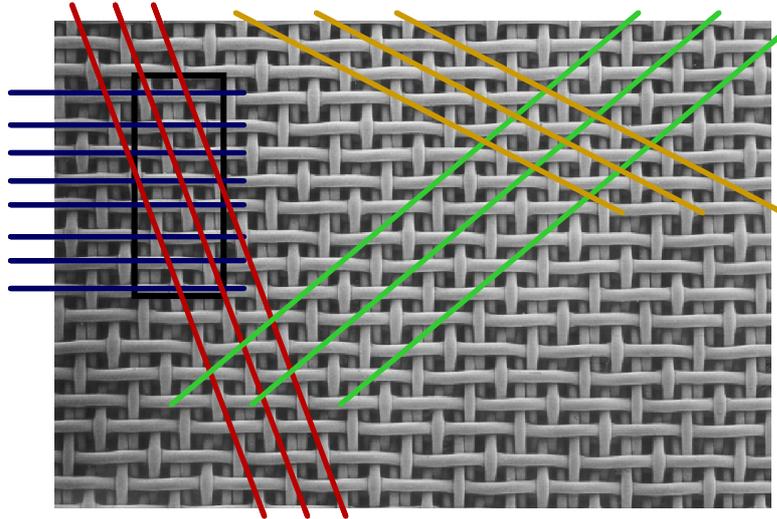


Abb. 6: Symmetrielinien auf der Papierseite einer 8-Schaft Siebstruktur

Die Winkel und die Frequenz der skizzierten Symmetrielinien korrespondieren direkt mit Peaks im Frequenzbereich. Sie bilden die Grundstruktur, wie in **Abbildung 7** dargestellt. Der Peak S repräsentiert die Ausrichtung und die Frequenz der Schussdrähte, die im Ortsbereich horizontal verlaufen. Die Peaks P1, P2 und P3 entsprechen den Symmetrielinien, die sich aus dem Bindungsrapport und der Steigungszahl ergeben. Die vertikal verlaufenden Kettfäden treten bei diesem Webmuster nicht in Erscheinung. Dennoch lässt sich die Position des Kettfadenpeaks K rechnerisch ermitteln. Da die Kettfäden in der Papiermaschine grundsätzlich in Laufrichtung (MD = machine direction) ausgerichtet sind, ist damit die Ausrichtung der Papierprobe bestimmt.

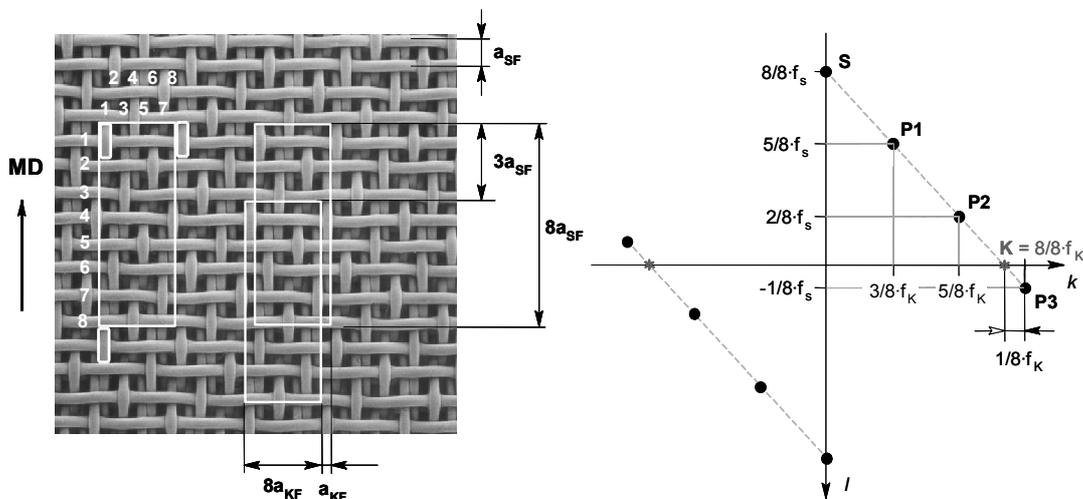


Abb. 7: Bindungsrapport und Steigungszahl im Orts- und Frequenzbereich

Zur Beschreibung einer Siebmarkierung im Frequenzbereich genügen zwei Vektoren. Die Addition ganzzahliger Vielfache dieser Vektoren führt zu den Peaks der Grundfrequenzen sowie deren Oberwellen. Die Menge W von Peaks einer Siebmarkierung kann als Abbildung von t_j aus der Menge der ganzen Zahlen \mathbf{Z}^2 mit der Matrix a_{ij} auf y_i in \mathbf{R}^2 d. h. als Transformation $\mathbf{Z}^2 \xrightarrow{a_{ij}} \mathbf{R}^2$

$$W = \left\{ y \in \mathbf{R}^2 : y_i = a_{ij} t_j : t_j \in \mathbf{Z}^2 \right\} \quad (2.1)$$

beschrieben werden. Für eine der beiden Siebmarkierungen sind in **Abbildung 8** Hilfslinien im Intervall $t = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$ bis $t = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ eingezeichnet.

2.3 Überlagerung von Siebmarkierungen

Bei der Analyse graphischer Papiere findet man in der Regel topographische Markierungen auf beiden Seiten der Papierprobe. Im Durchlicht überlagern sich die beiden Markierungen. **Abbildung 8** veranschaulicht die Überlagerung der Markierungen von Ober- und Unterseite. Die Siebmarkierung des Oberseite (blau) tritt stärker hervor als die Markierung der Unterseite (rot). Beide Markierungen zeigen die symmetrische Anordnung ihrer Peaks, wobei nur für die Markierung der Oberseite Hilfslinien eingezeichnet sind.

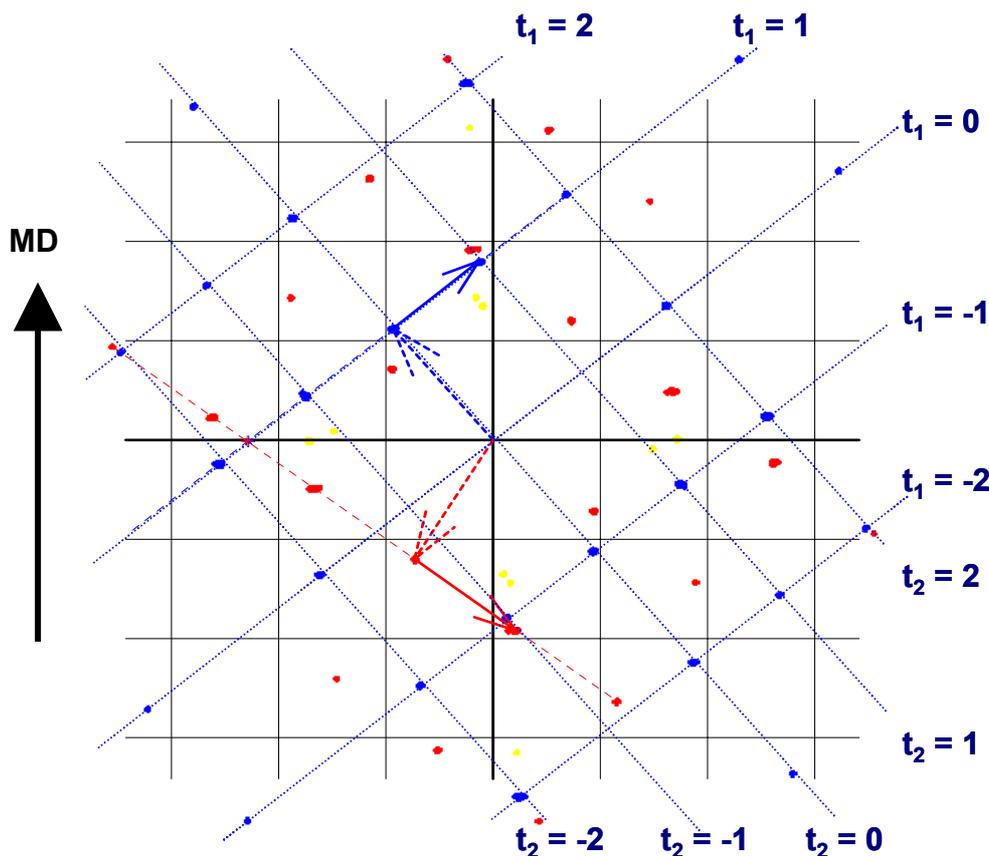


Abb. 8: Siebmarkierungen im Papier eines Doppelsiebformers mit Ober- und Untersieb

Durch die Glättung der Papieroberfläche oder durch Auftragen einer Deckschicht (Coating) können topografische Markierungen so stark gedämpft werden, dass sie sich im Amplitudenspektrum nicht mehr isolieren lassen.

3 Clustermethoden zur Siebmarkierungsanalyse

Clusteralgorithmen gehören zu den wichtigsten Methoden der Datenanalyse. Im Allgemeinen treten Strukturen und funktionale Zusammenhänge in Form von Häufungspunkten auf, die die mit Clustermethoden gefunden werden können. Das Ziel der Clustermethoden in der Siebmarkierungsanalyse ist es, die Koordinaten der Amplitudenspitzen im Frequenzbereich (Peaks) zu untersuchen und auszuwerten.

Das Problem bei der Auswertung dieser Peaks liegt darin, dass sie keine Häufungspunkte besitzen, sondern sich wie in **Abbildung 8** dargestellt, in ihrer geometrischen Anordnung durchdringen. Die den Clustermethoden zugrunde liegenden Ähnlichkeits- oder Distanzmaße müssen an das Problem angepasst werden, um sie anwenden zu können. Alternativ lässt sich der zu klassifizierende Datensatz mit Hilfe der Transformationsvorschrift **Gl. 2.1** auf ein gemeinsames t normieren, so dass zusammengehörige, aber räumlich getrennte Datenpunkte zusammengeführt werden. Damit eröffnen sich alle Möglichkeiten der Ähnlichkeitsermittlung bei metrischer Variablenstruktur [3].

3.1 Bestimmung der Distanzwerte

Betrachtet man eine einzelne Siebmarkierung, so stellen alle y der Menge W nach **Gl. 2.1** Prototypen ihrer Peaks dar. Maßgeblich für die Zugehörigkeit eines Punktes x_k des zu klassifizierenden Datensatz $\mathbf{X} = \{x_1, \dots, x_n\}$ ist die Euklidische Distanz (L_2 -Norm) zum nächstliegenden Prototyp in $W = \{y \in \mathbf{R}^2 : y_i = a_{ij} t_j : t \in \mathbf{Z}^2\}$

$$d(x_k, a_{ij}) = \|x_k - a_{ij} t_j\| \quad (3.1).$$

Im Bereich $d(x_k, a_{ij}) < \frac{1}{2} \|a_{ij}\|$ kann mit der Funktion floor, die auf ganze Zahlen abrundet, t_j bestimmt werden

$$t_j = \text{floor}(a_{kj}^{-1} x_k + 0,5) \quad (3.2).$$

Die Distanz $d(x_k, a_{ij})$ zum nächstliegenden Prototyp wächst mit t , so dass äußere Punkte trotz identischer Zugehörigkeit größere Abstände aufweisen. Es besteht zudem die Gefahr, dass der Bereich $d(x_k, a_{ij}) < \frac{1}{2} \|a_{ij}\|$ überschritten wird. Die Lösung liegt in der Berechnung des Distanzwertes über den relativen Abstand des Datenpunktes zum nächstliegenden Prototyp. Mit **Gl. 3.2** berechnet sich $d(x_k, a_{ij})$ zu

$$d(x_k, a_{ij}) = \left\| \frac{x_k}{a_{ij} t_j} - 1 \right\| \quad (3.3).$$

3.2 Fuzzy-Clustering

Eine Motivation für die Anwendung von Fuzzy-Clustermethoden liegt in dem Konzept der Zugehörigkeitsgrade. Die variablen Zugehörigkeitsgrade erlauben einen effizienten Umgang mit überlagerten oder überzähligen Datenpunkten. In der Siebmarkierungsanalyse tritt relativ häufig der Fall auf, dass sich Peaks unterschiedlicher Markierungen mehr oder weniger überlagern. Die Bildanalyse ermittelt den gemeinsamen Schwer-

punkt und integriert die Markierungsintensität über die Fläche. Die harte Klassifizierung würde diesen Punkt zu 100 % einem der Cluster zuordnen. In der Folge geht dieser Punkt mit ungenauen Koordinaten und übergroßer Intensität in die Berechnungen ein.

Im Fall des Fuzzy-Clustering wird der überlagerte Datenpunkt anteilig den Clustern zugeordnet. Während eindeutige Peaks Zugehörigkeitsgrade von mehr als 0,99 erreichen, liegen sie bei überlagerten Punkten deutlich niedriger. Dass der Zugehörigkeitsgrad bei der Berechnung der Clusterausrichtung berücksichtigt wird, reduziert den Fehler der Markierungsanalyse. Dieser Zugehörigkeitsgrad wird ebenfalls genutzt, um Markierungsintensitäten überlagelter Peaks zuzuordnen.

Wie bereits dargestellt, wird eine Siebmarkierung nach **Gl. 2.1** durch die Transformationsmatrix b_{ij} eindeutig festgelegt. Um eine Siebmarkierung zu repräsentieren sind mit dem Clusterzentrum 4 Freiheitsgrade verbunden, die einzeln oder in Gruppen optimiert werden müssen.

Der Vergleich unterschiedlicher Fuzzy-Clustermethoden bringt Aufschluss hinsichtlich ihrer Möglichkeiten und der erreichbaren Freiheitsgrade [4]. Die Clusterzentren des einfachsten und zugleich am weitesten verbreiteten Verfahrens, der Fuzzy c-Means-Methode **FCM** [5], besitzen im zweidimensionalen Raum 2 Freiheitsgrade, so dass 2 Zentren nötig wären, um eine Siebmarkierung abzubilden. Mit den im Folgenden kurz vorgestellten Verallgemeinerungen und Erweiterungen des FCM-Verfahrens lassen sich höhere Freiheitsgrade realisieren.

Ein Fuzzy-Clusterverfahren, das einen Datensatz nicht in kugelförmige Cluster aufteilt, ist der Fuzzy-C-Varieties-Algorithmus **FCV**. Er eignet sich zur Erkennung von Linien, Ebenen und Hyperebenen. Jeder Cluster wird als r -dimensionale lineare Mannigfaltigkeit dargestellt [4]. $r \in \{0, \dots, p-1\}$, p ist die Dimension des Vektorraums, hier R^2 . Der Prototyp eines Clusters $b_i = (z_i, \{e_{i1}, e_{i2}, \dots, e_{ir}\})$ besteht aus einem Punkt z_i sowie r linear unabhängige Richtungsvektoren der Länge 1, die einen unendlich ausgedehnten linearen Unterraum aufspannen. Das Distanzmaß ergibt sich aus dem Abstand zu diesem Unterraum. Für den Datensatz $\mathbf{X} = \{x_1, \dots, x_n\}$ und den Vektor $\mathbf{B} = \{b_1, \dots, b_c\}$ von Prototypen ist der Abstand $d^2(x_k, b_i)$ definiert als:

$$d(x_k, b_i) = \|x_k - z_i\|^2 - \sum_{j=1}^r \left((x_k - z_i)^T e_{ij} \right)^2 \quad (3.4).$$

Für $r = 1$ besteht der Unterraum aus einer Geraden. Man spricht von der Fuzzy c-Lines Methode **FCL**. Die FCL-Methode besitzt $r+2$ Freiheitsgrade. Zwei entfallen auf die Position des Clusterzentrums und eine auf den Richtungsvektor. Die Steigung der Geraden kann mit der Principal Component Analyse **PCA** oder der Minor Component Analyse **MCA** aus den Eigenwerten der Fuzzy-Kovarianzmatrix berechnet werden [6].

Kombiniert man die FCM-Methode mit der FCV-Methode, so entsteht das Fuzzy c-Elliptotypes Clustering **FCE**. Es erweitert das FCV-Modell um einen Gewichtungsfaktor α , der in die Berechnung des Abstands eingeht:

$$d(x_k, b_i) = \|x_k - z_i\|^2 - \alpha \sum_{j=1}^r \left((x_k - z_i)^T e_{ij} \right)^2 \quad (3.5).$$

Für kleine α dominiert der kugelförmige Einfluss des FCM-Anteils. Da α eine Konstante ist, wird die Zahl der Freiheitsgrade nicht erhöht. Die Berechnungsgrundlagen unterscheiden sich nicht vom FCV-Algorithmus [7].

Um die notwendige Zahl der Freiheitsgrade zu erreichen, bietet sich eine modifizierte FCL- Methode an. Wählt man die Dimension des Vektorraums zu $r = 2$, so besteht der Prototyp eines Clusters $b_i = (z_i, \{e_{i1}, e_{i2}\})$ aus einem Punkt z_i sowie 2 linear unabhängigen Richtungsvektoren der Länge 1. Führt man die Längenfaktoren l_i ein, so lassen sich die Clusterzentren z_i als Vektorsumme mit insgesamt 4 Freiheitsgraden darstellen:

$$z_i = l_{i1}e_{i1} + l_{i2}e_{i2} \quad (3.6).$$

Aus **Gl. 3.6** ergibt sich die zu $b_i = (z_i, \{e_{i1}, e_{i2}\})$ äquivalente Darstellung des Prototyps eines Clusters als Matrix

$$b_i = l_{i1}e_{i1} + l_{i2}e_{i2} = a_{ijk} \quad (3.7),$$

so dass der Abstand $d(x_k, a_{ij})$ nach **Gl. 3.3** berechnet werden kann. Die Richtungsvektoren werden in zwei Schritten aus den Eigenwerten von Fuzzy-Kovarianzmatrizen ermittelt, wobei der Datensatz $\mathbf{X} = \{x_1, \dots, x_n\}$ zunächst auf die Ursprungsgerade $t_1 = 0$ abgebildet wird. Durch die Punktsymmetrie des Amplitudenspektrums gruppieren sich die Punkte eines Clusters entlang dieser Ursprungsgeraden, so dass deren Ausrichtung für einen Richtungsvektor bestimmt werden kann. Im zweiten Schritt wird der Datensatz auf die Ursprungsgerade $t_2 = 0$ projiziert und der Eigenvektor berechnet. Die Abbildung des Datensatzes auf den Punkt $t = \{1, 1\}$ ermöglicht es, die Clustercenter z_i , die Zielfunktion und die Zugehörigkeiten analog der FCL-Methode zu ermitteln.

3.3 Initialisierung der Clusterprototypen

Jede Siebmarkierung lässt sich nach **Gl. 2.1** durch eine Transformationsmatrix $\mathbf{b} = (b_{ij})$ darstellen. Es die Aufgabe der Initialisierung, alle möglichen \mathbf{b} aus dem Datensatz $\mathbf{X} = \{x_1, \dots, x_n\}$ zu extrahieren, zu bewerten und auf Redundanz zu prüfen. Da Siebmarkierungen sehr dicht nebeneinander liegen können, sollen die Clusterprototypen mit einer optimalen Ausrichtung initialisiert werden. Durch die Auswahlkriterien werden die einzelnen Siebmarkierungen erkannt und Artefakte im Datensatz ausgeschlossen.

Die systematischen Suche erzeugt alle möglichen Matrizen $\mathbf{a}_k = (a_{kij}) = (x_k - x_i, x_k - x_j)^{-1}$ die Vektoren $\mathbf{F} = x_k - x_i$ und $\mathbf{G} = x_k - x_j$ für $k = 1 \dots n-2$, $i = k \dots n-1$ und $j = i \dots n$. Deren Beträge $|\mathbf{F}|$ und $|\mathbf{G}|$ müssen sich im Fangbereich befinden. Der Winkelbereich wird mit

der Bedingung $\left| \frac{\mathbf{F} \cdot \mathbf{G}}{|\mathbf{F}| |\mathbf{G}|} \right| \leq 0,5$ auf $90^\circ \pm 30^\circ$ eingeschränkt. Als Auswahlkriterien bei re-

dundanten Matrizen werden zunächst ihre L_1 -Norm (City-Block-Metrik), dann die Intensität der von \mathbf{a}_k tangierten Peaks herangezogen. Das Ergebnis ist ein Vektor von Transformations-Matrizen $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_c\}$ als Clusterprototypen.

4 Zusammenfassung

Mit Hilfe von Fuzzy-Clustermethoden ist eine weitgehend automatische Analyse von Querprofilen möglich. Die optimierte Markierungsanalyse bietet folgende Vorteile:

- Die Methode ist rotationsinvariant. Die Proben müssen nicht ausgerichtet werden, welches den Aufwand deutlich reduziert und Winkelfehler vermeidet.
- Mit der automatischen Siebmarkierungsanalyse entfällt das zeitraubende und fehlerträchtige manuelle Selektieren von Peaks am Bildschirm.

- Verlässliche und reproduzierbare Ergebnisse ergeben sich aus der Doppelbestimmung über die Markierungen von Ober- und Untersieb. Die lokale Dehnung im Papier lässt sich aus zwei unterschiedlichen Markierungen unabhängig voneinander bestimmen und zur Fehleranalyse vergleichen. So zeigt z. B. das Querschrumpfungprofil eines typischen Zeitungsdruckpapiers eine mittlere absolute Abweichung von 0,23 % und das Längsdehnungsprofil von 0,13 %.
- Optimierte, subpixelgenaue Auflösung. Die Genauigkeit wird maßgeblich durch externe Einflüsse begrenzt. Dazu zählen Schwankungen beim Siebherstellungsprozess wie z. B. die Gleichlaufschwankungen von Webstühlen und die Siebschrumpfung bei der Thermofixierung.

Mit Hilfe der neuen Auswertalgorithmen ist es möglich, bei der Auswertung von Querprofilen die Einflüsse aus Siebverzug und Querschrumpfung iterativ zu kompensieren und so den Anteil der reinen Schrumpfung an der Verformung der Siebmarkierung zu extrahieren.

5 Danksagung

Das Forschungsvorhaben (AiF 13365N) wird aus Haushaltsmitteln des Bundesministeriums für Wirtschaft und Technologie (BMWi) über die Arbeitsgemeinschaft industrieller Forschungsvereinigungen "Otto von Guericke" e.V. (AiF) gefördert.

6 Literatur

- [1] Herzig, R.; Johnson, D.: *Investigation of thin fibre mats formed at high velocity*. TAPPI 1997 Engineering and Papermakers Conference Proceedings, TAPPI PRESS, Atlanta, Vol. 1, p. 109.
- [2] Gullichsen, J.; Paulapuro, H.: *Papermaking Science and Technology. Papermaking Part I, Stock Preparation and Wet End*. Fapet Oy, Helsinki; 1998
- [3] Backhaus, K.; Erichson, B.; Plinke, W.; Weiber, R.: *Multivariate Analysemethoden*. Springer-Verlag, Berlin, Heidelberg, New York; 2003
- [4] Timm, H.: *Fuzzy-Clusteranalyse: Methoden zur Exploration von Daten mit fehlenden Werten sowie klassifizierten Daten*. Dissertation, Otto-von-Guericke-Universität Magdeburg; 2002
- [5] Bezdek, J. C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York; 1981
- [6] Honda, K.; Togo, N.; Fujii, T.; Ichihashi, H.: *Linear Fuzzy Clustering Based on Least Absolute Deviations*. Proc. of 2002 IEEE Int. Conf. on Fuzzy Syst.; 2002
- [7] Hanss, M.: *Identifikation von Fuzzy-Modellen und ihre Anwendung zur Regelung nichtlinearer Prozesse*. Dissertation, Universität Stuttgart; 1998

Multi-step-ahead prediction in control systems using Gaussian Processes and TS - Fuzzy Models

Rainer Palm¹
Reisinger Str. 19
80337 Munich
email: rub.palm@t-online.de

1. Introduction

Building a model of a system is motivated by many aspects. One aspect is the use of the model as prior knowledge in a control loop [6]. Other system models are used for observers for nonmeasurable states e.g. for fault diagnostics in complex systems [3]. Another important point of view is the prediction of the behavior of a system in a complex environment in the presence of disturbances and uncertainties for which the Kalman filter is one of the most useful tools [13].

The classical way of modeling is based on physical laws usually formulated in terms of analytical state equations of the system [6]. If the physical knowledge is poor or not available a black-box model of the system may be extracted from training data. In most cases, however, a mixture of data-driven and analytical methods is superior to pure analytical or black-box techniques. Fuzzy multiple modeling is such a technique where a global system $y(n+1) = f(\mathbf{x}(n)) + w(n)$ is approximated by a number of blended local linear or nonlinear models whose structures are supposed to be defined in advance [4, 9, 11, 12]. Location and area of validity of each local model is obtained from fuzzy clustering of training data [10].

On the other hand, modeling with Gaussian processes is a probabilistic and non-parametric method which allows a prediction accounting for the uncertainty of the model. This is of advantage for one-step-ahead or even multi-step-ahead predictions with noisy time series and disturbed closed loop control systems [1, 2, 5, 8, 14]. In these publications the model is trained *offline* which is reasonable for large sets of training data. The result is therefore a *global model* of the system. A comprehensive introduction to modeling with Gaussian Processes can be found in [7].

In this paper *local online modeling* with only few data is proposed where in each time step n the one-step-ahead prediction accounts for l previous consecutive samples $\mathbf{X} = (\mathbf{x}(n-l), \dots, \mathbf{x}(n))^T$ plus $\mathbf{x}(n+1)$ and a target vector $\mathbf{t} = (t(n-l), \dots, t(n))^T$ representing the previous l inputs for the Gaussian estimator computing the estimate $\hat{y}(n+1)$. This modeling works as a filter with a constant but moving time horizon for the past.

The multi-step-ahead prediction for a system $\mathbf{x}(n+1) = \mathbf{f}(\mathbf{x}(n), \mathbf{u}(n))$ assumes the previous outputs and control values to be known as well as the future control values. A “naive” multi-step-ahead prediction is a successive one-step-ahead prediction whereas the outputs in each consecutive step are considered as inputs for the next step of prediction [1]. Usually for closed loop control systems the nominal output trajectory is known in advance. However, because of the uncertainties and disturbances in the control loop the resulting control trajectory is only known up to the present time step n but not for the

¹ Adjunct Professor at AASS,
Technical University Örebro, Sweden

future steps. To obtain the future control inputs for the multi-step-ahead prediction the system is modeled by a TS fuzzy model which was trained in advance to generate a nominal control trajectory in closed loop for a given nominal output trajectory. Simulations with nonlinear systems with built-in uncertainties illustrate the good performance of the multi-step-ahead prediction with the combination of TS fuzzy models and Gaussian process models.

The paper is organized as follows. In Sect. 2 Gaussian processes as a method for regression and modeling are presented. Section 3 deals with modeling of dynamical systems by Gaussian processes. In Sect. 4 an efficient one-step-ahead prediction with a constant but moving past horizon is discussed. Section 5 deals with the so-called “naive” multi-step-ahead prediction for open loop systems. In Sect. 6 the multi-step-ahead prediction in a closed loop control system using a global Takagi-Sugeno fuzzy model is presented. Section 7 finishes with some conclusion

2. Gaussian processes for modeling and regression

The following approach is based on the Bayesian framework [7]. Let this be made clear by a stochastic Gaussian process $y = f(x)$ which reflects an input-output mapping $x \rightarrow y$.

The goal is to infer the conditional probability

$$P(y(n+1) | \mathbf{y}_n, \mathbf{x}_n) = \frac{P(y(n+1), \mathbf{y}_n, \mathbf{x}_n)}{P(\mathbf{y}_n, \mathbf{x}_n)} = \frac{P(\mathbf{y}_n | y(n+1), \mathbf{x}_n) \cdot P(y(n+1))}{P(\mathbf{y}_n | \mathbf{x}_n)} \quad (1)$$

for an output measurement $y(n+1)$ given the previous output measurements $\mathbf{y}_n = (y(1), \dots, y(n))^T$ and input measurements $\mathbf{x}_n = (x(1), \dots, x(n))^T$. This conditional distribution is used to make a prediction about $y(n+1)$.

To be more general, a stochastic Gaussian process is a collection of random variables $\{Y(\mathbf{x}) | \mathbf{x} \in X\}$ with a multivariate Gaussian probability distribution with a mean function $\mu(\mathbf{x}) = E[Y(\mathbf{x})]$ and the covariance function

$$C(\mathbf{x}, \mathbf{x}^*) = E[(Y(\mathbf{x}) - \mu(\mathbf{x}))(Y(\mathbf{x}^*) - \mu(\mathbf{x}^*))^T]. \quad \mathbf{x} = (x_1, x_2, \dots, x_m)^T \text{ and } \mathbf{x}^* = (x_1, x_2, \dots, x_m)^{*T} \text{ are } m\text{-dimensional input vectors.}$$

With respect to modeling $Y = f(\mathbf{x})$ is an input-output mapping $\mathbf{x} \rightarrow Y$ (see Fig 1)

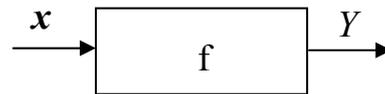


Fig 1. Input-output mapping

The model is trained by n pairs of input data $\mathbf{x}(i)$ and target (output) data $t(i)$ ($i=1 \dots n$). Then an input vector $\mathbf{X} = (\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n))^T$ and a target vector

$\mathbf{t} = (t(1), t(2), \dots, t(n))^T$ are defined where the inputs are m -dimensional vectors

$\mathbf{x}(i) = (x_1(i), x_2(i), \dots, x_m(i))^T$ and the targets $t(i)$ are scalars.

The covariance function $C(\mathbf{x}(i), \mathbf{x}(j))$ for two data samples $\mathbf{x}(i), \mathbf{x}(j)$ is chosen as

$$C_{ij} = C(\mathbf{x}(i), \mathbf{x}(j)) = v_0 \exp\left(-\frac{1}{2}(\Delta\mathbf{x}^T \cdot \text{diag}(w_1, \dots, w_m) \cdot \Delta\mathbf{x})\right) + a_0 + a_1 \mathbf{x}^T(i) \mathbf{x}(j) + v_1 \delta(i, j) \quad (2)$$

where $\Delta\mathbf{x} = \mathbf{x}(i) - \mathbf{x}(j)$

$v_0, v_1, a_0, a_1, w_1, \dots, w_m$ are hyperparameters to be determined. C_{ij} consists of an exponential term with factor v_0 , a constant bias term a_0 , a bilinear regression term with factor a_1 , and a noise term $v_1 \delta(i, j)$

where

$$\delta(i, j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

The w_1, \dots, w_m account for scaling of the distances in each input dimension

$x_1(i), \dots, x_m(i)$.

The exponential term suggests that nearby input vectors lead to highly correlated outputs.

The covariance matrix is denoted by

$$\mathbf{K} = \begin{pmatrix} C_{11} & \dots & C_{1n} \\ \vdots & C_{ij} & \vdots \\ C_{n1} & \dots & C_{nn} \end{pmatrix} \quad (3)$$

The aim is: given a new input vector $\mathbf{x}(n+1)$ find the distribution of the corresponding output $y(n+1)$. According to Bayes' law [7,14] a predictive Gaussian distribution for a test input vector $\mathbf{x}_{new} = \mathbf{x}(n+1)$ is obtained with mean and variance

$$\begin{aligned} \mu_{\hat{y}}(\mathbf{x}_{new}) &= \mathbf{k}^T(\mathbf{x}_{new}) \mathbf{K}^{-1} \mathbf{t} \\ \sigma_{\hat{y}}^2(\mathbf{x}_{new}) &= C(\mathbf{x}_{new}, \mathbf{x}_{new}) - \mathbf{k}^T(\mathbf{x}_{new}) \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}_{new}) \end{aligned} \quad (4)$$

where

$$\begin{aligned} \mathbf{k}(\mathbf{x}_{new}) &= (C(\mathbf{x}_{new}, \mathbf{x}(1)), \dots, C(\mathbf{x}_{new}, \mathbf{x}(n)))^T \\ \mathbf{K}_{ij} &= C_{ij} \\ \mathbf{t} &= (t(1), t(2), \dots, t(n))^T \end{aligned} \quad (5)$$

3. Gaussian processes for modeling of dynamical systems

The previous results are now applied to dynamical systems. Let a dynamical system be defined

$$\begin{aligned} \mathbf{x}(n+1) &= \mathbf{f}(\mathbf{x}(n), \mathbf{u}(n)) \\ \mathbf{y}(n) &= \mathbf{x}(n) \end{aligned} \quad (6)$$

where for a time step n the measured inputs to the system are $\mathbf{z}(n) = (\mathbf{x}(n), \mathbf{u}(n))^T$

$$\begin{aligned} \mathbf{x}(n) &= (x_1(n), x_2(n), \dots, x_r(n))^T \\ \mathbf{u}(n) &= (u_1(n), u_2(n), \dots, u_s(n))^T \\ \mathbf{z}(n) &= (x_1(n), x_2(n), \dots, x_r(n), u_1(n), u_2(n), \dots, u_s(n))^T \\ &= (z_1(n), z_2(n), \dots, z_{r+s}(n))^T \end{aligned} \quad (7)$$

with the output vector $\mathbf{y}(n+1)$ at time step $n+1$ (see Fig.2)

$$\mathbf{y}(n+1) = \mathbf{x}(n+1) = (x_1(n+1), x_2(n+1), \dots, x_r(n+1))^T \quad (8)$$

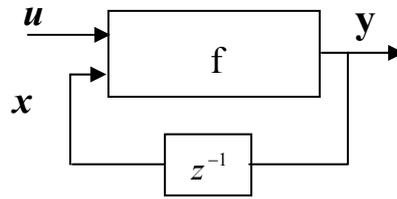


Fig 2. Input-output mapping for a dynamical system

Given the input-output data $\{\mathbf{z}(i), \mathbf{y}(i)\} \quad i=1 \dots n$ we compute the output $\mathbf{y}(n+1)$ by Gaussian estimation. According to (6-8) there are r individual models

$$\begin{aligned} x^k(n+1) &= f^k(\mathbf{z}(n)) \\ y^k(n) &= x^k(n) \end{aligned} \quad k=1 \dots r \quad (9)$$

with the covariance matrices

$$\mathbf{K}^k = \begin{pmatrix} C^k_{11} & \dots & C^k_{1n} \\ \vdots & C^k_{ij} & \vdots \\ C^k_{n1} & \dots & C^k_{nn} \end{pmatrix} \quad i, j=1 \dots n \quad (10)$$

where $C^k_{ij} = C^k(\mathbf{z}_i, \mathbf{z}_j)$.

Furthermore

$$\begin{aligned} \mathbf{k}^k &= (C^k(\mathbf{z}(n+1), \mathbf{z}(1)), \dots, C^k(\mathbf{z}(n+1), \mathbf{z}(n)))^T \\ \mathbf{t}^k &= (y^k(2), y^k(3), \dots, y^k(n))^T \\ W &= \text{diag}(w_1, \dots, w_r, w_{r+1}, \dots, w_{r+s}) \end{aligned} \quad (11)$$

$$C^k_{ij} = C^k(\mathbf{z}(i), \mathbf{z}(j)) = v_0 \exp\left(-\frac{1}{2}(\Delta \mathbf{z}^T \cdot \text{diag}(w_1, \dots, w_{r+s}) \cdot \Delta \mathbf{z})\right) + a_0 + a_1 \mathbf{z}^T(i) \mathbf{z}(j) + v_1 \delta(i, j) \quad (12)$$

$$\Delta \mathbf{z} = \mathbf{z}(i) - \mathbf{z}(j)$$

From this we obtain for mean and variance

$$\begin{aligned} \mu_{\hat{y}}^k(\mathbf{z}(n)) &= (\mathbf{k}^k)^T (\mathbf{K}^k)^{-1} \mathbf{t}^k \\ (\sigma_{\hat{y}}^k)^2 &= C^k(\mathbf{z}(n+1), \mathbf{z}(n+1)) - (\mathbf{k}^k)^T (\mathbf{K}^k)^{-1} \mathbf{k}^k \end{aligned} \quad (13)$$

4. One-step-ahead prediction

How can this approach be used for modeling and control? The modeling aspect is quite clear. In fact, if we have generated a representative set of data for the states, control inputs, and outputs we can find an estimate for the new output given the new state and the new control input which means that a model is created that interpolates between given points in the $\{\mathbf{z}(n), \mathbf{y}(n)\}$ -space. In the following the mean $\mu_{\hat{y}}^k$ is set to be equal to the estimate $\hat{y}^k(n+1)$. In order to compute $\hat{y}^k(n+1)$ the one-step-ahead prediction accounts for l previous consecutive samples $\mathbf{X} = (\mathbf{x}(n-l), \dots, \mathbf{x}(n))^T$ plus $\mathbf{x}(n+1)$ and a target vector $\mathbf{t}^k = (t^k(n-l), \dots, t^k(n))^T$ representing the previous l inputs for the Gaussian estimator. Once a new sample $n+1$ has arrived the oldest input/output pair is cancelled and a new inverse covariance matrix $(\mathbf{K}^k(n+1))^{-1}$ and new vectors $(\mathbf{k}^k(n+2))^T$ and $\mathbf{t}^k(n+1)$ are calculated (see Fig. 3).

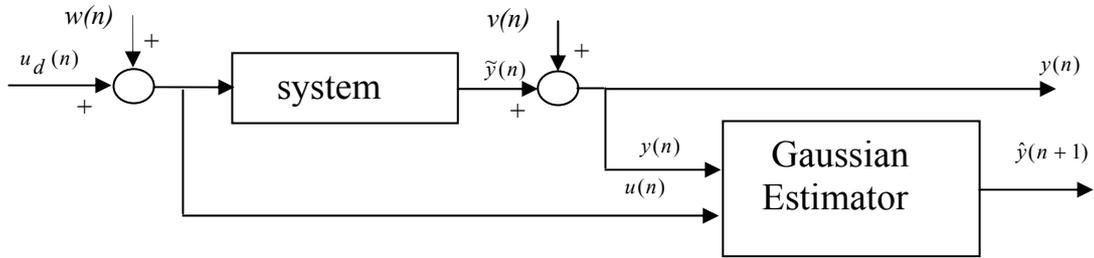


Fig. 3 Block scheme of one-step-ahead prediction by Gaussian processes

A 1st order linear SISO system illustrates this

$$\dot{x}(n) = a_{11}x(n) + a_{12}x^2(n) + bu_d(n) + w(n) \quad (14)$$

$$y(n) = x(n) + v(n)$$

$$\dot{x}(n) \approx (x(n) - x(n-1)) / dt$$

$$a_{11} = -1.4; a_{12} = -0.3; b = 1$$

$$u_d = 1.387 \cdot \sin(\omega \cdot n \cdot dt); \omega = 0.1;$$

$$d = 0.1,$$

$v(n), w(n)$ - Gaussian noise with $\sigma = 0.004$,

number of previous samples $l=10$ steps.

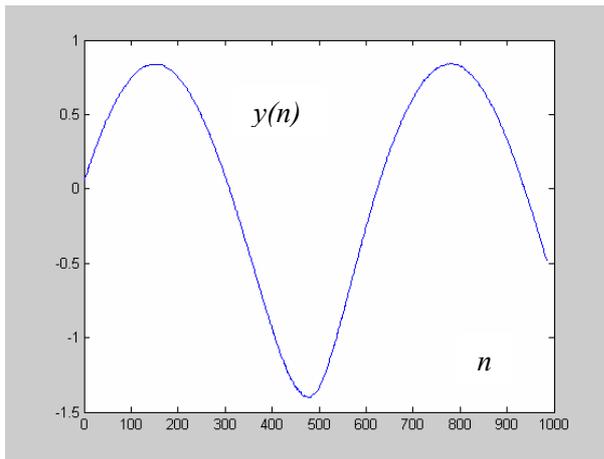
Figure 4 shows the simulation $x(n)$ of the system and the error

$e_{\hat{y}} = \hat{y}(n+1) - y(n+1)$ between estimated and real output with a very good result

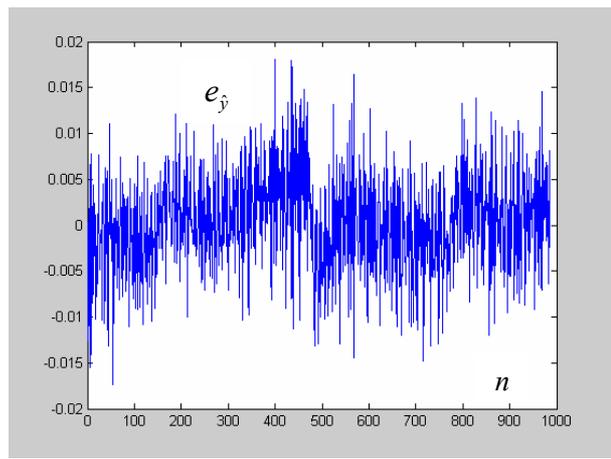
showing a standard deviation $\sigma_{e_{\hat{y}}} = 0.0057$. Increasing the standard deviation of the

added noise by a factor of ten yields a ten times higher standard deviation

$\sigma_{e_{\hat{y}}} = 0.0537$ (see Fig. 5).

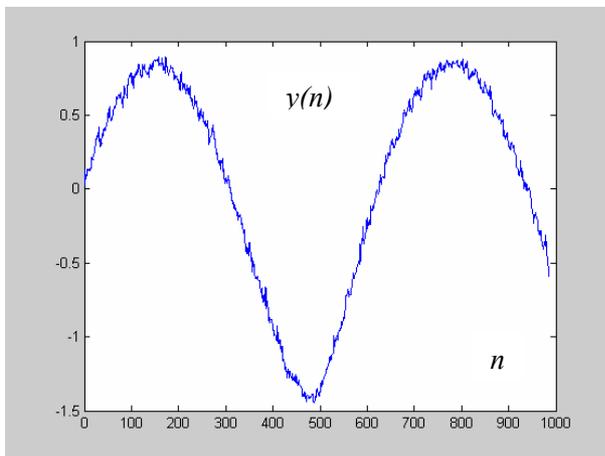


a) System output y

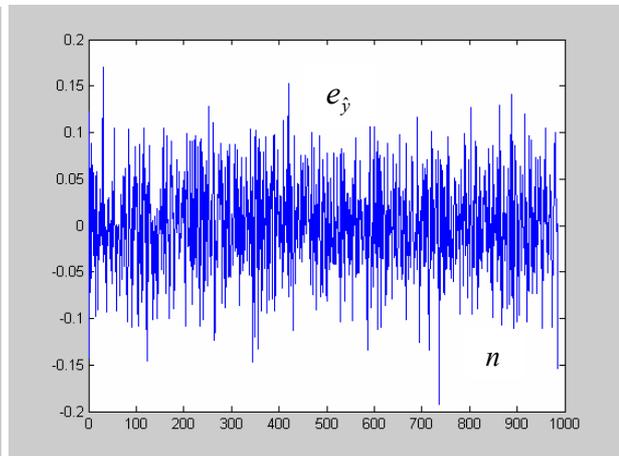


b) Error $e_{\hat{y}} = \hat{y}(n+1) - y(n+1)$

Fig. 4. One-step-ahead prediction for a linear system corrupted with noise $\sigma = 0.004$



a) System output y



b) Error $e_{\hat{y}} = \hat{y}(n+1) - y(n+1)$

Fig. 5. One-step-ahead prediction for a linear system corrupted with noise $\sigma = 0.04$

In order to illustrate that the method also works with time varying systems let the system parameter a_{11} be time-dependent $a_{11} = -1.4 + 1.3 \cdot \sin(\omega \cdot n \cdot dt)$. The standard deviation of the added noise is again $\sigma = 0.04$. The regarding results are shown in Fig. 6.

The resulting standard deviation of the error $e_{\hat{y}} = \hat{y}(n+1) - y(n+1)$ becomes

$\sigma_{e_{\hat{y}}} = 0.0676$. This shows that the method is also well applicable to mild time varying systems.

completed with the input $\mathbf{y}_{tot\ r-1} = (\mathbf{y}_{n+r-2}^T, \hat{y}(n+r-1))^T$, the control

$\mathbf{u}_{tot\ r-1} = (\mathbf{u}_{n+r-2}^T, u_d(n+r-1))^T$ and the output $\hat{y}(n+r)$ where

$\mathbf{y}_{n+r-2} = (y(n-l), \dots, y(n), \hat{y}(n+1), \dots, \hat{y}(n+r-2))^T$,

$\mathbf{u}_{n+r-2} = (u(n-l), \dots, u(n), u_d(n+1), \dots, u_d(n+r-2))^T$ (Block scheme see Fig. 9).

The term “naive” is due to the fact that this approach does not account for the uncertainty generated by each iteration step of prediction. In [1] an analytical way of propagating the uncertainty is shown. Figure 10 demonstrates an excellent simulation result of this ”naive” prediction approach.

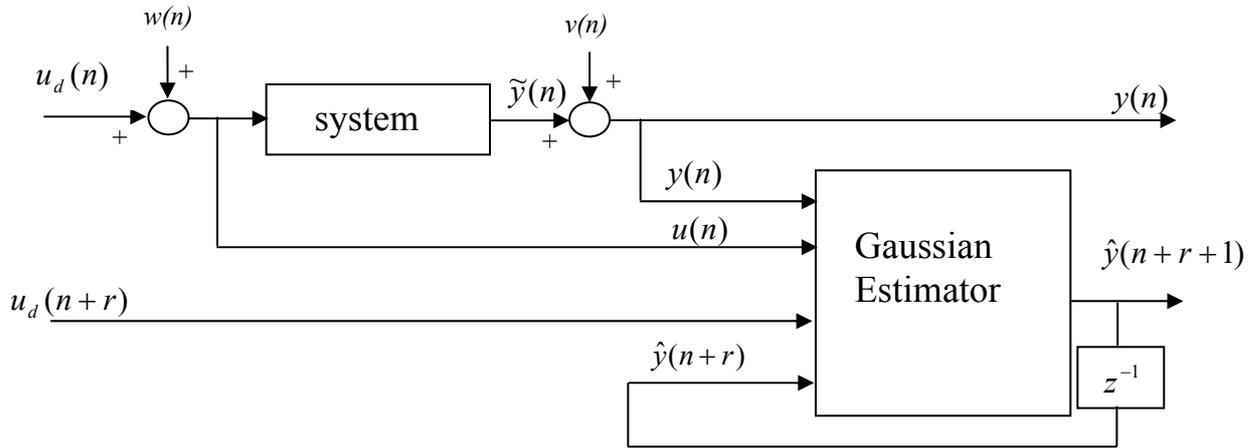


Fig. 9 Block scheme of the iterative r -step-ahead prediction

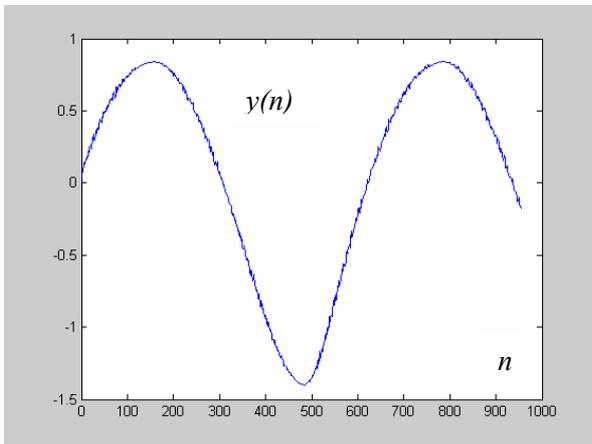


Fig. 10 a) System output y ($l=5, r=10$)

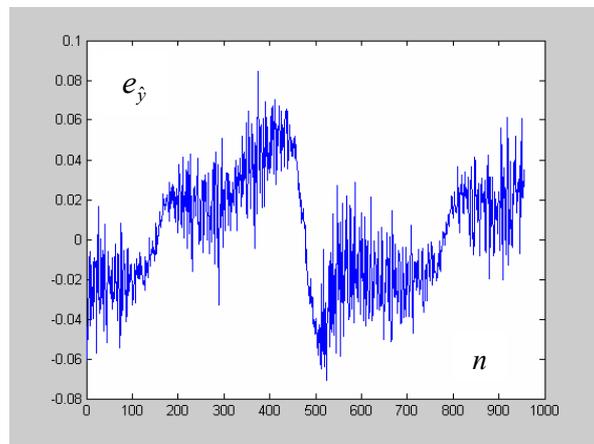


Fig. 10 b) Error between predicted and real y

$$\sigma_{e_{\hat{y}}} = 0.0287$$

6. On-line generation of the control trajectory using TS fuzzy models

In the previous examples it was supposed that the control trajectory is given in advance. However, in most cases this cannot be assumed. Usually one is faced with a closed loop system controlled by a given control law $u_d = Ctr(y, y_d)$. Let furthermore the nominal outputs $y_d(n)$ be defined in advance up to a given prediction horizon r .

Then the control inputs $u(n)$ of the system up to the present time step n are known but not the future steps. To achieve the nominal values for $u_d(n+r)$ a TS-model is built that generates the required control values in closed loop using the same control law as for the real system. Consider the open loop system of m -th order

$$\begin{aligned}\mathbf{x}(n+1) &= f(\mathbf{x}(n), u(n)) \\ y(n) &= x(n)\end{aligned}\quad (17)$$

where $\mathbf{x} = (x(n), \dots, x(n-m+1))^T$ is the state, u is the control variable, y is the output, f is a nonlinear function of \mathbf{x} and u . Let (17) be approximated by an *offline* trained Takagi-Sugeno Fuzzy system represented by c fuzzy rules

$$R^i : \text{IF } \mathbf{x}(n) \text{ is } \mathbf{X}^i \text{ AND } u \text{ is } U^i \text{ THEN } \mathbf{x}(n+1) = \mathbf{A}_i \mathbf{x}(n) + \mathbf{b}_i u(n) + \mathbf{q}_i \quad (18)$$

where \mathbf{X}^i, U^i are fuzzy sets, $\mathbf{A}_i \in \mathfrak{R}^{m \times m}$, $\mathbf{b}_i \in \mathfrak{R}^{m \times 1}$, $\mathbf{q}_i \in \mathfrak{R}^{m \times 1}$ are local matrices, c is the number of local linear models which are computed by fuzzy clustering and subsequent local modeling (linear regression) concluding with the multiple model

$$\mathbf{x}(n+1) = \sum_{i=1}^c w_i (\mathbf{A}_i \mathbf{x}(n) + \mathbf{b}_i u(n) + \mathbf{q}_i) \quad (19)$$

where $w_i = w_i(\mathbf{x}, u) \in (0, 1)$, $\sum_i w_i = 1$, is a weighting function [10].

Let furthermore the nominal output trajectory $y_d(n)$ and an appropriate control law $u_d = \text{Ctr}(y, y_d)$ be given. To generate a nominal control trajectory $u_d(n+r)$ for the future steps of the real system the control laws $u_d = \text{Ctr}(y, y_d)$ for the closed loop system and the closed loop TS fuzzy model are chosen to be identical. At each time step n the TS fuzzy model calculates a forecast for the next r time steps. These future control values are fed into the Gaussian estimator together with the previous inputs and outputs of the system and the r output estimates $\hat{y}(n+r)$ (see Fig. 11). In this way we have two models working: the *local* Gaussian estimator which is a *online* generated predicting the outputs for a given time horizon from noisy measurements and an *offline* trained *global* TS fuzzy model predicting the corresponding *nominal* control values.

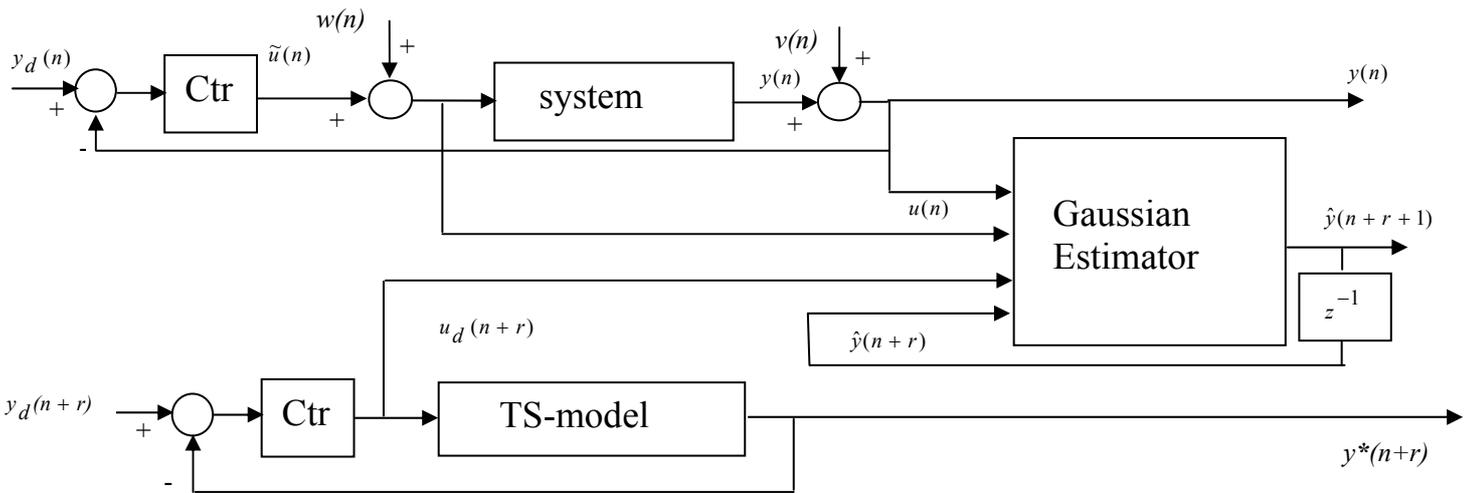


Fig. 11 Block scheme for on-line generation of the control trajectory

Figure 12 shows the results for a past horizon $l=5$ and a prediction horizon $r=10$. Compared with the last example without TS fuzzy model the prediction error is of the same small size. Fig. 12 c shows the error $e_{\hat{y}}$ for a time variant parameter

$$a_{11} = -1.4 + 0.3 \cdot \sin(\omega \cdot n \cdot dt).$$

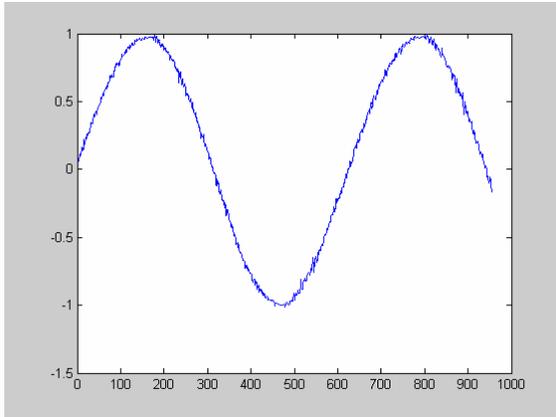


Fig. 12 a) System output y ($l=5, r=10$)

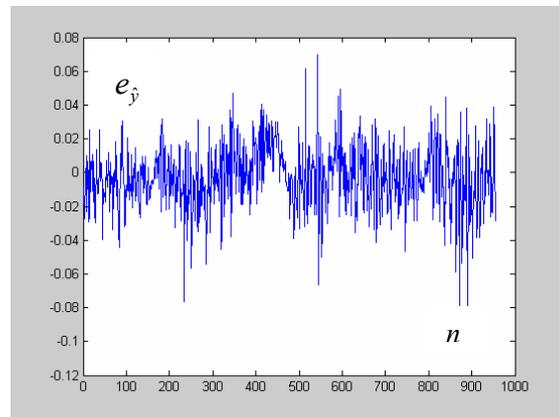


Fig. 12 b) Error between predicted and real y

$$\sigma_{e_{\hat{y}}} = 0.0245$$

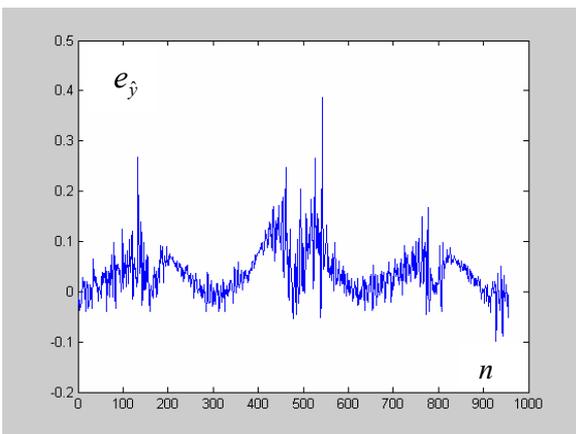


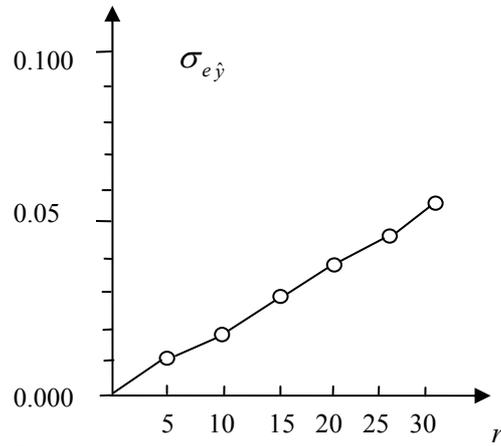
Fig. 12 c) Error between predicted and real time variant a_{11} ,

$$\sigma_{e_{\hat{y}}} = 0.0459$$

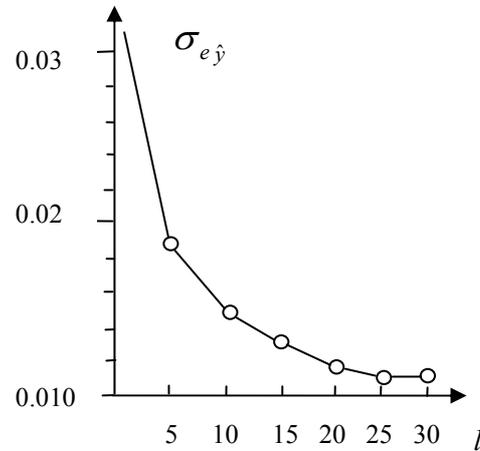
An interesting point is the evolution of the standard deviation $\sigma_{e_{\hat{y}}}$ of $e_{\hat{y}}$ depending both on the predictive horizon r and the past horizon l . It can be assumed that the standard deviation of $e_{\hat{y}}$ grows for a frozen past horizon l but an increasing predictive horizon r . On the other hand an increasing past horizon l with a fixed predictive horizon r leads to a declining standard deviation. This can be confirmed by the simulation examples the results of which are presented in Tables 1-2 and Figs. 13-14. For the increase of r an almost linear increase of $\sigma_{e_{\hat{y}}}$ can be identified (see Table 1 and Fig. 13). For the increase of l a step decrease of the standard deviation $\sigma_{e_{\hat{y}}}$ with a soon convergence to a constant value can be stated (see Table 2 and Fig. 14).

Table 1. Evolution $\sigma_{e\hat{y}}(r)$

$l=5, r$	$\sigma_{e\hat{y}}$
1	0.0083
5	0.0126
10	0.0193
15	0.0298
20	0.0392
25	0.0491
30	0.0574

Fig. 13 Evolution $\sigma_{e\hat{y}}(r)$ Table 2. Evolution $\sigma_{e\hat{y}}(l)$

$r=10, l$	$\sigma_{e\hat{y}}$
1	0.0690
5	0.0190
10	0.0145
15	0.0136
20	0.0117
25	0.0114
30	0.0115

Fig. 14 Evolution $\sigma_{e\hat{y}}(l)$

7. Conclusions

This paper deals with the modeling of dynamical systems with Gaussian process models and TS fuzzy models. This method is of advantage for one-step-ahead or even multi-step-ahead predictions with noisy time series and disturbed closed loop control systems. The approach is based on the Bayesian framework where the conditional distribution of output measurements of the system is used to make a one-step-ahead prediction for the system output. Normally global models are trained *offline* because of large training data sets. In this paper *local online modeling* with only few data is proposed where in each time step n the one-step-ahead prediction accounts for l previous consecutive input samples. The method was tested on disturbed time invariant and time variant systems for different past horizons. Increasing past horizons lead to decreasing standard deviations between predicted and measured data. The multi-step-ahead prediction assumes the previous outputs and control values to be known as well as the future control values. A “naive” multi-step-ahead prediction has been applied which is a successive one-step-ahead prediction whereas the outputs in each consecutive step are considered as inputs for the next step of prediction. For unknown future control inputs the system is modeled by a global TS fuzzy model which was trained in advance to generate a nominal control trajectory in closed loop for a given nominal output trajectory. Simulations with linear and nonlinear systems with built-in uncertainties show a good performance of the multi-step-ahead prediction with the combination of TS fuzzy models and Gaussian process models.

8. References

- [1] Girard A., Rasmussen, C.E., Murray-Smith, R.: Gaussian Process priors with Uncertainty Inputs: Multiple-Step-Ahead Prediction. Technical Report TR-2002-119, Dept. of Computer Science, University of Glasgow, Oct. 2002
- [2] Gregorcic G., Lightbody G.: Gaussian Process for Internal Model Control. Advances in Supervision and Control Systems, Strunjan 2002
- [3] Isermann, R.: Supervision, Fault-detection and fault-diagnosis Methods – An Introduction, *Control Eng. Practice* (1997), Vol. 5, No. 5, pp639-652
- [4] Johansen T. A., Shorten R., and Murray-Smith R. , "On the Interpretation and Identification of Takagi-Sugeno Fuzzy Models" , *IEEE Trans. Fuzzy Systems*, Vol. 8, pp. 297-313, 2000
- [5] Kocijan J., Murray-Smith R., Rasmussen C.E., Girard, A.: Gaussian Process Model Based Predictive Control. American Control Conference, Boston, 2004.
- [6] Ljung L., Glad T.: Modeling of dynamic systems. Prentice Hall 1994
- [7] MacKay, D.J.C.: Introduction to Gaussian processes. In Bishop, C., (ed.) Neural Networks and Machine Learning. NATO Asi Series. Series F, Computer and Systems Sciences, Vol. 168, 1998
- [8] Murray-Smith R., Sbarbaro D., Rasmussen C.E., Girard, A.: Adaptive, Cautious, Predictive control with Gaussian Process priors. 13th IFAC Symposium on System Identification, IFAC, Rotterdam, 2003
- [9] Palm R., Driankov D., Hellendoorn H.: Model based fuzzy control. Springer-Verlag Ltd., 1996
- [10] Palm R., Stutz Ch.: Open loop dynamic trajectory generator for a fuzzy gain scheduler. Engineering Applications of Artificial Intelligence EAAI 16 (2003) pp. 213-225
- [11] Takagi T., Sugeno M.: Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Trans. on Syst., Man, and Cyb., Vol. SMC-15. No. 1, pp. 116-132, January/February 1985.
- [12] Tanaka K., M. Sugeno M.: Stability Analysis and Design of Fuzzy Control Systems. Fuzzy Sets and Systems, North-Holland, No. 45, pp. 135-156, 1992.
- [13] Welch G., Bishop G.: An Introduction to the Kalman Filter. Technical Report, TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 2002
- [14] Williams C.K.I. and Rasmussen C.E. : Gaussian Processes for Regression. Proceedings of the 1995 Conference on "Advances in Neural Information Processing Systems 8" MIT Press 1996

Erweiterung des Fuzzy-ROSA-Verfahrens für die datenbasierte Regelgenerierung bei kleinen und stark gestörten Datensätzen

Martin Sternke, Peter Krause¹, Harro Kiendl²

Lehrstuhl für Systemanalyse

Universität Dortmund, 44221 Dortmund

Tel.: (0231) 9700-951 Fax: (0231) 9700-959

E-Mail: martin.sternke@cs.uni-dortmund.de

¹NuTech Solutions GmbH

Martin-Schmeisser-Weg 15, 44227 Dortmund

Tel.: (0231) 725463-0 Fax: (0231) 725463-29

E-Mail: krause@nutechsolutions.de

²E-Mail: kiendl@hyperfuzzy.de

Zusammenfassung

In den Bereichen des Data-Minings und der datenbasierten Modellierung werden verstärkt Fuzzy-Systeme eingesetzt. Von potenziellen Anwendern werden dabei an die Modelle folgende Anforderungen gestellt: Genauigkeit, Transparenz und Kompaktheit. Häufig werden Mamdani-Systeme oder Takagi-Sugeno-Kang(TSK)-Systeme eingesetzt. Mamdani-Systeme zeichnen sich dabei durch eine sehr gute Interpretierbarkeit aus, TSK-Systeme hingegen weisen eine höhere Approximationsgüte auf.

Eine Methode für die automatische und datenbasierte Generierung von Mamdani-Systemen ist das Fuzzy-ROSA-Verfahren¹. In diesem Beitrag werden zwei neue Strategieelemente hierfür vorgestellt. Das erste betrifft das Testen von Hypothesen (Regeln) innerhalb des Fuzzy-ROSA-Verfahrens. Eine neue Test- und Bewertungsstrategie, die auf dem Medianwert basiert, ist für die Regelgenerierung bei kleiner Datenbasis bzw. bei gestörten Daten vorteilhaft. Das zweite neue Strategieelement betrifft die in dem Fuzzy-ROSA-Verfahren optional vorgesehene Transformation eines generierten Mamdani-Systems in ein TSK-System. Nunmehr ist es möglich, diese Transformation so vorzunehmen, dass einzelne Ausreißer in den Daten und das Vorliegen kleiner Datensätze die Modellierungsgenauigkeit weniger stark als bisher beeinträchtigen. Die Anwendbarkeit der Verfahren und die damit erzielten Ergebnisse werden anhand von Beispielen gezeigt.

1 Einleitung

Mit zunehmender Komplexität moderner technischer wie auch nicht technischer Systeme besteht ein steigender Bedarf an Verfahren zur Modellierung solcher Systeme. Dabei werden in den Bereichen des Data-Minings und der datenbasierten Modellierung verstärkt Verfahren aus der Computational Intelligence eingesetzt. Fuzzy-Systeme haben sich dabei für viele Aufgabenstellungen als geeignet erwiesen und sie werden von Anwendern wegen ihrer Transparenz häufig besonders gut akzeptiert.

¹Regelorientierte statistische Analyse

Es gibt verschiedene Typen von Fuzzy-Systemen, die sich in Hinblick auf ihre Interpretierbarkeit und ihre Genauigkeit unterscheiden. Grundsätzlich sind beide Eigenschaften wünschenswert, bei vielen bekannten Verfahren muss jedoch einer der Vorteile zugunsten des anderen aufgegeben werden.

Zur automatischen und datenbasierten Generierung von Mamdani-Systemen kann z.B. das Fuzzy-ROSA-Verfahren eingesetzt werden. Dieses Verfahren basiert auf der statistischen Relevanzanalyse von Hypothesen zur Generierung von Fuzzy-Regeln, die für sich betrachtet einen relevanten Teilaspekt des zu modellierenden Systems beschreiben. Auf diese Weise wird eine sehr kompakte Regelbasis erzeugt. Die so generierten Mamdani-Regeln sind gut zu interpretieren, was für die Akzeptanz bei Anwendern von großer Bedeutung ist. Durch eine nachgeschaltete Transformation in TSK-Regeln kann die Modellgenauigkeit oftmals noch deutlich verbessert werden, wobei die Lesbarkeit der Regeln erhalten bleibt.

Der derzeitige Entwicklungsstand des Fuzzy-ROSA-Verfahrens wird in Abschnitt 2 — soweit für das Folgende benötigt — zusammengefasst. Das Verfahren hat derzeit Anwendungsgrenzen, wenn die Lerndatensätze nur einen geringen Umfang haben oder darin stark gestörte Daten enthalten sind. Solche Fälle werden in Anwendungen aber vielfach angetroffen. In den Abschnitten 3 und 4 werden zwei neue Strategieelemente — eine Medianwertbasierte Test- und Bewertungsstrategie für die Regeln und eine datensensitive Transformation von Mamdani- in TSK-Regeln — vorgestellt, mit denen sich die Modellierungsgüte bei Vorliegen derartiger Lerndatensätze verbessern lässt. Die Leistungsfähigkeit dieser Strategieelemente wird in Abschnitt 5 anhand von Beispielen gezeigt.

Die vorgestellten Arbeiten werden in enger Kooperation zwischen der Universität Dortmund und der NuTech Solutions GmbH im Rahmen eines von der Deutschen Forschungsgemeinschaft geförderten Transferbereiches durchgeführt.

2 Das Fuzzy-ROSA-Verfahren

Die zwei gängigsten Typen von Fuzzy-Systemen sind das Mamdani-System [1] und das TSK-System [2]. Ein Mamdani-System verwendet Regeln folgender Art:

$$\text{Wenn } \langle P_i \rangle \quad \text{Dann } \langle C_i \rangle \quad . \quad (1)$$

Bei einem TSK-System haben die Regeln die Form:

$$\text{Wenn } \langle P_i \rangle \quad \text{Dann } \langle F_i(\mathbf{x}) \rangle \quad . \quad (2)$$

Der Aufbau bzw. die Auswertung einer Regelprämisse P_i geschieht bei beiden Systemtypen in gleicher Weise. Die Systemtypen unterscheiden sich jedoch im Aufbau der Konklusion einer Regel und folglich auch in der Berechnung des Ausgangsgrößenwertes y_D des Fuzzy-Systems. Die Konklusion C_i einer Mamdani-Regel ist eine Elementaraussage bezogen auf die Ausgangsgröße. Im Gegensatz dazu ist $F_i(\mathbf{x})$ bei einer TSK-Regel eine frei wählbare Funktion der Eingangsgrößen.

Zu einer Eingangssituation \mathbf{x} wird bei einem Mamdani-System zunächst die ausgangsseitige Zugehörigkeitsfunktion $\mu(\mathbf{x}, y)$ über alle R Regeln wie folgt berechnet:

$$\mu(\mathbf{x}, y) = \bigvee_{i=1}^R [\mu_{P_i}(\mathbf{x}) \wedge \mu_{Y_{C_i}}(y)] \quad . \quad (3)$$

Dabei ist $\mu_{P_i}(\mathbf{x})$ der Erfülltheitsgrad der Prämisse P_i einer Regel und $\mu_{Y_{C_i}}(y)$ ist die Zugehörigkeit zu dem in der Konklusion C_i spezifizierten linguistischen Wert der Ausgangsgröße. Um aus der ausgangsseitigen Zugehörigkeitsfunktion $\mu(\mathbf{x}, y)$ einen scharfen Ausgangsgrößenwert y_D zu erhalten, wird z.B. die Schwerpunkt-Defuzzifizierung (COG) eingesetzt. Bei einem TSK-System geschieht dies durch eine gewichtete Mittelwertbildung über die Empfehlungen aller Regeln.

Das Fuzzy-ROSA-Verfahren [3] ermöglicht eine automatische datenbasierte Generierung eines Mamdani-Systems. Dabei besteht die grundlegende Idee darin, die Regelbasis aus einzelnen Regeln aufzubauen, die jeweils für sich betrachtet einen signifikanten Teilaspekt des zu modellierenden Systems beschreiben. Die Aufgabenstellung, eine geeignete Regelbasis zu erzeugen, wird so auf die Aufgabe reduziert, einzelne relevante Regeln zu finden. Auf diese Weise können transparente und kompakte Regelbasen generiert werden. Der Prozess der Modellerstellung lässt sich in folgende Unterpunkte gliedern: Projektdefinition, Regelgenerierung, Regelreduktion, Analyse und Optimierung der Regelbasis und Regeltransformation. Im Rahmen der Projektdefinition werden die Ein- und Ausgangsvariablen des zu modellierenden Systems ausgewählt und entsprechende linguistische Werte festgelegt. Weitere frei wählbare Parameter des Verfahrens müssen eingestellt werden. Nach Abschluss dieser Vorarbeiten ist der Suchraum für die nachfolgende Regelgenerierung strukturiert. Je nach Größe des Suchraumes kann zwischen einer vollständigen Suche, einer evolutionären Suche oder einer Mischform aus beiden Ansätzen gewählt werden. Die Anzahl der Regeln kann nachgeschaltet durch verschiedene Reduktionsverfahren verringert werden. Dabei ist zwischen unterschiedlichen Zielsetzungen zu unterscheiden, wie z.B. der möglichst gleichmäßigen Ausnutzung der Daten oder der Verringerung des Modellierungsfehlers. Zur Erhöhung der Modellgenauigkeit können die Mamdani-Regeln in TSK-Regeln transformiert werden. Dies geschieht auf eine Weise, die die Interpretierbarkeit der Regeln nicht tangiert.

Zwei spezielle Eigenschaften des Fuzzy-ROSA-Verfahrens werden an dieser Stelle erläutert, da sie nachfolgend von Bedeutung sind. Zum einen unterstützt das Fuzzy-ROSA-Verfahren die Verwendung generalisierender Regeln. Diese zeichnen sich dadurch aus, dass nicht alle Eingangsvariablen in der Prämisse vorkommen. Zum anderen können negative Regeln [4] generiert werden. Im Gegensatz zu herkömmlichen Regeln sprechen diese keine Empfehlungen, sondern Warnungen bzw. Verbote aus.

Die hier vorgestellten Arbeiten betreffen verschiedene Teilaspekte des Fuzzy-ROSA-Verfahrens. Zur Einordnung der Arbeiten werden diese Teilaspekte kurz vorgestellt und es wird der jeweilige Status vor Beginn der Erweiterung beschrieben.

2.1 Regelgenerierung

Eine potenziell relevante Regel wird als Hypothese formuliert und dann einem statistischen Test unterzogen um festzustellen, ob sie einen relevanten Teilaspekt des Systems beschreibt. Ist dies der Fall, wird die Regel geeignet bewertet und dann der Regelbasis hinzugefügt. Andernfalls wird die Regel verworfen. Auf diese Weise wird die Regelbasis sukzessive aus relevanten Regeln aufgebaut.

2.1.1 Test- und Bewertungsstrategien

Eine detaillierte Übersicht über die unterschiedlichen Test- und Bewertungsstrategien findet sich in [5]. Es kann generell zwischen zwei Klassen von Strategien unterschieden werden: zum einen solche, die anhand der Häufigkeit der Daten testen und bewerten und zum anderen solche, die dies anhand der Verteilung der Daten tun. Im Folgenden werden zwei ausgewählte Test- und Bewertungsstrategien genauer vorgestellt.

Relevanzindex

Diese Strategie gehört zu der ersten Klasse. Der Relevanztest prüft die Frage, ob zwischen der Prämisse P und der Konklusion C einer Regel ein kausaler Zusammenhang besteht. Die Grundidee basiert darauf, die unbedingte Wahrscheinlichkeit $p(C)$ für das Auftreten der Konklusion C und die bedingte Wahrscheinlichkeit $p(C|P)$ für das Auftreten der Konklusion C unter der Voraussetzung, dass die in der Prämisse P definierte Eingangssituation vorliegt, miteinander zu vergleichen. Da diese Wahrscheinlichkeiten unbekannt sind, wird der Vergleich anhand von Schätzwerten durchgeführt, die basierend auf den D vorhandenen Daten bestimmt werden:

$$\hat{p}(C) = \frac{1}{D} \cdot \sum_{q=1}^D \mu_{Y_j}(y_q) \quad \text{und} \quad \hat{p}(C|P) = \frac{1}{D_{sup}} \cdot \sum_{q=1}^D \mu_{Y_j}(y_q) \wedge \mu_P(\mathbf{x}_q) \quad . \quad (4)$$

Dabei bezeichnet D_{sup} den Daten-Support der Prämisse P :

$$D_{sup} = \sum_{q=1}^D \mu_P(\mathbf{x}_q) \quad . \quad (5)$$

Zu den Schätzungen der o.a. Wahrscheinlichkeiten wird jeweils ein einseitiges Konfidenzintervall $I_l^\alpha = [0, \hat{p}_l^\alpha]$ bzw. $I_u^\alpha = [\hat{p}_u^\alpha, 1]$ bestimmt, in dem sich die geschätzte Wahrscheinlichkeit \hat{p} mit einer vorgegebenen Irrtumswahrscheinlichkeit α befindet [6]. Eine zu testende Regel wird als positive bzw. negative Regel angesehen, wenn $\hat{p}_u^\alpha(C|P) > \hat{p}_l^\alpha(C)$ bzw. $\hat{p}_l^\alpha(C|P) < \hat{p}_u^\alpha(C)$ gilt. Eine als relevant eingestufte Regel wird mit dem Relevanzindex J_{RI} bewertet:

$$J_{RI} = \begin{cases} \frac{\hat{p}_u^\alpha(C|P) - \hat{p}_l^\alpha(C)}{1 - \hat{p}(C)} & \text{für positive Regeln} \\ \frac{\hat{p}_u^\alpha(C) - \hat{p}_l^\alpha(C|P)}{1 - \hat{p}(C)} & \text{für negative Regeln} \end{cases} \quad . \quad (6)$$

Mittelwertbasierter Test

Diese Strategie [7] orientiert sich für das Testen und Bewerten nicht an den relativen Häufigkeiten der Daten, sondern an der Datenverteilung². Es werden zunächst für die unbedingte Verteilung der Mittelwert \bar{y} und die Standardabweichung σ der Ausgangsgröße berechnet. Gleiches wird für die bedingte Verteilung durchgeführt:

$$\bar{y}_P = \frac{1}{D_{sup}} \cdot \sum_{q=1}^D \mu_P(\mathbf{x}_q) \cdot y_q \quad . \quad (7)$$

$$\sigma_P = \sqrt{\frac{\sum_{q=1}^D \mu_P(\mathbf{x}_q) \cdot (y_q - \bar{y}_P)^2}{D_{sup} - 1}} \quad . \quad (8)$$

Mit Hilfe des Mittelwertbasierten Tests wird untersucht, ob sich die Mittelwerte der unbedingten und der bedingten Verteilung signifikant voneinander unterscheiden. Als zu überprüfende Nullhypothese H_0 wird angenommen, dass die Mittelwerte gleich sind. Zur Überprüfung wird der statistische t -Test eingesetzt. Der Wert t_n der Prüfgröße berechnet sich für die zu untersuchenden Verteilungen wie folgt:

$$t_n = \left| \frac{\bar{y}_P - \bar{y}}{\sigma_P / \sqrt{D_{sup}}} \right| \quad . \quad (9)$$

Die Nullhypothese H_0 kann für eine gegebene Irrtumswahrscheinlichkeit α verworfen werden, falls $t_n > t(D_{sup}, \alpha)$ gilt. In diesem Fall unterscheiden sich die Mittelwerte signifikant voneinander und die Regel wird als zulässig erklärt. Der kritische Wert $t(D_{sup}, \alpha)$ kann dabei aus Tabellen entnommen oder geeignet approximiert werden. Der Bewertungsindex für eine übernommene Regel wird definiert als:

$$J_{MVB} = e^{-\sigma_P/\sigma} \quad . \quad (10)$$

2.2 Transformation in ein TSK-Fuzzy-System

Mit dem Fuzzy-ROSA-Verfahren können Mamdani-Systeme generiert werden, die Regeln in der Form von Gl. (1) verwenden. Regeln dieses Typs sind gut zu interpretieren, was insbesondere für die Akzeptanz bei industriellen Anwendern von großer Bedeutung ist. Neben der Interpretierbarkeit wird jedoch auch eine hohe Approximationsgüte gefordert. In dieser Hinsicht sind TSK-Systeme besser geeignet, sie sind jedoch weniger gut zu interpretieren. Um diese konträren Eigenschaften der unterschiedlichen Systemtypen zu überwinden, wurde ein Verfahren entwickelt, das die Mamdani-Regeln unter Beibehaltung der Lesbarkeit in TSK-Regeln umwandelt [8, 9]. Die Transformation ist z.Z. auf positive Regeln beschränkt. Negative Regeln können derzeit noch nicht transformiert werden.

²Mit Strategien dieser Klasse lassen sich ausschließlich positive Regeln erzeugen.

2.2.1 Auswahl der Daten und der Eingangsgrößen

Zur Transformation einer Regel sind verschiedene Strategien möglich [8]. Diese unterscheiden sich in Hinblick auf die Auswahl der Daten, die zur Transformation herangezogen werden und in Hinblick auf die Eingangsvariablen, die verwendet werden.

2.2.2 Ansätze für die Approximationsfunktion

Für die Konklusion $F_i(\mathbf{x})$ kommen prinzipiell beliebig komplexe Ansätze in Frage. Dabei ist zu bedenken, dass mit zunehmender Komplexität u.U. die Gefahr eines Overfitting besteht und zudem der Berechnungsaufwand steigt. Im Fuzzy-ROSA-Verfahren wird folgender Ansatz für $F_i(\mathbf{x})$ verwendet:

$$F_i(\mathbf{x}) = a_{i,0} + \sum_{j=1}^m a_{i,j} \cdot g_j(\mathbf{x}) \quad . \quad (11)$$

In diesem Fall ist $F_i(\mathbf{x})$ eine Linearkombination verschiedener Elementarfunktionen $g_j(\mathbf{x})$, wobei diese sowohl lineare als auch nicht lineare Funktionen sein können. Das Fuzzy-ROSA-Verfahren unterstützt z.Z. die Approximation durch eine Hyperebene und durch eine multilineare Funktion.

2.2.3 Least-Squares-Approximation

Im Fall einer Linearkombination können die freien Parameter $a_{i,j}$ von $F_i(\mathbf{x})$ mit Hilfe eines Least-Squares-Algorithmus bestimmt werden, sofern ausreichend viele linear unabhängige Daten die zu transformierende Regel stützen. Das zu lösende Problem kann in Matrixform wie folgt geschrieben werden:

$$\mathbf{G} \cdot \mathbf{a} = \mathbf{y} \quad . \quad (12)$$

Der Least-Squares-Algorithmus findet den Satz \mathbf{a} von Parametern, der die Gütefunktion $J_1(\mathbf{a})$ minimiert:

$$J_1(\mathbf{a}) = \sum_{q=1}^{D_R} (\mathbf{g}^T(\mathbf{x}_q) \mathbf{a} - y_q)^2 \quad . \quad (13)$$

Dabei ist D_R die Anzahl der Daten, die zur Transformation herangezogen werden. Falls nicht ausreichend viele linear unabhängige Daten die Regel stützen, wird der Ausgangsgrößenmittelwert der entsprechenden Daten verwendet. Nur die erste Komponente des Lösungsvektors \mathbf{a} ist dann von Null verschieden.

3 Medianwertbasierte Test- und Bewertungsstrategie

Bei schwachen Zusammenhängen in den Daten stoßen die derzeit vorgesehenen Test- und Bewertungsstrategien des Fuzzy-ROSA-Verfahrens auf Anwendungsgrenzen. Derartig schwache Zusammenhänge treten insbesondere dann auf, wenn die

Ausgangsgröße in sehr komplexer Weise von den Eingangsgrößen abhängt oder die Datensätze sehr klein sind oder durch stochastische Störungen oder unbekannte Einflussgrößen verfälscht werden. Bei Test- und Bewertungsstrategien, die — wie der o.a. Relevanzindex — auf relativen Häufigkeiten basieren, liegt die Ursache hierfür oftmals darin, dass die Konfidenzintervalle zu den geschätzten Wahrscheinlichkeiten bedingt durch die geringe statistische Absicherung an Daten entsprechend groß ausfallen und daher oftmals nicht disjunkt sind. Dies hat dann zur Folge, dass nur sehr wenige Regeln generiert werden und die Modellierung entsprechend schlecht gelingt.

Vor diesem Hintergrund ist die Mittelwertbasierte Test- und Bewertungsstrategie eingeführt worden. Diese Strategie ist in Hinblick auf kleine Datensätze und Störungen vergleichsweise robuster. Dennoch birgt die Verwendung des Mittelwertes bezogen auf eine kleine Stichprobe von gestörten Daten oft eine unerwünscht große Unsicherheit, die zudem durch Ausreißer stark beeinflusst wird. Günstigere Eigenschaften in Hinblick auf die Robustheit hat die im Folgenden vorgestellte Test- und Bewertungsstrategie auf der Basis des Medianwertes.

3.1 Wilcoxon-Rangsummen-Test

Ein Verfahren zum Testen von Hypothesen über Medianwerte ist der Wilcoxon-Rangsummen-Test [10]. Dieser Test liefert eine Aussage darüber, ob sich die Mediane zweier Stichproben signifikant voneinander unterscheiden. Die beiden Stichproben werden zunächst zu einer gepoolten Probe zusammengefasst und nach aufsteigenden Werten sortiert. Die sortierten Werte werden mit einer fortlaufenden Rangnummer versehen. Falls in der sortierten gepoolten Probe *Tieds* (identische Werte) auftreten, werden die jeweiligen Rangnummern entsprechend gemittelt. Das weitere Vorgehen ist abhängig von dem Umfang der Stichproben. Für insgesamt kleine³ Stichproben kann eine spezielle Teststrategie verfolgt werden. Bei einem entsprechend großen Umfang der Stichproben werden die Rangnummern nach Stichproben getrennt aufsummiert und daraus die Prüfgröße z gebildet. Die Prüfgröße z verhält sich in diesem Fall näherungsweise wie eine Standard-Normalverteilung. Es wird das z -Quantil u_z der Standard-Normalverteilung approximiert. Falls $u_z < \alpha$ gilt, so unterscheiden sich die Mediane signifikant voneinander, andernfalls nicht.

3.2 Anwendung innerhalb des Fuzzy-ROSA-Verfahrens

Die mit dem Wilcoxon-Rangsummen-Test zu untersuchenden Stichproben ergeben sich durch die unbedingte und die bedingte Datenverteilung, die sich für die jeweils zu testende Regel einstellen. Falls der Test bestanden wird, muss die als relevant eingestufte Regel geeignet bewertet werden. Die Relevanz der Regel wird gemäß des Bewertungsindex J_{CVB} wie folgt festgelegt:

$$J_{CVB} = e^{-sP/\varsigma} \quad . \quad (14)$$

³Als „insgesamt klein“ kann z.B. der Fall angesehen werden, dass die kleinere Stichprobe weniger als zehn Daten enthält und die gepoolte Probe weniger als 20 Daten. Da die gepoolte Probe aber immer mindestens den Umfang der Lerndaten (unbedingte Verteilung) hat, tritt dieser Fall in der Praxis nur sehr selten auf.

Dabei sind ς bzw. ς_P die *median average deviation* der unbedingten bzw. der bedingten Verteilung. Dieses Maß für die Streuung ist robuster gegenüber einzelnen Ausreißern als z.B. die Standardabweichung. Es orientiert sich ebenfalls an dem Medianwert und eignet sich daher für den Einsatz in Verbindung mit dem Wilcoxon-Rangsummen-Test. Für die unbedingte Verteilung wird ς hier wie folgt berechnet:

$$\varsigma = \text{median}(|y_{\xi_{0.5}} - y_j|) \quad . \quad (15)$$

Dabei ist $y_{\xi_{0.5}}$ der Median der unbedingten Verteilung. Für die bedingte Verteilung wird die Berechnung geringfügig modifiziert:

$$\varsigma_P = \text{median}(\mu_P(\mathbf{x}_j) \cdot |y_{P_{\xi_{0.5},w}} - y_j|) \quad . \quad (16)$$

Dabei ist $y_{P_{\xi_{0.5},w}}$ der gewichtete Median der bedingten Verteilung. Bei dieser Form der Medianbestimmung wird für jedes Datum $d_j = [\mathbf{x}_j, y_j]$ der bedingten Verteilung der jeweilige Erfülltheitsgrad $\mu_P(\mathbf{x}_j)$ der Prämisse mitbeachtet.

Die Exponentialfunktion wird für die Bewertung verwendet, um den Relevanzwert einer Regel auf das Intervall $[0, 1]$ zu begrenzen. Die Relevanz einer Regel wird dabei umso höher, je kleiner das Verhältnis ς_P/ς ist.

4 TSK-Transformation bei kleinen und/oder stark gestörten Datensätzen

Die obige Einführung der neuen Medianwertbasierten Test- und Bewertungsstrategie für Regeln wird durch zwei Aspekte motiviert, die im Rahmen einer datenbasierten Modellierung generell von Bedeutung sind: zum einen Ausreißer in den zu modellierenden Daten und zum anderen die Modellierung bei kleiner Datenbasis. Im Folgenden wird gezeigt, wie diese Aspekte auch bei der Transformation von Mamdani- in TSK-Regeln berücksichtigt werden können.

4.1 Robuste Regression

4.1.1 Motivation

Ersichtlich tragen in Gl. (13) alle Daten gleichermaßen zu der Gütefunktion J_1 bei. Wenige, evtl. fehlerhafte Daten können so die resultierende Approximation entscheidend beeinflussen. Dies entspricht der oben diskutierten Anfälligkeit des Mittelwertes, der dort durch den vorgestellten Übergang zur Verwendung des Median begegnet wird. Es ist daher übertragen auf das vorliegende Problem der Transformation einer Mamdani-Regel in eine TSK-Regel nahe liegend, eine Lösung anzustreben, die vergleichsweise robuster gegen Ausreißer ist. Dies ist der Ansatz, den die robuste Regression [11, 12] verfolgt.

Selbstverständlich kann eine so gefundene Lösung $\tilde{\mathbf{a}}$ bzgl. der quadratischen Gütefunktion J_1 nicht besser sein kann als die Lösung \mathbf{a} , die aus der Minimierung von J_1 resultiert. Ein Anwender wird bei der Bewertung einer Lösung jedoch nicht notwendig diese quadratische Gütefunktion, sondern ein anderes Gütemaß wie z.B. den

absoluten Fehler:

$$J_2(\mathbf{a}) = \sum_{q=1}^{D_R} |(\mathbf{g}^T(\mathbf{x}_q) \mathbf{a} - y_q)| \quad (17)$$

für aussagekräftiger halten. Wenn nun $J_1(\tilde{\mathbf{a}}) > J_1(\mathbf{a})$, aber gleichzeitig $J_2(\tilde{\mathbf{a}}) < J_2(\mathbf{a})$ gilt, so wird der Anwender die Lösung $\tilde{\mathbf{a}}$ bevorzugen.

4.1.2 Verfahren

Zu jedem Datum $d_i = [\mathbf{x}_i, y_i]$ wird zunächst ein Maß $0 < \iota_i < 1$ berechnet, das den Einfluss des Datums auf die Regression anhand seiner Lage im Raum der Eingangsgrößen \mathbf{x} unter Beachtung der übrigen Daten beschreibt. Dieses Maß wird als *leverage* [13] bezeichnet. Je größer der Wert ist, desto relevanter ist das Datum für die Regression. Aus diesem Maß wird für jedes Datum ein Modifikationsfaktor h_i berechnet:

$$h_i = 1/\sqrt{(1 - \iota_i)} \quad . \quad (18)$$

Der nachfolgende Pseudocode zeigt den Ablauf des Verfahrens. Ausgangspunkt ist die Bestimmung der Least-Squares-Lösung in Bezug auf die unmodifizierten Daten \mathbf{G} und \mathbf{y} . Es folgt eine iterative Wiederholung verschiedener Schritte, solange die maximal zulässige Anzahl an Iterationen *MaxNumIter* nicht erreicht worden ist und gleichzeitig eine signifikante Veränderung $\Delta\mathbf{a}$ in der gefundenen Lösung auftritt.

procedure Robuste Regression

- 1: $\mathbf{a}, \tilde{\mathbf{a}} \leftarrow$ Least-Squares-Approximation (\mathbf{G}, \mathbf{y})
- 2: $\Delta\mathbf{a} \leftarrow (Tol + 1), NumIter \leftarrow 0$
- 3: **while** ($NumIter < MaxNumIter$ **AND** $\Delta\mathbf{a} > Tol$) **do**
- 4: $\hat{\mathbf{y}} \leftarrow$ Approximation berechnen (\mathbf{G}, \mathbf{a})
- 5: $\mathbf{r} \leftarrow$ Residuen berechnen ($\mathbf{y}, \hat{\mathbf{y}}$)
- 6: $\tilde{\mathbf{r}} \leftarrow$ Residuen anpassen (\mathbf{r}, \mathbf{h})
- 7: $\mathbf{w} \leftarrow$ Gewichtungen berechnen ($\tilde{\mathbf{r}}$)
- 8: $(\tilde{\mathbf{G}}, \tilde{\mathbf{y}}) \leftarrow$ Daten modifizieren ($\mathbf{G}, \mathbf{y}, \mathbf{w}$)
- 9: $\tilde{\mathbf{a}} \leftarrow$ Least-Squares-Approximation ($\tilde{\mathbf{G}}, \tilde{\mathbf{y}}$)
- 10: $\Delta\mathbf{a} \leftarrow$ Veränderung berechnen ($\mathbf{a}, \tilde{\mathbf{a}}$)
- 11: $\mathbf{a} \leftarrow \tilde{\mathbf{a}}$
- 12: $NumIter \leftarrow NumIter + 1$
- 13: **end while**

Die Anpassung der Residuen geschieht in der Art, dass jedes Residuum r_i zunächst mit dem zugehörigen Modifikationsfaktor h_i gewichtet wird:

$$\tilde{r}_i = r_i \cdot h_i \quad . \quad (19)$$

Somit wird ein Residuum umso mehr verstärkt, je stärker der Einfluss des zugehörigen Datums auf die Regression ist. Die Standardabweichung $\hat{\sigma}$ der angepassten Residuen wird dann mit Hilfe deren *median average deviation* geschätzt, um die Residuen ein weiteres Mal anzupassen. Es gibt verschiedene Ansätze, aus den so angepassten Residuen die Gewichtungen zur Modifikation der Daten zu berechnen

[12]. Der hier verwendete Ansatz für die Gewichtungsfunktion lautet:

$$w_i = \begin{cases} 0 & \text{falls } |\tilde{r}_i| \geq 1 \\ (1 - \tilde{r}_i^2) & \text{falls } |\tilde{r}_i| < 1 \end{cases} . \quad (20)$$

Die Modifikation der Daten mit Hilfe der Gewichtungen geschieht wie folgt:

$$\tilde{G}_{i,j} = G_{i,j} \cdot w_i \quad \text{bzw.} \quad \tilde{y}_i = y_i \cdot w_i . \quad (21)$$

Ein schlecht approximiertes Datum $d_i = [\mathbf{x}_i, y_i]$ mit $w_i = 0$ wird so folglich zu $\tilde{d}_i = [0, \dots, 0]$ transformiert. Es ist ersichtlich, dass so manipulierte Daten keinen entscheidenden Einfluss auf die Bestimmung der Lösung \tilde{a} gemäß Gl. (13) haben.

4.2 Transformation bei zu wenig linear unabhängigen Daten

Steht für die Modellgenerierung nur eine relativ kleine Datenbasis zur Verfügung, so kann verstärkt der Fall auftreten, dass das derzeit im Fuzzy-ROSA-Verfahren vorgesehene Verfahren zur Transformation einer Mamdani-Regel in eine TSK-Regel nicht angewendet werden kann, da nicht ausreichend viele linear unabhängige Daten die zu transformierende Regel stützen. In diesem Zusammenhang ist mit „kleiner“ Datenbasis nicht die absolute Anzahl von Daten gemeint, sondern vielmehr das Verhältnis der Anzahl von Daten zu der Anzahl von freien Funktionsparametern. Speziell bei höherdimensionalen Problemen besteht daher tendenziell eher die Gefahr, dass sich die zuvor gefundenen Regeln nicht auf die in Abschnitt 2.2.3 beschriebene Weise transformieren lassen. Eine einfache im Fuzzy-ROSA-Verfahren für diesen Fall bisher vorgesehene Defaultstrategie besteht darin, den Ausgangsgrößenmittelwert der Daten zu verwenden, die die zu transformierende Regel stützen. Der Vorteil dieser Strategie liegt in dem geringen Aufwand und der einfachen Realisierung. Nachteilig ist, dass sich so keine signifikante Erhöhung der Modellgenauigkeit gegenüber einem Mamdani-System erzielen lässt.

Da in den Anwendungen zunehmend Modellierungsaufgaben auftreten, bei denen die Datenbasis — häufig aus Kostengründen — vergleichsweise klein ist, ist es sinnvoll, diese einfache Defaultstrategie durch eine zwar aufwendigere, aber im Ergebnis günstigere Strategie zu ersetzen. Sie besteht darin, eine Lösung \mathbf{a} zu berechnen, die höchstens $k = \text{rank}(\mathbf{G})$ von Null verschiedene Komponenten aufweist [14].

5 Beispiele

In den Beispielen sind die vorhandenen Daten jeweils zufällig in zwei gleich große Datensätze (Lern- und Validierungsdaten) geteilt worden. Die Ein- und Ausgangsvariablen sind jeweils durch fünf gleichmäßig verteilte linguistische Werte partitioniert worden. Die maximal zulässige Anzahl an Elementaraussagen in einer Prämisse betrug $c_{max} = 2$. Für den Relevanzindex J_{RI} wurde die zulässige Irrtumswahrscheinlichkeit zu $\alpha = 0.1$ gewählt, bei dem Mittelwert- bzw. bei dem Medianwertbasierten Test (J_{MVB} bzw. J_{CVB}) jeweils zu $\alpha = 0.01$. Bei den TSK-Modellen ist als Approximationsfunktion $F_i(\mathbf{x})$ eine Hyperebene verwendet worden.

Es werden jeweils, nach Datensätzen getrennt, der mittlere absolute (MAE) und der

mittlere quadratische (MSE) Modellierungsfehler angeben. Vergleiche des Fuzzy-ROSA-Verfahrens mit anderen Verfahren finden sich in [5, 8, 9]. Auf Grund unterschiedlicher Lern- und Validierungsdatensätze sowie teilweise voneinander abweichender Projektdefinitionen sind die hier vorgestellten Ergebnisse nur eingeschränkt mit den dort aufgeführten vergleichbar.

5.1 Mackey-Glass Zeitfolge

Diese chaotische Zeitfolge [15] wird häufig zum Testen von Modellierungsverfahren genutzt. Die Aufgabe besteht darin, anhand des aktuellen Messwertes $x(t)$ und dreier zurückliegender Messwerte $x(t - \Delta t_i)$ den Wert $y = x(t + 6)$ vorherzusagen. Die Ausgangsgröße y kann dabei Werte zwischen 0.37 und 1.35 annehmen. Die Anzahl der Eingangsgrößen beträgt $V = 4$ und die Anzahl der Daten $D = 1000$.

J	Regeln	Typ	Lerndaten		Validaten		Gesamtdaten	
			MAE	MSE	MAE	MSE	MAE	MSE
J_{RI}	91	Mamdani	0.0506	0.0039	0.0514	0.0041	0.0510	0.0040
J_{RI}	91	TSK _{LSA}	0.0284	0.0012	0.0291	0.0012	0.0287	0.0012
J_{RI}	91	TSK _{RR}	0.0254	0.0011	0.0261	0.0011	0.0258	0.0011
J_{MVB}	94	Mamdani	0.0857	0.0101	0.0889	0.0106	0.0873	0.0103
J_{MVB}	94	TSK _{LSA}	0.0287	0.0013	0.0293	0.0013	0.0292	0.0013
J_{MVB}	94	TSK _{RR}	0.0254	0.0011	0.0255	0.0010	0.0255	0.0010
J_{CVB}	85	Mamdani	0.0808	0.0090	0.0828	0.0091	0.0818	0.0091
J_{CVB}	85	TSK _{LSA}	0.0286	0.0013	0.0291	0.0013	0.0289	0.0013
J_{CVB}	85	TSK _{RR}	0.0252	0.0010	0.0255	0.0010	0.0253	0.0010

Tabelle 1: Modellierungsergebnisse für *Mackey-Glass Zeitfolge*

Aus Tabelle 1 geht hervor, dass der Relevanzindex J_{RI} in diesem Fall am besten für die Erstellung eines Mamdani-Systems geeignet ist. Die Modellierungsfehler sind deutlich kleiner als die bei dem Mittelwertbasierten Test J_{MVB} bzw. dem Medianwertbasierten Test J_{CVB} . Vergleicht man nur die beiden letzteren Tests miteinander, so liefert der Medianwertbasierte Test bei einer ca. 10% kleineren Regelbasis einen ca. 6% kleineren Fehler als der Mittelwertbasierte Test. Durch die Transformation in ein TSK-System wird der Modellierungsfehler in allen Fällen erheblich verkleinert. Durch den Einsatz der robusten Regression (TSK_{RR}) wird der mittlere absolute Modellfehler (MAE) gegenüber der Least-Squares-Approximation (TSK_{LSA}) nochmals um ca. 10% reduziert.

5.2 Boston Housing Problem

Die Aufgabe bei diesem Benchmarkproblem besteht darin, anhand verschiedener Merkmale die Grundstückspreise in der Stadt Boston zu modellieren bzw. zu prognostizieren [16]. Dabei liegen die Grundstückspreise zwischen 5000\$ und 50000\$. In Tabelle 2 sind die Fehlerwerte MAE bzw. MSE jeweils in 1000\$ angegeben. Die Anzahl der Eingangsgrößen beträgt $V = 13$ und die Anzahl der Daten $D = 506$.

Der Einsatz des Relevanzindex J_{RI} ist in diesem Fall eher ungünstig. Mit dem Mittelwertbasierten Test J_{MVB} gelingt die Modellierung deutlich besser. Auch bei diesem Beispiel wird durch die Transformation in ein TSK-System der Modellierungsfehler erheblich verkleinert. Es tritt dabei aber zudem der Fall ein, dass auf die einfache Ersatzstrategie (Mittelwertprognose) zurückgegriffen werden muss. Wird hingegen die aufwendigere Strategie zur Behandlung dieses Falles eingesetzt (hier: $TSK_{RR \& ULS}$), so reduziert sich der Modellierungsfehler abermals deutlich. Die Unterschiede zwischen J_{MVB} und J_{CVB} bzw. zwischen TSK_{RR} und TSK_{LSA} sind in diesem Fall nur minimal, daher werden diese Ergebnisse in Tabelle 2 nicht aufgeführt.

J	Regeln	Typ	Lerndaten		Validaten		Gesamtdaten	
			MAE	MSE	MAE	MSE	MAE	MSE
J_{RI}	102	Mamdani	5.672	51.71	5.673	54.07	5.673	52.89
J_{MVB}	214	Mamdani	3.745	30.86	4.348	37.74	4.047	34.30
J_{MVB}	214	TSK_{RR}	2.304	13.31	3.053	21.93	2.679	17.62
J_{MVB}	214	$TSK_{RR \& ULS}$	1.796	7.53	2.574	15.94	2.185	11.74

Tabelle 2: Modellierungsergebnisse für *Boston Housing Problem*

Die vorhandenen Daten wurden für eine weitere Untersuchung im Verhältnis von $D_{Lern} = 20$ zu $D_{Vali} = 486$ getrennt, um die Modellierung bei einer kleinen Datenbasis und schwachen Zusammenhängen zu simulieren. Die Lerndaten wurden dabei so ausgewählt, dass sie das Spektrum der auftretenden Grundstückspreise stichpunktartig abdecken. Auf Grund des geringen Umfangs der Lerndaten wurde für alle Teststrategien die zulässige Irrtumswahrscheinlichkeit zu $\alpha = 0.1$ gewählt.

J	Regeln	Typ	Lerndaten		Validaten		Gesamtdaten	
			MAE	MSE	MAE	MSE	MAE	MSE
J_{RI}	0	Mamdani	8.769	108.00	8.666	105.30	8.769	108.00
J_{MVB}	59	Mamdani	5.642	45.58	7.205	77.09	7.144	75.85
J_{CVB}	46	Mamdani	4.221	24.47	6.867	78.57	6.763	76.43

Tabelle 3: Ergebnisse für *Boston Housing* für $D_{Lern} = 20$ und $D_{Vali} = 486$

Mit Hilfe des Relevanzindex J_{RI} wird in diesem Fall keine Regel gefunden. Das beste Ergebnis wird mit dem Medianwertbasierten Test J_{CVB} erzielt. Die Regelbasis ist im Vergleich zu dem Mittelwertbasierten Test J_{MVB} um ca. 20% kleiner, trotzdem gelingt die Modellierung deutlich besser. Dies ist ein Indiz dafür, dass der Mittelwertbasierte Tests in diesem Fall auch Regeln generiert, die keinen relevanten Zusammenhang beschreiben und so die Modellierung eher verschlechtern.

6 Zusammenfassung und Ausblick

In diesem Beitrag sind zwei neue Strategieelemente für das Fuzzy-ROSA-Verfahren vorgestellt worden, die zum einen die Regelgenerierung und zum anderen die Regeltransformation auf TSK-Regeln bei kleiner und/oder gestörter Datenbasis verbessern. Dies ist deshalb von Bedeutung, da Anwender zunehmend an einer datenbasierten Modellierung auch bei dürftigen Datenbasen interessiert sind.

Wenn nur schwache Zusammenhänge in den Daten vorhanden sind, liefert die Regelsuche mit Test- und Bewertungsverfahren auf Basis der relativen Häufigkeiten oftmals kein befriedigendes Ergebnis. Abhilfe wird teilweise durch den Mittelwertbasierten Regelttest geschaffen. Durch die Anfälligkeit gegenüber Ausreißern können aber relevante Zusammenhänge verdeckt bzw. vorgetäuscht werden. Zur Überwindung dieser Probleme wird hier ein Test- und Bewertungsverfahren vorgestellt, das auf dem Median basiert, dessen Vorteil in der größeren Robustheit gegenüber Ausreißern liegt. Ferner macht der hierzu verwendete statistische Test keine spezielle Annahme über die Verteilung der Daten in den zu untersuchenden Stichproben.

Durch die Transformation von Mamdani- in TSK-Regeln kann die Modellgenauigkeit oftmals erheblich verbessert werden. Einzelne Ausreißer können sich — wie bei der Regelgenerierung — negativ auf das Resultat auswirken. Um auch für diesen Modellierungsschritt eine höhere Robustheit zu erzielen, wird eine Approximation auf der Basis der bekannten robusten Regression eingesetzt. Bei einer kleinen Datenbasis sind oftmals nicht ausreichend viele linear unabhängige Daten zu einer Regel für die angestrebte Transformation vorhanden, so dass auf eine Ersatzstrategie zurückgegriffen werden muss. Diese besteht derzeit darin, einfach den Mittelwert der vorhandenen Daten zu verwenden. Die Modellgenauigkeit kann gegenüber dem Mamdani-System so aber nicht signifikant erhöht werden. Zur Verbesserung ist eine aufwendigere, aber im Ergebnis günstigere Strategie in das Fuzzy-ROSA-Verfahren implementiert worden. Die Regeltransformation lässt sich so auch bei einer kleinen Datenbasis erfolgreich anwenden.

Die Leistungsfähigkeit der neuen Strategieelemente wird anhand von Beispielen gezeigt. In der weiteren Arbeit sollen andere Funktionsansätze, speziell nicht lineare, für die Transformation in TSK-Regeln untersucht werden. Dabei ist langfristig auch daran gedacht, anhand der Charakteristik der zugrunde liegenden Daten den am besten geeigneten Funktionsansatz, evtl. sogar für jede Regel individuell, automatisch auszuwählen. Ein weiteres Ziel besteht darin, die Transformation in TSK-Regeln auch auf negative Regeln auszudehnen.

Danksagung

Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft im Rahmen des Transferbereiches "*Datenbasierte Generierung und Optimierung von Modellen mit Methoden der Computational Intelligence*" (37) gefördert.

Literatur

- [1] Mamdani, E. H. ; Gaines, B. R.: *Fuzzy Reasoning and its Applications*. London : Academic Press. 1981.
- [2] Takagi, T ; Sugeno, M.: Fuzzy Identification of Systems and its Application to Modeling and Control. In: *IEEE Transactions on Systems, Man, and Cybernetics* 15 (1985), Nr. 1, S. 116–132.

- [3] Kiendl, H. ; Krabs, M.: Ein Verfahren zur Generierung regelbasierter Modelle für dynamische Systeme. In: *at - Automatisierungstechnik* 37 (1989), Nr. 11, S. 423–430.
- [4] Kiendl, H.: *Fuzzy Control methodenorientiert*. München : Oldenbourg 1997.
- [5] Slawinski, T.: *Analyse und Generierung von relevanten Fuzzy-Regeln in hochdimensionalen Suchräumen*. Düsseldorf : VDI, 2001 (Fortschritt-Berichte VDI, Reihe 10, Nr. 686)
- [6] Krone, A.: *Datenbasierte Generierung von relevanten Fuzzy-Regeln zur Modellierung von Prozesszusammenhängen und Bedienstrategien*. Düsseldorf : VDI, 1999 (Fortschritt-Berichte VDI, Reihe 10, Nr. 615)
- [7] Jessen, H.: *Test- und Bewertungsverfahren zur regelbasierten Modellierung und Anwendung in der Lastprognose*. Düsseldorf : VDI, 2000 (Fortschritt-Berichte VDI, Reihe 8, Nr. 836)
- [8] Krause, P.: Generierung von Takagi-Sugeno-Fuzzy-Systemen aus relevanten Fuzzy-Regeln. In: *Proceedings 10. Workshop Fuzzy-Control des GMA-FA 5.22* (Mikut, R.; Jäkel, J., Hg.), S. 84–97. Karlsruhe : Wissenschaftliche Berichte Forschungszentrum Karlsruhe, FZKA 6509.
- [9] Krause, P.: *Datenbasierte Generierung von transparenten und genauen Fuzzy-Modellen für mehrdeutige Daten und komplexe Systeme*. Düsseldorf : VDI, 2001 (Fortschritt-Berichte VDI, Reihe 10, Nr. 691)
- [10] Wilcoxon, F.: Individual Comparisons by Ranking Methods. In: *Biometrics* 1 (1945), S. 80–83.
- [11] DuMouchel, W. H. ; O'Brien, F. L. Integrating a robust option into a multiple regression computing environment. In: *Computer Science and Statistics: Proceedings of the 21st Symposium on the Interface, American Statistical Association, Alexandria, VA (1989)*, S. 297–301.
- [12] Holland, P. W. ; Welsch, R. E. Robust regression using iteratively reweighted least-squares. In: *Communications in Statistics - Theory and Methods*, v. A6 (1977), S. 813–827.
- [13] Goodall, C. R.: Computation using the QR decomposition. Handbook in Statistics, Volume 9. Statistical Computing, Amsterdam, NL, 1993.
- [14] Anderson, E. ; Bai, Z. ; Bischof, C. ; Blackford, S. ; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A. ; Hammarling, S. ; McKenney, A. ; Sorensen, D.: *LAPACK User's Guide*, Third Edition, Society for Industrial and Applied Mathematics, Philadelphia, USA, 1999.
- [15] Mackey, M. ; Glass, L. Oscillation and Chaos in Physiological Control Systems. In: *Science* 197 (1977), S. 287–289.
- [16] Harrison, D. ; Rubinfeld, D. L.: *Hedonic Proces and the Demand for Clean Air*. In: *Economics & Management* 5 (1978), S. 81–102.

Benchmarkproblem: Takagi-Sugeno Fuzzy Modell zur Bestimmung von Parametern in einem Druckbehälter

A. Seeliger , M. Wagenknecht, R. Hampel

Institut für Prozeßtechnik, Prozeßautomatisierung und Meßtechnik (IPM),
Hochschule Zittau / Görlitz (FH),
Theodor-Körner-Allee 16,
02763 Zittau

Abstract:

Die Auswertung einer Reihe von gemessenen Daten an einem Sensorsystem, bestehend aus einer Meßblanze, soll Rückschlüsse auf die in einem Druckbehälter vorherrschenden Parameter ermöglichen. Für die Modellierung wurde der Einsatz von Soft Computing-Methoden in Form eines Takagi-Sugeno Fuzzy Modells präferiert.

Für die Identifikation auf der Grundlage einer umfassenden Datenbasis kamen unterschiedliche Clusterverfahren zur Anwendung, z.B.:

- k-means
- Single Linkage
- Fuzzy c-means
- k-medoids

Das Takagi-Sugeno Model selbst besteht aus einer Basis von Regeln der Form

$$\text{If } x \text{ is } A_m \text{ then } y = b_{m0} + \sum_{i=1}^p b_{mi} x(i).$$

Die Verifikationen ergaben, das letztendlich mit keinem der erstellten TS-Modelle verlässliche Daten generiert werden konnten. Einige der resultierenden Parameter überschreiten den gegebenen Wertebereich um ein Vielfaches.

Die Problemlösung wurde folgendermaßen gestaltet: Es wird nun alternativ auf zwei klassische Methoden zurückgegriffen, welche zur Laufzeit fortwährend von dem Bestand experimenteller Daten Gebrauch machen. Die gesuchten Parameter werden hierbei durch Interpolation mittels inverser Distanzwichtung oder Unterraumapproximation ermittelt.

Am Ende steht die Fragestellung hinsichtlich der Verwendung von Takagi-Sugeno-Modellen: Was ist ausschlaggebend für eine erfolgreiche Modellierung? Besteht lediglich eine Abhängigkeit vom Clusterverfahren, welches für die Modellidentifikation zum Einsatz kommt? Welche Clusteralgorithmen könnten für den vorgestellten Datensatz geeignet sein?

Konzepte zur entscheidungstheoretischen Bewertung von Fuzzy-Systemen

Sebastian Beck, Ralf Mikut

Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik,
D-76021 Karlsruhe, Postfach 3640,
Telefon: (07247) 82-5731, Fax: (07247) 82-5702, E-Mail: ralf.mikut@iai.fzk.de

Zusammenfassung: Bei Entscheidungsproblemen mit unterschiedlichen Konsequenzen der möglichen Fehlentscheidungen ist das Minimieren der Anzahl der Fehlentscheidungen nicht unbedingt die beste Strategie. Für einige Verfahren zum Entwurf von Entscheidungssystemen sind Bewertungsmaße bekannt, die unterschiedliche Konsequenzen der Fehlentscheidungen berücksichtigen. Für statistische Entscheidungssysteme wird zum Beispiel das Risiko einer Entscheidung minimiert. Dieses entscheidungstheoretische Konzept wird auf die Bewertung von Fuzzy-Systemen übertragen. Es wird dabei auf die Besonderheiten bei der Bewertung von einzelnen Regeln und vollständigen Regelbasen eingegangen. Für den Benchmarkdatensatz Kfz-Aggregate werden Ergebnisse vorgestellt.

1 Einleitung

Der Einsatzbereich von Systemen, die automatisch Entscheidungen treffen oder Handlungsempfehlungen geben, erstreckt sich unter anderem auf technische, medizinische und betriebswirtschaftliche Probleme. Zum ersten Bereich gehören beispielsweise Probleme der Fehlerdiagnose von Prozessen [1, 2, 3] oder die Qualitätskontrolle von Kfz-Aggregaten [4, 5]. Zu den betriebswirtschaftlichen Problemen zählen die Investitionsentscheidungen eines Unternehmens oder die Bewertung des Kunden einer Bank, der einen Kredit beantragt. Aber auch im Bereich der Medizin befinden sich Systeme zur Entscheidungsunterstützung in der Entwicklung [6, 7].

Die präskriptive Entscheidungstheorie [8, 9, 10, 11] bietet eine Basis für die Bewertung eines Entscheidungssystems nach den Konsequenzen der getroffenen Entscheidungen. Das Grundmodell der Entscheidung besteht aus den Handlungsalternativen (Entscheidungen) des Entscheidungsträgers e_k , den möglichen Umweltzuständen z_j und den Ergebnissen (meistens Kosten) für die Kombinationen aus tatsächlichem Umweltzustand und getroffener Entscheidung $L(e_k, z_j)$. Dieses Grundmodell kann durch verschiedene Modifikationen (Beobachtungen, Unschärfen, Kostenarten) erweitert werden. Eine entscheidungstheoretische Bewertung ist besonders dann vorteilhaft, wenn aus mehreren Kandidaten das Entscheidungssystem mit den geringsten Gesamtkosten ausgesucht werden soll.

In Abhängigkeit vom Umfang der vorhandenen Informationen bezüglich des tatsächlichen Umweltzustandes z_j können Entscheidungsprobleme eingeordnet werden (Bild 1). Bei vollständiger Information - der tatsächliche Umweltzustand ist bekannt - geht das Entscheidungsproblem in ein Optimierungsproblem - die Suche nach der besten Entscheidung für den bekannten Umweltzustand - über. Bei unbekanntem Umweltzustand wird zusätzlich unterschieden, ob eine Wahrscheinlichkeitsverteilung der Umweltzustände vorliegt (Risiko) oder nicht (Ungewissheit).

Die zur Entscheidung verwendete Wahrscheinlichkeitsverteilung der Umweltzustände kann zum Beispiel aus Merkmalen zur Prozesssituation geschätzt werden. Dieses zusätzliche Wissen über den tatsächlichen Umweltzustand wird als Beobachtung bezeichnet und stellt eine Erweiterung des Grundmodells dar.

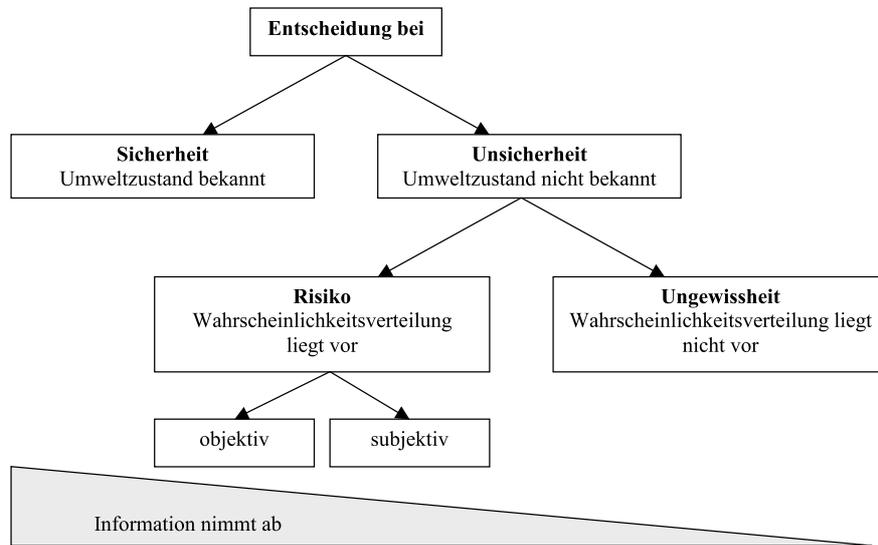


Bild 1: Entscheidungsprobleme bei Sicherheit, Unsicherheit, Ungewissheit und Risiko (erweiterte Darstellung nach [9]).

In dieser Arbeit werden die Beobachtungen durch Fuzzy-Regeln dargestellt. Dazu werden Gebiete des Raumes der Merkmale zusammengefasst (Prämissen der Regeln) und die optimalen Entscheidungen für diese Gebiete bestimmt (Konklusionen). Das Ziel besteht darin, die Situation der Unsicherheit möglichst in eine Situation bei Sicherheit zu überführen. Dabei sind die Einzelregeln mit Prämissen und Konklusionen so zu gestalten und zu einer Regelbasis zusammenzustellen, dass sie die erwarteten Kosten pro Entscheidung minimieren. In praktischen Fällen ist vollständige Information und damit Entscheidung bei Sicherheit meist nicht erreichbar. Es resultiert dann eine Risikosituation mit subjektiven Wahrscheinlichkeiten.

Zum Entwurf der Entscheidungssysteme steht ein Lerndatensatz mit N Beispielen, N_x Merkmalen $\mathbf{x}[n] = (x_1[n] \cdots x_l[n] \cdots x_{N_x}[n])$ und einer bestimmten Ausgangsvariablen y mit N_z unterschiedlichen Umweltzuständen z_j zur Verfügung.

Im Einzelnen wird auf

- die entscheidungstheoretische Bewertung von Fuzzy-Regeln und Fuzzy-Regelbasen (Abschnitt 2),
- entscheidungstheoretische Entwurfsverfahren für Fuzzy-Regeln und Fuzzy-Regelbasen (Abschnitt 3) sowie
- den KFZ-Benchmarkdatensatz zur Demonstration von Ergebnissen (Abschnitt 4)

eingegangen.

2 Entscheidungstheoretische Bewertung von Fuzzy-Systemen

2.1 Entscheidungstheorie

Die Gesamtkosten für ein Entscheidungssystem L_T bestehen aus Kosten für (Fehl-) Entscheidungen L_D und Kosten L_C , die zum Treffen dieser Entscheidung anfallen. Letztere setzen sich hauptsächlich aus Kosten $L_{C,l}$ für die Auswertung von verwendeten Merkmalen x_l zusammen. Eine detaillierte Auflistung verschiedener Kostenarten bei Entscheidungssystemen ist in [12] zu finden.

Die Bewertung der Kombinationen aus Entscheidungen und Umweltzuständen werden in der Kostenmatrix \mathbf{L} zusammengefasst. In den Zeilen der Matrix stehen die einzelnen Entscheidungen, in den Spalten die Umweltzustände:

$$\mathbf{L} = \begin{pmatrix} L(e_1, z_1) & \cdots & L(e_1, z_j) & \cdots & L(e_1, z_{N_z}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ L(e_k, z_1) & \cdots & L(e_k, z_j) & \cdots & L(e_k, z_{N_z}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ L(e_{N_e}, z_1) & \cdots & L(e_{N_e}, z_j) & \cdots & L(e_{N_e}, z_{N_z}) \end{pmatrix}. \quad (1)$$

Für Klassifikationsprobleme mit richtigen und falschen Entscheidungen werden meist Kostenmatrizen mit Hauptdiagonalelementen gleich Null verwendet, zum Beispiel:

$$\mathbf{L} = \begin{pmatrix} 0 & 1 \\ L_R & 0 \end{pmatrix} \quad (2)$$

für $N_e = 2$ Entscheidungen und $N_z = 2$ Umweltzustände. Mittels dieser Matrix lassen sich über das Kostenverhältnis L_R die Beziehungen der Kosten der Fehlentscheidungen untereinander abbilden. Mit der Verteilung $\hat{p}(e_k \wedge z_j)$ der Kombinationen aus Entscheidungen e_k und Umweltzuständen z_j lässt sich ein Erwartungswert der Kosten der Entscheidungen bestimmen [17]:

$$\hat{L}_D = \sum_{k=1}^{N_e} \sum_{j=1}^{N_z} L(e_k, z_j) \cdot \hat{p}(e_k \wedge z_j). \quad (3)$$

Die beste Entscheidung für eine gegebene Beobachtung b_i und bei Kenntnis der Verteilung der Zustände ergibt sich dann zu:

$$e_{opt}(b_i) = \arg \min_{e_k} \underbrace{\sum_{j=1}^{N_z} L(e_k, z_j) \cdot \hat{p}(z_j | b_i)}_{\hat{L}_D(e_k | b_i)}. \quad (4)$$

Mögliche Informationsdefizite unterteilen sich in (statistische) Unsicherheiten, unscharfe bzw. vage Aussagen im Sinne gradueller Zuordnungen zu einer Aussage sowie impräzise Aussagen (z.B. durch Intervalle von Werten). Dabei werden hier die Begriffe in Anlehnung an [13] verwendet¹. Das Grundmodell der Entscheidungstheorie betrachtet nur statistische Unsicherheiten.

¹In der Literatur existiert hier leider keine konsistente und anerkannte Begriffsbildung, insbesondere die Begriffe Unsicherheit und Ungewissheit (engl. uncertainty) werden mit unterschiedlichen Bedeutungen verwendet. Eine Reihe von Autoren unterscheidet eine Vielzahl weiterer Arten von Informationsdefiziten, vgl. z.B. [14, 15]. Die Autoren dieses Beitrags danken Lutz Gröll und Jens Jäkel für ihre Unterstützung bei der Analyse unterschiedlicher Begriffswelten.

Eine zusätzliche Unschärfe bzw. Vagheit kann in einem oder mehreren Elementen des Entscheidungsmodells auftreten:

1. Die Beobachtungen b_i sind graduell zugeordnet: μ_{b_i} .
2. Die Entscheidungen e_k sind graduell zugeordnet: μ_{e_k} .
3. Die Umweltzustände z_j sind graduell zugeordnet: μ_{z_j} (z.B. bei Regressionsaufgaben).
4. Die Entscheidungskosten $L(e_k, z_j)$ sind graduell zugeordnet: $\mu_{L(e_k, z_j)}$ (siehe z.B. [16]).
5. Die Merkmalskosten $L_{C,l}$ sind graduell zugeordnet: $\mu_{L_{C,l}}$.

Bei graduellen Zuordnungen der Entscheidungen und bzw. oder Umweltzuständen muss auch eine geeignete Berechnungsvorschrift für deren Behandlung in (1) bereitgestellt werden.

Beispiele für impräzise Elemente sind:

1. Die Entscheidungskosten $L(e_k, z_j)$ umfassen ein Intervall von reellen Werten: $L_{min} \leq L(e_k, z_j) \leq L_{max}$.
2. Die Merkmalskosten $L_{C,l}$ umfassen ein Intervall von reellen Werten: $L_{min} \leq L_{C,l} \leq L_{max}$.

In dieser Arbeit werden hauptsächlich Unschärfen vom Typ 1 (Beobachtungen sind Prämissen von Fuzzy-Regeln) verwendet. Entscheidungen, Umweltzustände und Kosten werden hingegen scharf formuliert. Innerhalb des Fuzzy-Systems und während des Entwurfsprozesses treten allerdings weitere Unschärfen auf: Mögliche Entscheidungen sind während der Auswertung der Inferenz unscharf.

Im Entwurfsprozess kommt es zu impräzisen Entscheidungs- und Merkmalskosten für Einzelregeln und unvollständige Regelbasen.

2.2 Fuzzy-System

Die zu entwerfenden Fuzzy-Systeme bestehen aus Regeln R_r der Struktur:

$$R_r : \text{WENN } \underbrace{x_1 = A_{1,R_r}}_{\text{Teilprämisse } P_{r,1}} \text{ UND } \cdots \text{ UND } \underbrace{x_{N_x} = A_{N_x,R_r}}_{\text{Teilprämisse } P_{r,N_x}} \text{ DANN}$$

$$\underbrace{\hspace{15em}}_{\text{Prämisse } P_r}$$

$$e = C_r, \quad r = 1, \dots, N_r,$$

und einer Default-Regel $R_{N_r+1} : \text{SONST } e = C_{N_r+1}$.

Die Prämisse P_r besteht aus 'UND'-Verknüpfungen von Teilprämissen $P_{r,1}, \dots, P_{r,N_x}$. Die linguistischen Terme A_{l,R_r} können einzelne linguistische Terme $A_{l,i} (i = 1, \dots, m_l)$ des Merkmals x_l sowie 'ODER'-Verknüpfungen der benachbarten Terme $A_{l,i_{min}}$ bis $A_{l,i_{max}}$ oder aller linguistischer Terme des Merkmals x_l sein. Dabei ist m_l die Anzahl linguistischer Terme des Merkmals x_l . Bild 2 zeigt ein Beispiel für die Zugehörigkeitsfunktionen eines Merkmals x_l . Die m_l Parameter $a_{l,i}$ werden so initialisiert, dass die Intervalle $a_{l,j} < x_l < a_{l,j+1}$ stets ungefähr die selbe Anzahl an Beispielen von ca. $N/(m_l - 1)$ beinhalten und möglichst gerundete

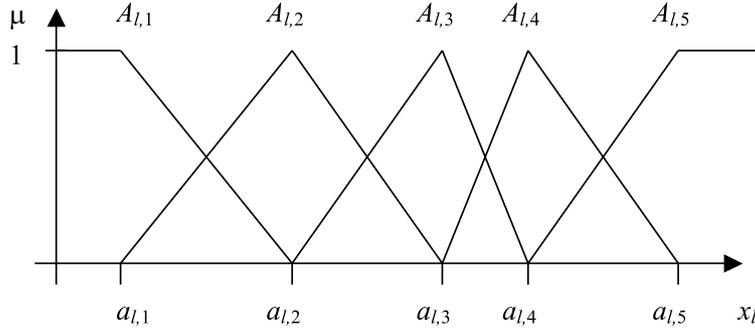


Bild 2: Beispiel: Zugehörigkeitsfunktionen des Merkmals x_l (fünf Terme).

Werte aufweisen [18]. Nicht alle Merkmale x_l müssen in den Teilprämissen einer Regel vertreten sein.

Die Wahrscheinlichkeit $p(A_{l,i})$ der unscharfen Menge $A_{l,i}$ ergibt sich bei bekannter Verteilung des Merkmals $f_{x_l}(x_l)$ zu [19, 20]:

$$p(A_{l,i}) = \int_{-\infty}^{\infty} \mu_{A_{l,i}}(x_l) f_{x_l}(x_l) dx_l. \quad (5)$$

Die tatsächliche Verteilung f_{x_l} ist allerdings unbekannt. Die Wahrscheinlichkeit $p(A_{l,i})$ wird durch Aufsummieren der Zugehörigkeiten aller Elemente des Lern Datensatzes zu $A_{l,i}$ bestimmt:

$$\hat{p}(A_{l,i}) = \frac{1}{N} \sum_{n=1}^N \mu_{A_{l,i}}(x_l[n]). \quad (6)$$

Besteht die Teilprämisse $P_{r,l}$ aus ODER-Verknüpfungen benachbarter Terme $i_{min} \dots i_{max}$ eines Merkmals x_l , wird als ODER-Operator die beschränkte Summe verwendet:

$$\mu_{P_{r,l}}(x_l) = \min\left(1, \sum_{i=i_{min}}^{i_{max}} \mu_{A_{l,i}}(x_l)\right). \quad (7)$$

Für die UND-Verknüpfung von einzelnen Teilprämissen $\mu_{P_{r,l}}$ verschiedener Merkmale wird als UND-Operator das Produkt verwendet:

$$\mu_{P_r}(\mathbf{x}) = \prod_{l=1}^{N_x} \mu_{P_{r,l}}(\mathbf{x}). \quad (8)$$

Über (7) und (8) können die Zugehörigkeiten der Beispiele zu beliebigen Prämissen bestimmt werden. Für die Berechnung von \hat{p}_{P_r} und \hat{p}_{z_j} sind in (6) die Zugehörigkeiten zu einzelnen Termen $\mu_{A_{l,i}}$ durch die Zugehörigkeiten $\mu_{P_r}(\mathbf{x}[n])$ bzw. $\mu_{z_j}(y[n])$ zu ersetzen. Für $\hat{p}_{z_j|P_r}$ gilt

$$\hat{p}(z_j|P_r) = \frac{\hat{p}(z_j \wedge P_r)}{\hat{p}(P_r)} = \frac{\sum_{n=1}^N \mu_{P_r}(x_l[n]) \cdot \mu_{z_j}(y[n])}{\sum_{n=1}^N \mu_{P_r}(x_l[n])}, \mu_{z_j}(y[n]) \in \{0; 1\}. \quad (9)$$

Der UND-Operator im Zähler kann durch ein einfaches Produkt ersetzt werden, da $\mu_{z_j}(y[n])$ in dieser Arbeit nur die Werte $\{0; 1\}$ annehmen kann (keine Unschärfe der Umweltzustände).

Bei diesem Fuzzy-System ergeben sich zwei mögliche Bedeutungen der Konklusionen C_r der Regeln:

- **Zustandsschätzung:** Die Konklusion der Regel entspricht einer Schätzung der Wahrscheinlichkeiten der Umweltzustände. Aus den Regeln sind die Wahrscheinlichkeiten der Umweltzustände $\hat{p}(z_j|\mathbf{x})$ in Abhängigkeit von den Eingangsgrößen \mathbf{x} zu schätzen. Die beste Entscheidung wird anschließend mit (4) über die erwarteten Kosten bestimmt. Dieser Weg entspricht dem Vorgehen bei der Bayes-Klassifikation.
- **Entscheidungsschätzung:** Die Konklusion ist die jeweils beste Entscheidung für die Beispiele im Gebiet der Prämisse nach (4). Über alle für \mathbf{x} aktivierten Regeln ($\mu_{P_r}(\mathbf{x}) > 0$) der Regelbasis werden Zugehörigkeiten zu den Entscheidungen $\mu_{e_k}(\mathbf{x})$ bestimmt und über ein Inferenzverfahren aggregiert. Die Entscheidung mit der maximalen Zugehörigkeit wird getroffen.

Bild 3 zeigt diese beiden Ansätze an einem einfachen Beispiel mit zwei Umweltzuständen z_1 und z_2 , deren Verteilung im unteren Teil dargestellt ist. Im mittleren Teil sind die Aktivierungen zweier Prämissen über dem Merkmal x aufgetragen. Die Matrix der bedingten Wahrscheinlichkeiten der beiden betrachteten Umweltzustände ergibt sich für die dargestellten Daten zu:

$$\hat{\mathbf{p}}(z_j|P_r) = \begin{pmatrix} 0.85 & 0.15 \\ 0.15 & 0.85 \end{pmatrix}. \quad (10)$$

Aus dem Verlauf der Zugehörigkeitsfunktionen mit den Knickpunkten [1, 4] ergibt sich über die beschränkte Summe als ODER-Operator die Schätzung der Wahrscheinlichkeiten der Zustände für ein gegebenes x :

$$\hat{\mathbf{p}}(z_j|x) = \begin{pmatrix} 0.85 \cdot \mu_{P_1}(x) + 0.15 \cdot \mu_{P_2}(x) \\ 0.15 \cdot \mu_{P_1}(x) + 0.85 \cdot \mu_{P_2}(x) \end{pmatrix}. \quad (11)$$

Der obere Teil von Bild 3 zeigt den Verlauf der Positionen, an denen für variable L_R der Kostenmatrix (2) die Entscheidung von e_1 nach e_2 kippt. Die gepunktete Linie entspricht einer Entscheidung auf der Basis der Schätzung der Wahrscheinlichkeit der Umweltzustände für die Werte von x in Kombination mit der Kostenmatrix (4). Die Entscheidung mit dem geringeren Risiko wird getroffen. Allerdings sind Kostenverhältnisse möglich, bei denen stets die selbe Entscheidung getroffen wird. Die gekreuzte Linie entspricht einem α -cut bei 0.5, dadurch geht die Matrix der bedingten Wahrscheinlichkeiten der beiden betrachteten Umweltzustände über in:

$$\hat{\mathbf{p}}(z_j|P_r) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (12)$$

Für die durchgezogene Linie werden den Prämissen die Entscheidungen e_1 beziehungsweise e_2 zugeordnet. Die Entscheidung wird auf der Basis der Aktivierung der Prämissen getroffen und ist damit unabhängig von L_R . An diesem Beispiel wird deutlich, dass die Abhängigkeit der Entscheidung von den Kosten bei Zustandsschätzern zu widersprüchlichen Ergebnissen führen kann, obwohl das spezielle Problem

fehlerfrei lösbar ist. Daher wird das Fuzzy-System in dieser Arbeit nicht als Zustandsschätzer, sondern als Entscheidungsschätzer betrachtet. Die Abhängigkeit von der Kostenmatrix wird über das Design der einzelnen Regeln erreicht.

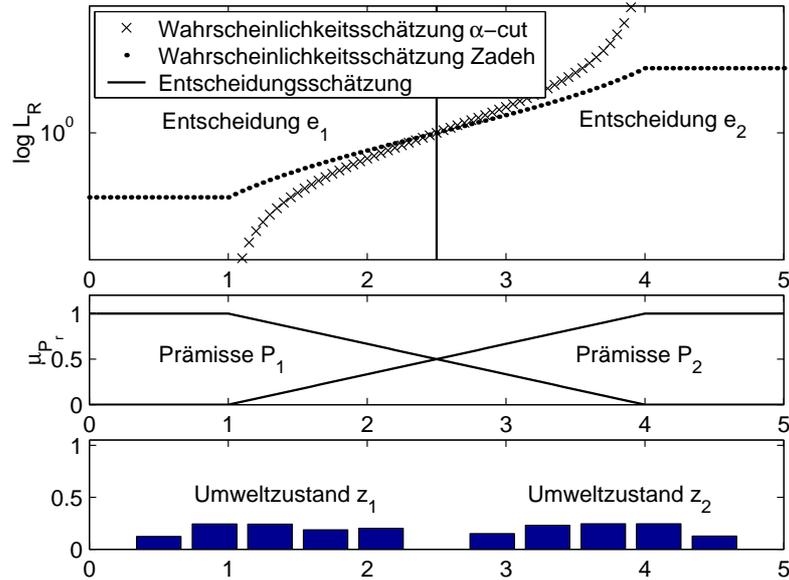


Bild 3: Entscheidungen in Abhängigkeit vom Kostenverhältnis L_R und dem Merkmal x für die Zustandsschätzung mit zwei verschiedenen Berechnungen für die Wahrscheinlichkeit und für die Entscheidungsschätzung.

Für die Bestimmung der Verbundwahrscheinlichkeiten der einzelnen Kombinationen aus Entscheidungen und Umweltzuständen $\hat{p}(e_k \wedge z_j)$ wird das in [21] veröffentlichte Inferenzverfahren verwendet. Dieses trifft auch bei überlappenden ($\sum_r \mu_{P_r} > 1$) oder widersprüchlichen Prämissen eindeutige Entscheidungen.

2.3 Bewertung von Fuzzy-Einzelregeln

Die Prämisse einer Fuzzy-Einzelregel deckt nicht den gesamten Wertebereich der Eingangsgrößen ab. Sie teilt den Lerndatensatz in zwei Teile mit graduellen Zugehörigkeiten: die Beispiele in der Prämisse und die restlichen Beispiele. Für die Beispiele in der Prämisse liegt eine Risikosituation vor, da die Wahrscheinlichkeiten der Umweltzustände nach (9) bestimmt sind. Für die restlichen Beispiele können die Wahrscheinlichkeiten der Umweltzustände ebenfalls über (9) bestimmt werden. Jedoch ist während des Prunings einer Prämisse ungewiss, welche weiteren Prämissen die restlichen Beispiele teilweise oder vollständig erfassen.

Da eine Regelbasis aus möglichst wenig Regeln bestehen soll, sind die Prämissen der einzelnen Regeln möglichst weit zu generalisieren. Eine optimale Einzelregel erfasst alle Beispiele, für die die Entscheidung $e(P_r)$ die beste ist. Alle Beispiele, für die $e(P_r)$ nicht die beste Entscheidung ist, sollen von der Prämisse nicht erfasst werden. Bei Problemen mit nicht kompakten Verteilungen eines oder mehrerer Umweltzustände sind mehrere Regeln mit der selben Konklusion in die Regelbasis aufzunehmen, um eine hohe Anzahl Fehlentscheidungen zu vermeiden. Allerdings ist bei der Bewertung einer Einzelregel noch nicht bekannt, ob in der Regelbasis weitere Regeln mit gleicher

Entscheidung existieren. Der Kompromiss zwischen der Abdeckung aller Beispiele mit gleicher Entscheidung und der Vermeidung von Fehlentscheidungen ist über das verwendete Bewertungsmaß zu formalisieren.

In die Bewertung gehen alle Beispiele der Prämisse über die Kosten L_{D,P_r} ein. Damit ist sichergestellt, dass Fehlentscheidungen für Beispiele in der Prämisse zu höheren Kosten führen. Die Kosten L_Z bewerten diejenigen Beispiele im Gebiet der Restprämisse, für die $e(P_r)$ die beste Entscheidung ist und für die somit eine andere Regel eine Fehlentscheidung treffen kann. Diese Zusatzkosten für nicht erfasste Beispiele von z_{C_r} stellen potentielle Kosten dar. Sie bestrafen Regeln mit zu kleinen und dafür fehlerfreien Prämissen. Werden nur die Kosten der Beispiele in der Prämisse betrachtet, werden kleine fehlerfreie Prämissen am besten bewertet. Werden alle Beispiele in der Bewertung berücksichtigt (Risikosituation für alle Beispiele mit einer Entscheidung für die Restprämisse), führt dies insbesondere bei Problemen mit mehr als zwei Zuständen zu nicht plausiblen Ergebnissen. Ein Beispiel dafür sind Regeln, bei denen vergleichsweise billige Fehlentscheidungen im Prämissegebiet in Kauf genommen werden, um teurere Fehlentscheidungen im Gebiet der Restprämisse zu vermeiden. Die Fehlentscheidungen im Prämissegebiet sind allerdings durch weitere Regeln kaum noch zu kompensieren, was für die Regelbasis tendenziell zu höheren Kosten führt.

Die Kosten der Prämisse ergeben sich zu:

$$L_{D,P_r} = \sum_{j=1}^{N_z} L(e(P_r), z_j) \cdot \hat{p}(z_j | P_r). \quad (13)$$

Der Anteil Beispiele von z_{C_r} in der negierten Prämisse ist $\hat{p}(z_{C_r} \wedge \bar{P}_r)$. Ob für diese Beispiele eine weitere Regel eine Fehlentscheidung treffen kann, hängt von der Anzahl Beispiele der übrigen Umweltzustände innerhalb \bar{P}_r ab. Im Folgenden wird eine obere Abschätzung der so entstehenden Kosten vorgestellt.

Zur Untersuchung dieses Falles wird geprüft, welche Bedingungen zur Existenz solcher Regeln zu erfüllen sind. Dazu wird eine überlappungsfreie Aufteilung des Gebietes der negierten Prämisse in mindestens eine Prämisse P_s mit $e(P_s) \neq e(P_r)$ angenommen. Da die Konklusion für eine Prämisse nach (4) bestimmt wird, muss für alle P_s gelten:

$$\sum_{i=1}^{N_z} L(e(P_s), z_i) \cdot \hat{p}(z_i | P_s) \leq \sum_{i=1}^{N_z} L(e(P_r), z_i) \cdot \hat{p}(z_i | P_s). \quad (14)$$

Existiert nur eine Prämisse P_s , ergibt sich aus (14) nach Multiplikation mit $\hat{p}(\bar{P}_r)$ die Bedingung für die maximale Verbundwahrscheinlichkeit für Beispiele von z_{C_r} in der negierten Prämisse $\bar{P}_r = P_s$, so dass $e(P_s) \neq e(P_r)$ gilt:

$$\hat{p}(z_{C_r} \wedge \bar{P}_r) \leq \frac{\sum_{z_j=\{z_1, \dots, z_{C_r-1}, z_{C_r+1}, \dots, z_{N_z}\}} [L(e(\bar{P}_r), z_j) - L(e(P_r), z_j)] \cdot \hat{p}(z_j \wedge \bar{P}_r)}{\underbrace{L(e(P_r), z_{C_r}) - L(e(\bar{P}_r), z_{C_r})}_{\hat{p}_{max}(z_{C_r} \wedge \bar{P}_r)}}. \quad (15)$$

Diese Gleichung bedeutet aber, dass es eine obere Schranke $\hat{p}_{max}(z_{C_r} \wedge \bar{P}_r)$ für den Anteil der Beispiele von z_{C_r} gibt, für die eine Fehlentscheidung getroffen werden kann. Bild 4 zeigt die möglichen Aufteilungen von \bar{P}_r schematisch.

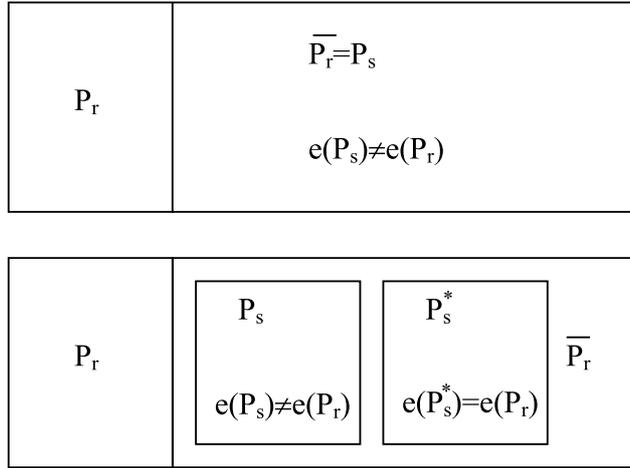


Bild 4: Aufteilungen der Restprämisse. Oben: vollständiges Gebiet der negierten Prämisse kann unterschiedliche Konklusion haben. Unten: mindestens eine Prämisse P_s^* mit $e(P_s^*) = e(P_r)$ existiert.

Es werden nun die maximal möglichen erwarteten Kosten je Beispiel von z_{C_r} für den Fall bestimmt, dass eine andere Regel R_s für diese Beispiele eine Fehlentscheidung trifft. Diese Kosten ergeben sich, wenn für alle Beispiele von z_{C_r} , für die eine falsche Entscheidung getroffen werden kann, diese auch getroffen wird. Betrachtet wird die Prämisse P_s , die alle Beispiele des Umweltzustandes $z_{C_s} \neq z_{C_r}$ innerhalb \bar{P}_r erfasst. Der maximale Anteil an Beispielen von z_{C_r} für die die Fehlentscheidung $e(P_s)$ getroffen werden kann, ergibt sich aus (15):

$$\hat{p}_{max}(z_{C_r} \wedge P_s) = \underbrace{\frac{L(e(P_s), z_{C_s}) - L(e(P_r), z_{C_s})}{L(e(P_r), z_{C_r}) - L(e(P_s), z_{C_r})}}_{V_{s,r}} \cdot \hat{p}(z_{C_s} \wedge \bar{P}_r). \quad (16)$$

Dabei entspricht $V_{s,r}$ dem maximalen Verhältnis der Beispiele von z_{C_r} zu z_{C_s} in P_s . Mit dem tatsächlichen Anteil von z_{C_r} in der negierten Prämisse folgt:

$$\hat{p}(z_{C_r} \wedge P_s) = \min(\hat{p}_{max}(z_{C_r} \wedge P_s), \hat{p}(z_{C_r} \wedge \bar{P}_r)). \quad (17)$$

Im ungünstigsten Fall gilt für die negierte Prämisse

$$\hat{p}(z_{C_s} \wedge P_s) = \frac{\hat{p}(z_{C_r} \wedge P_s)}{V_{s,r}}. \quad (18)$$

Damit ergibt sich die Wahrscheinlichkeit von P_s mit (17) und (18) über den Satz der totalen Wahrscheinlichkeit zu:

$$\hat{p}(P_s) = \hat{p}(z_{C_r} \wedge P_s) + \hat{p}(z_{C_s} \wedge P_s). \quad (19)$$

Die Kosten für die Prämisse P_s bezogen auf den gesamten Datensatz sind dann:

$$\hat{L}_{D,P_s,total} = \hat{p}(z_{C_r} \wedge P_s) \cdot L(e(P_s), z_{C_r}) + \hat{p}(z_{C_s} \wedge P_s) \cdot L(e(P_s), z_{C_s}). \quad (20)$$

Die Kosten für die nicht erfassten Beispiele L_Z werden dann als gewichtete Summe der Kosten der Prämisen P_s bestimmt. Die Kosten der Prämisen P_s werden dabei

in der Reihenfolge absteigender Kosten der Fehlentscheidungen für z_{C_r} bestimmt. Begonnen wird mit der Fehlentscheidung, die die größten Kosten verursacht. Dabei wird $\hat{p}(z_{C_r} \wedge \bar{P}_r)$ jeweils um den Teil verringert, der bereits durch ein berechnetes P_s entschieden wird. Für L_Z folgt mit (19) und (20):

$$L_Z = \frac{\sum_{P_s} \hat{L}_{D,P_s,total}}{\sum_{P_s} \hat{p}(P_s)}. \quad (21)$$

Zur Bewertung der Regelhypothese R_r werden die Kosten der Beispiele in der Prämisse L_{D,P_r} und die Kosten L_Z für die nicht erfassten Beispiele z_{C_r} , für die eine Fehlentscheidung getroffen werden kann, gemäß ihrer Anteile gewichtet:

$$\hat{L}_{D,P_r,Z} = \frac{\hat{L}_{D,P_r} \cdot \hat{p}(P_r) + \alpha_{ER} \cdot L_Z \cdot \min(\hat{p}(z_{C_r} \wedge \bar{P}_r), p_{max}(z_{C_r} \wedge \bar{P}_r))}{\hat{p}(P_r) + \alpha_{ER} \cdot \min(\hat{p}(z_{C_r} \wedge \bar{P}_r), p_{max}(z_{C_r} \wedge \bar{P}_r))}, 0 \leq \alpha_{ER} \leq 1. \quad (22)$$

Der Faktor α_{ER} drückt eine Annahme über die ungewisse Güte der anderen Regeln aus, die nur impräzise bekannt ist. Dabei entspricht $\alpha_{ER} = 0$ der optimistischen Annahme, dass die von der Prämisse nicht erfassten Beispiele von z_{C_r} von einer anderen Regel korrekt entschieden werden. Die Bewertung beschränkt sich dann auf die Beispiele der Prämisse. Der Wert $\alpha_{ER} = 1$ drückt die pessimistische Einstellung aus, dass wenn möglich für jedes Beispiel von einer anderen Regel eine Fehlentscheidung getroffen wird. Damit existiert wegen $0 < \alpha_{ER} < 1$ eine impräzise Regelbewertung.

Einige Effekte der Bewertung von Einzelregeln werden an dem in Bild 5 dargestellten schematischen Beispiel mit den fünf Prämissen P_1, \dots, P_5 bezüglich des einzigen Merkmals x vorgestellt. Die Zahlenwerte verdeutlichen die Anzahl der Beispiele der Umweltzustände, die von den einzelnen Prämissen erfasst werden. Die dicken Linien in Bild 5 markieren exemplarisch die Beispiele von z_{C_r} , die nicht von der Prämisse P_2 erfasst werden.

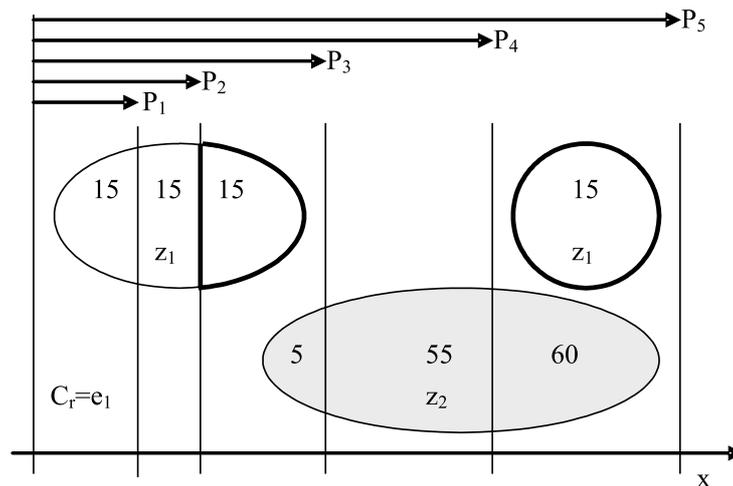


Bild 5: Veranschaulichung der Bewertung mittels der Beispiele in der Prämisse und z_{C_r} . Die Zahlen in den Bereichen entsprechen der jeweiligen Anzahl der Beispiele.

Werden nur die Beispiele in der Prämisse berücksichtigt ($\alpha_{ER} = 0$), erhalten P_1 und P_2 die beste Bewertung. Eine Differenzierung der beiden Regeln durch die Be-

	Prämisse P_r				
	1	2	3	4	5
$N(z_{c_r} \wedge \bar{P}_r)/\hat{p}(z_{c_r} \wedge \bar{P}_r)$	45/0.25	30/0.167	15/0.083	15/0.083	0/0
$N_{max}(z_{c_r} \wedge \bar{P}_r)/\hat{p}_{max}(z_{c_r} \wedge \bar{P}_r)$	12/0.06	12/0.06	11.5/0.057	6/0.02	0/0
$\hat{L}_{D,P_r,Z}$ mit $L_Z = 0.909$,					
$\hat{p}(z_{C_r} \wedge P_s)$, (21), (22), $\alpha_{ER} = 0$	0	0	0.1	0.57	0.67
$\hat{p}(z_{C_r} \wedge P_s)$, (21), (22), $\alpha_{ER} = 0.5$	0.26	0.15	0.18	0.58	0.67
$\hat{p}(z_{C_r} \wedge P_s)$, (21), (22), $\alpha_{ER} = 0.9$	0.38	0.240	0.238	0.59	0.67
$\hat{p}(z_{C_r} \wedge P_s)$, (21), (22), $\alpha_{ER} = 1$	0.40	0.26	0.25	0.59	0.67
$\hat{L}_{D,RB}$ kompletter Datensatz	0.67	0.67	0.67	0.67	0.67

Tabelle 1: Bewertungen $\hat{L}_{D,P_r,Z}$ für die Prämissen aus Bild 5. $L_R = 10$

wertung ist damit nicht möglich. Die größeren Prämissen ab P_3 verursachen Fehlentscheidungen, die zu höheren Kosten führen. Mit zunehmenden α_{ER} sind P_2 (z.B. bei $\alpha_{ER} = 0.5$) und dann P_3 (z.B. bei $\alpha_{ER} = 0.9$) optimal. Für $\alpha_{ER} = 1$ erhält P_3 die beste Bewertung. Dies ist ein guter Kompromiss, da P_4 lediglich die Fehlerzahl erhöht. Bei einer Bewertung als Entscheidung unter Risiko über den gesamten Datensatz erhalten bei diesem Beispiel alle Prämissen die selbe Bewertung, da für alle Prämissen und ihre jeweilige Negation stets e_1 die optimale Entscheidung ist. Eine solche Bewertung ist für die Regelsuche wertlos, da die Regeln hinsichtlich ihrer Güte nicht geordnet werden können.

2.4 Bewertung von Fuzzy-Regelbasen

Eine vollständige Regelbasis deckt den gesamten Wertebereich der Eingangsgrößen ab, wobei der nicht von wenigstens einer Prämisse abgedeckte Bereich zur SONST-Regel gehört. Eine komplette Regelbasis ordnet jedem Eingangsgrößenvektor \mathbf{x} eine Entscheidung zu: $\mathbf{x} \mapsto e_k$. Für den gesamten Datensatz liegt somit eine Risikosituation vor.

Bei einer Generierung von Regelbasen durch schrittweises Hinzufügen von Einzelregeln sind unvollständige Regelbasen zu bewerten. Eine unvollständige Regelbasis teilt den Lerndatensatz wie eine Einzelregel in zwei Teile: die Beispiele in den Prämissen und die Beispiele der SONST-Regel. Für die Beispiele in den Prämissen liegt wiederum eine Risikosituation vor. Die vorgestellten Inferenzverfahren bestimmen die Entscheidungen für diese Beispiele. Für die Beispiele der SONST-Regel wird eine Rückweisungsentscheidung mit Kosten \hat{L}_{RW} eingeführt. Der Anteil der Beispiele, für die die Rückweisungsentscheidung getroffen wird, sei \hat{p}_{RW} . Ungewissheit besteht darüber, ob eine weitere Regel der Regelbasis hinzugefügt und die Anzahl der Beispiele der SONST-Regel reduziert werden kann. Diese Ungewissheit ist in der Bewertung der Regelbasis analog zur Bewertung von Einzelregeln in (22) durch

$$\hat{L}_{D,P_{RB},RW} = (1 - \hat{p}_{RW}) \cdot \hat{L}_{D,P_{RB}} + \alpha_{RB} \cdot \hat{p}_{RW} \cdot \hat{L}_{RW} \quad (23)$$

zu berücksichtigen. Dabei sind \hat{L}_{RW} die Kosten für die nicht von den Prämissen erfassten Beispiele. Für \hat{L}_{RW} werden die Kosten angesetzt, die pro Beispiel entstehen,

wenn eine Default-Entscheidung für den gesamten Datensatz getroffen wird. Der Faktor α_{RB} drückt wiederum die Erwartung über die Qualität der weiteren Regeln aus, die als Kandidaten für die Aufnahme in die Regelbasis bereit stehen. Optimismus bezüglich der impräzisen Kosten spiegelt sich in $\alpha_{RB} \rightarrow 0$ und Pessimismus in $\alpha_{RB} \rightarrow 1$ wider.

3 Entscheidungstheoretische Entwurfsverfahren für Fuzzy-Systeme

3.1 Hypothesengenerierung Fuzzy-Einzelregeln

Die Generierung der Einzelregeln erfolgt in zwei Stufen [22]: Zuerst werden ein oder mehrere Entscheidungsbäume generiert und aus den Endknoten Regeln extrahiert. Diese Regeln werden anschließend über ein Pruningverfahren generalisiert.

Bei der Entwicklung der Entscheidungsbäume wird der Lerndatensatz Schritt für Schritt mittels verschiedener Merkmale aufgetrennt. Ausgehend von einem Ursprungsknoten für alle Beispiele befinden sich in den nachfolgenden Knoten Teile des Datensatzes. Jedem Knoten wird auf der Basis der enthaltenen Beispiele mittels (4) die optimale Entscheidung zugewiesen. Für die erwarteten Kosten gilt (3). Für die Auftrennung eines Knotens wird über ein noch nicht verwendetes Merkmal eine Regelbasis mit m_l -Regeln und den zugehörigen optimalen Entscheidungen erzeugt. Eine Aufspaltung erfolgt für das Merkmal mit den niedrigsten Kosten für die Regelbasis, wenn diese niedriger sind als die Kosten des übergeordneten Knotens.

Ziel des Prunings ist es, aus der Menge der spezialisierten Regelhypothesen durch Generalisierung (Vergrößerung der Gebiete der Prämissen) die Kandidaten für die folgende Suche einer Regelbasis zu bilden. Die Einzelregeln werden während des Prunings nach (22) bewertet.

Das Pruning wird sequentiell für die Hypothesen durchgeführt. Zunächst werden aus der aktuellen Regel generalisierte Hypothesen abgeleitet. Dabei stehen zwei Optionen zur Generalisierung zur Verfügung. Entweder wird eine Teilprämisse aus der Regel gestrichen, oder zu einer Teilprämisse werden benachbarte Terme hinzugefügt [22]. Aus den so erzeugten Hypothesen wird die beste für den nächsten Pruningschritt ausgewählt. Das Pruning einer Einzelregel ist beendet, wenn keine Verbesserung mehr eintritt.

Im einfachsten Fall werden die Merkmalskosten einer Regel $L_{C,R}$ direkt zu den Entscheidungskosten $\hat{L}_{D,P_r,Z}$ hinzuaddiert:

$$\hat{L}_{T,R} = \hat{L}_{D,P_r,Z} + L_{C,R}. \quad (24)$$

Dies kann allerdings dazu führen, dass Regeln zu weit generalisiert werden. Dieser Fall tritt ein, wenn die Reduktion der Merkmalskosten durch das Weglassen einer Teilprämisse größer ist als eine dadurch verursachte Erhöhung der Entscheidungskosten. Die Regel wird dann besser bewertet und die vorhergehende gelöscht.

Bessere Ergebnisse liefert ein zweikriterielles Optimierungsproblem:

$$\begin{aligned} \min \hat{L}_{D,P_r,Z} \\ \min \hat{L}_{T,R}. \end{aligned}$$

Beim Pruning der einzelnen Regeln ist ungewiss, welche Merkmale in der zu generierenden Regelbasis verwendet werden. Die durch eine einzelne Regel verursachten (zusätzlichen) Merkmalskosten für eine Regelbasis sind nicht präzise bekannt und liegen in einem Intervall zwischen 0 (die Merkmale werden bereits durch eine andere Regel verwendet und verursachen keine Zusatzkosten, die Generalisierung einer Regel zugunsten geringerer Merkmalskosten ist somit überflüssig) und $L_{C,R}$ (keine andere Regel verwendet diese Merkmale, die Generalisierung ist sinnvoll).

Folgende Situationen für die Komponenten aus (24) sind möglich:

- bei gleichen Merkmalskosten liegen unterschiedliche Entscheidungskosten vor,
- bei gleichen Entscheidungskosten liegen unterschiedliche Merkmalskosten vor,
- bei niedrigeren Merkmalskosten liegen niedrigere Entscheidungskosten vor,
- bei niedrigeren Merkmalskosten liegen höhere Entscheidungskosten vor.

Die ersten drei Fälle sind eindeutig. Bei gleichen Merkmalskosten dominiert die Regel mit den niedrigeren Entscheidungskosten, bei gleichen Entscheidungskosten die Regel mit den niedrigeren Merkmalskosten. Sind bei einer Regel beide Kostenarten niedriger, so ist diese Regel ebenfalls dominant. Im vierten Fall sind zusätzlich die Gesamtkosten zu berücksichtigen. Betrachtet werden zwei Regeln R_1 und R_2 . Dabei sei R_1 die beste Regel des aktuellen Pruningschrittes nach (22). Liegt die folgende Situation vor:

$$\begin{aligned}\hat{L}_{D,P_1,Z} &< \hat{L}_{D,P_2,Z} \\ L_{C,R_1} &> L_{C,R_2} \\ \hat{L}_{T,R_1} &> \hat{L}_{T,R_2},\end{aligned}\tag{25}$$

ist ohne Berücksichtigung von Merkmalskosten (nur \hat{L}_D) R_1 vorzuziehen. Bei der Berücksichtigung von Merkmalskosten nach (24) ist dagegen R_2 zu wählen, da die Gesamtkosten niedriger sind. In diesem Fall sollen beide Regeln für das weitere Pruning als Kandidaten für die Regelbasis erhalten bleiben. Die diskutierten Zusammenhänge sind in Bild 6 grafisch dargestellt.

3.2 Hypothesengenerierung Fuzzy-Regelbasen

Das Eröffnungsverfahren beginnt mit einer Default-Entscheidung (SONST-Regel) für alle Beispiele des Lerndatensatzes und einer leeren Regelbasis ($N_R = 0$). Anschließend werden der Regelbasis Schritt für Schritt Regeln hinzugefügt. Dabei wird stets die Regel R_i^{Hyp} aus der Menge aller Regelhypothesen ausgewählt, die die Kosten \hat{L}_T pro Entscheidung minimiert. Die ausgewählte Regel wird an der Stelle N_R+1 in die Regelbasis eingefügt:

$$R_{N_R+1}^{RB} = \arg \min_{i \in \{1 \dots N_{R,Hyp}, \emptyset\}} \hat{L}_T(R_1^{RB}, \dots, R_{N_R}^{RB}, R_i^{Hyp}).\tag{26}$$

Dabei ist $N_{R,Hyp}$ die Anzahl der Regeln nach dem Pruning. Die Merkmalskosten L_C pro Entscheidung können optional Null gesetzt werden. Die Suche nach weiteren Regeln bricht ab, wenn keine Reduktion der Kosten durch eine weitere Regel eintritt. Um große Regelbasen zu vermeiden, kann eine Schranke für die minimale Kostenreduktion einer zusätzlichen Regel gesetzt werden. Zusätzlich ist eine Obergrenze für die Anzahl der Regeln möglich.

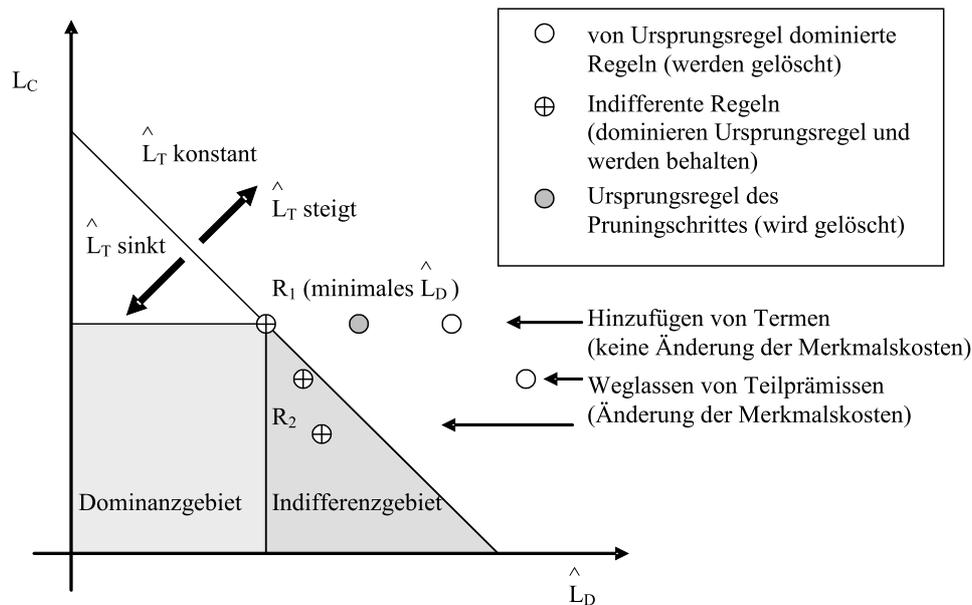


Bild 6: Darstellung der Dominanzsituationen beim Pruning von Einzelregeln.

Folgende Entwurfsoptionen stehen zur Verfügung:

1. Die Bewertung erfolgt als vollständige Regelbasis. Eine SONST-Regel mit vom Anwender vorgegebener Default-Entscheidung wird der Regelbasis hinzugefügt. Diese Option ist dann sinnvoll, wenn vornehmlich Regeln für alle anderen Klassen außer der Default-Entscheidung gefunden werden sollen.
2. Die Bewertung erfolgt als vollständige Regelbasis. Eine SONST-Regel mit vom Algorithmus festgelegter Entscheidung wird der Regelbasis hinzugefügt. Dies entspricht der pessimistischen Annahme, dass keine weitere Regel mehr zu finden ist. Diese Option ist sinnvoll, um eine stark streuende und/oder nichtkompakte Klasse durch eine SONST-Regel abzudecken.
3. Die Bewertung erfolgt als unvollständige Regelbasis. Eine SONST-Regel mit der Rückweisungsentscheidung wird der Regelbasis hinzugefügt. Die Kosten \hat{L}_{RW} je Beispiel für diese Entscheidung werden vom Algorithmus bestimmt. Erst im letzten Schritt wird eine SONST-Regel entsprechend 2. ergänzt. Diese Option findet tendenziell Regelbasen, bei denen für jede Klasse wenigstens eine Regel existiert.

Bild 7 zeigt die Aufteilung der Kosten für die Verwendung der Rückweisungsentscheidung schematisch. Zunächst sind die Kosten vollständig ungewiss. Mit jeder zusätzlichen Regel geht ein Teil in Kosten unter Risiko über.

Weitere Optionen umfassen eine evolutionäre Auswahl von Regeln oder eine Parameteroptimierung von Zugehörigkeitsfunktionen im Anschluss an die Auswahl einer Regelbasis. Allerdings konnten dadurch bei den meisten Anwendungen keine signifikanten Vorteile gegenüber dem dargestellten Verfahren erzielt werden.

4 Benchmarkdatensatz Kfz-Aggregate

Zur Demonstration der Leistungsfähigkeit des vorgestellten Ansatzes wird der Benchmarkdatensatz Kfz-Aggregate der Universität Dortmund und des GMA-Fach-

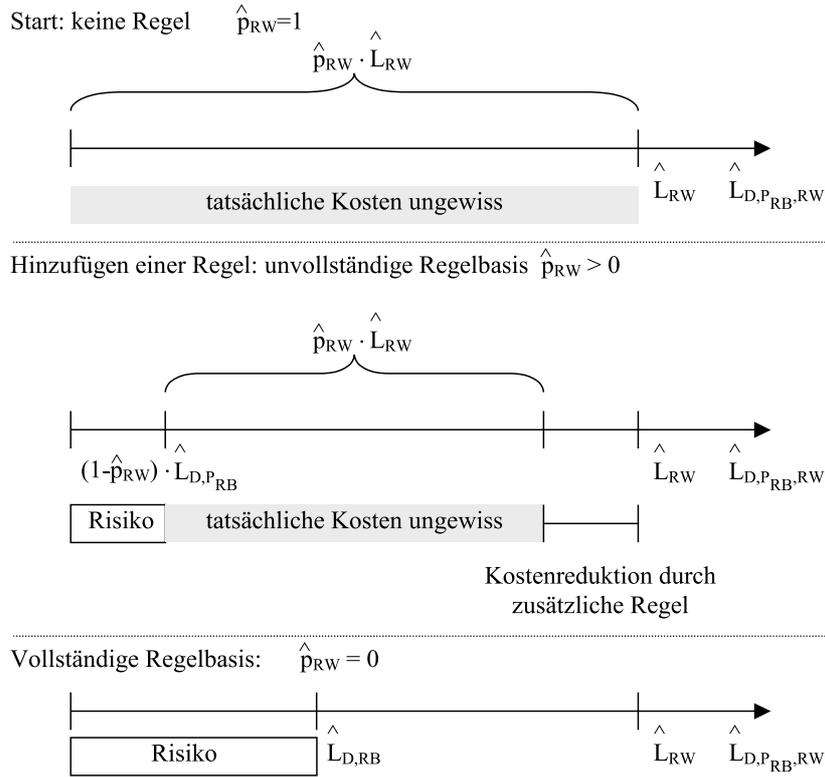


Bild 7: Aufteilung der Entscheidungskosten bei der Regelbasissuche. Die Kosten der Rückweisung \hat{L}_{RW} sind als konstant angenommen.

ausschusses 5.22 verwendet [4, 5, 23, 24]. Die Daten stammen von Prüfläufen von Kfz-Getrieben. Ziel ist die automatische Erkennung von defekten Aggregaten anhand von akustischen Signalen. Merkmalskosten werden bei diesem Datensatz nicht berücksichtigt. Der Benchmarkdatensatz besteht aus den drei Teilen A, B und C mit 2612, 4373 und 10613 Beispielen. Aus den akustischen Signalen des Prüflaufs eines Aggregates werden 304 Merkmale (Spalten im Datensatz $x_4 \dots x_{307}$ wie in [5]) abgeleitet, wobei für jeden Teil und jedes Merkmal eine separate Normierung auf Mittelwert Null und Standardabweichung Eins [4] sowie eine Reduktion auf 15 Merkmale mit einer multivariaten Varianzanalyse [5] erfolgt. Der Anteil der defekten Aggregate beträgt in allen Teilen des Datensatzes jeweils ca. 1%. Die beiden Umweltzustände und entsprechend auch die beiden Entscheidungen sind 'Defekt (nio)' (z_1/e_1) oder kein Defekt (io)' (z_2/e_2). Es wird die Kostenmatrix aus (2) verwendet.

Wegen der starken Asymmetrie der Umweltzustände wird bei der Suche der Regelbasen die Option eins (feste Default Entscheidung 'io') verwendet. Dadurch werden gezielt Regeln zur Identifikation defekter Getriebe in die Regelbasis aufgenommen.

Der Hersteller der Aggregate stellt Anforderungen an die Erkennungsrate ER der defekten Aggregate

$$ER = \hat{p}(e_1|z_1) > 0.7 \quad (27)$$

und die Glaubwürdigkeit G

$$G = \frac{\hat{p}(e_1 \wedge z_1)}{\hat{p}(e_1 \wedge z_1) + \hat{p}(e_1 \wedge z_2)} = \frac{\hat{p}(e_1 \wedge z_1)}{\hat{p}(e_1)} > 0.3 \quad (28)$$

eines Entscheidungssystems. Bild 8 zeigt die Erkennungsrate und Glaubwürdigkeit für verschiedene Kostenverhältnisse L_R . Teil C des Datensatzes wird dabei als Lerndatensatz verwendet. Erwartungsgemäß steigt die Erkennungsrate mit steigendem Kostenverhältnis und die Glaubwürdigkeit nimmt ab. Bei steigendem Kostenverhältnis nimmt die Zahl der Fehlentscheidungen für Beispiele von z_2 zu, die durch ein zusätzlich erkanntes Beispiel von z_1 kompensiert werden.

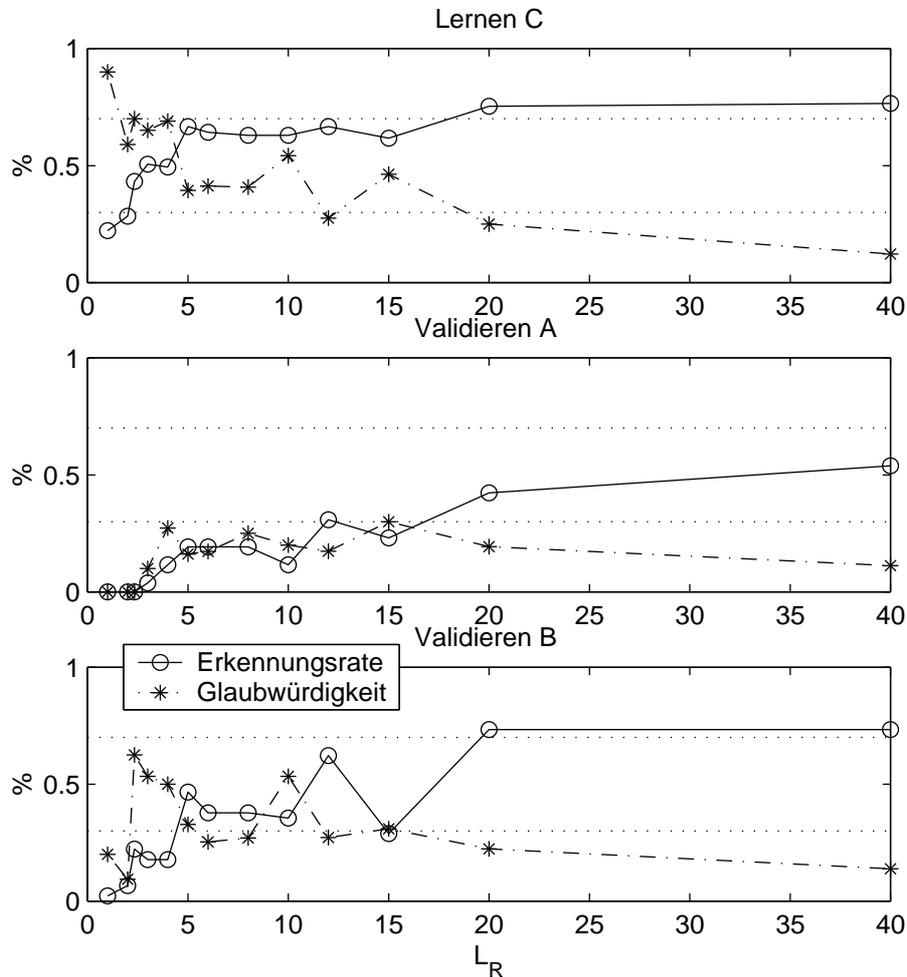


Bild 8: Erkennungsrate und Glaubwürdigkeit für den Kfz-Benchmark-Datensatz bei variablem Kostenverhältnis für die Option vollständige Prämisse.

Ein Ergebnis, das den Anforderungen über allen Datensätzen (Lernen mit Teil C) relativ nahe kommt, stellt sich für $L_R = 20$ ein (Tabelle 3). Die entsprechende Regelbasis ist in Tabelle 2 dargestellt. Der Vergleich mit anderen Ansätzen für diesen Datensatz zeigt die Vorteile des in dieser Arbeit vorgestellten Verfahrens. Das in [5] vorgestellte Verfahren bedarf einer Vorverarbeitung der Merkmale. Zusätzlich wird die Zusammensetzung des Lerndatensatzes (Teil C) geändert. Die Zahl der defekten Aggregate wird konstant gehalten und die Zahl der einwandfreien erheblich reduziert. Folgende Werte für Erkennungsrate und Glaubwürdigkeit sind mit diesem Verfahren für Teil C als Lerndatensatz zu erreichen:

Die Ergebnisse über den Lerndaten sind beim entscheidungstheoretischen Verfahren etwas besser. Über den beiden Validierungsdatensätzen ergeben sich deutlichere

R_1 :	WENN $(x_{86}=\text{PK} \cup \text{PG}) \cap (x_{103}=\text{PK} \cup \text{PG}) \cap (x_{87}=\text{PK} \cup \text{PG}) \cap (x_{70}=\text{PK} \cup \text{PG}) \cap (x_{219}=\text{PK} \cup \text{PG})$	DANN $y=e_1$
R_2 :	WENN $x_{22}=\text{PG} \cap (x_{103}=\text{ZE} \cup \text{PK} \cup \text{PG})$	DANN $y=e_1$
R_3 :	WENN $(x_{67}=\text{PK} \cup \text{PG}) \cap x_{70}=\text{PG}$	DANN $y=e_1$
R_4 :	WENN $x_{14}=\text{ZE} \cap (x_{86}=\text{PK} \cup \text{PG}) \cap (x_{22}=\text{PK} \cup \text{PG}) \cap (x_{103}=\text{NK} \cup \text{ZE}) \cap (x_{87}=\text{PK} \cup \text{PG}) \cap (x_{220}=\text{NG} \cup \text{NK}) \cap x_{16}=\text{ZE} \cap (x_{187}=\text{NK} \cup \text{ZE})$	DANN $y=e_1$
R_5 :	WENN $(x_{14}=\text{NK} \cup \text{ZE}) \cap (x_{86}=\text{PK} \cup \text{PG}) \cap x_{22}=\text{ZE} \cap (x_{87}=\text{PK} \cup \text{PG}) \cap (x_{220}=\text{NG} \cup \text{NK}) \cap x_{16}=\text{PK} \cap (x_{218}=\text{ZE} \cup \text{PK})$	DANN $y=e_1$
R_6 :	WENN $(x_{218}=\text{PK} \cup \text{PG}) \cap x_{219}=\text{PG}$	DANN $y=e_1$
R_7 :	WENN $x_{103}=\text{PG} \cap (x_{70}=\text{PK} \cup \text{PG})$	DANN $y=e_1$
R_8 :	WENN $(x_{14}=\text{PK} \cup \text{PG}) \cap (x_{86}=\text{PK} \cup \text{PG}) \cap (x_{103}=\text{PK} \cup \text{PG}) \cap (x_{218}=\text{PK} \cup \text{PG}) \cap (x_{236}=\text{PK} \cup \text{PG})$	DANN $y=e_1$
R_9 :	WENN $(x_{14}=\text{PK} \cup \text{PG}) \cap (x_{86}=\text{PK} \cup \text{PG}) \cap (x_{103}=\text{PK} \cup \text{PG}) \cap (x_{218}=\text{PK} \cup \text{PG}) \cap (x_{70}=\text{PK} \cup \text{PG})$	DANN $y=e_1$
	SONST	DANN $y=e_2$

Tabelle 2: Entworfenene Regelbasis für $L_R = 20$ (Terme: NG, NK, ZE, PK, PG mit N: negativ, P: positiv, K: klein, G: groß, ZE: null).

Unterschiede. Für den Teil A liefert das Verfahren aus [5] die besseren Ergebnisse, für den Teil B das entscheidungstheoretische. Der entscheidungstheoretische Zugang stellt hier die benutzerfreundlichere Variante dar. Bis auf die ebenfalls durchgeführte Reduktion der Merkmale von 304 auf 15 entfallen Manipulationen der Daten. Ohne diese Manipulationen liefert das Verfahren aus [5] keine brauchbaren Ergebnisse (Tabelle 3). Über den Lerndaten wird eine Erkennungsrate von 26% bei einer Glaubwürdigkeit von 80% erreicht. Über dem Teil A des Datensatzes ergibt sich mit der erzeugten Regelbasis eine Erkennungsrate von 0%. Über dem Teil B des Datensatzes wird eine Erkennungsrate von 4.4% bei einer Glaubwürdigkeit von 33% erreicht. Diese Ergebnisse sind deutlich schlechter als bei entscheidungstheoretischer Bewertung.

In [4] werden Ergebnisse für das Fuzzy-ROSA²-Verfahren unter zusätzlicher Berücksichtigung von Expertenwissen bei der Merkmalsauswahl vorgestellt. Es werden mit 40 Merkmalen deutlich mehr Merkmale verwendet als beim entscheidungstheoretischen Ansatz dieser Arbeit. Die Güte der Ergebnisse ist vergleichbar. Jedoch sind die Regelbasen des entscheidungstheoretischen Ansatzes wesentlich kleiner. Für das Kostenverhältnis von $L_R = 20$ enthält die Regelbasis 9 Regeln zuzüglich der Default-Regel. Die in [4] erzeugten Regelbasen haben zwischen 206 und 5200 Regeln.

Zusammenfassend liefert der entscheidungstheoretische Ansatz bei diesem Datensatz vergleichbare Ergebnisse zu anderen regelbasierten Verfahren. Der Vorteil des entscheidungstheoretischen Ansatzes resultiert daraus, dass weder zusätzliches Expertenwissen noch eine Manipulationen der Beispiele des Lerndatensatzes nötig ist.

²Rule Oriented Statistical Analysis

Ansatz	Daten	N_R	Erkennungsrate	Glaubwürdigkeit
Entscheidungstheorie $L_R = 20$	Teil A	9	42.3%	19.3%
	Teil B		73.3%	22.3%
	Teil C		75.3%	25.0%
[5] mit Vorverarbeitung/ Datenreduktion	Teil A	4	53.8%	26.9%
	Teil B		60.0%	20.6%
	Teil C		72.8%	21.8%
[5] ohne Vorverarbeitung/ Datenreduktion	Teil A	3	0.0%	0.0%
	Teil B		4.4%	33.0%
	Teil C		26.0%	80.0%
Ergebnisse aus [4]	Teil A	5200	50.0%	12.9%
	Teil B		60.0%	21.8%
	Teil C		66.7%	100.0%

Tabelle 3: Vergleich der Ergebnisse für den Kfz-Benchmark-Datensatz. Lernen Teil C.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurden entscheidungstheoretische Maße und Verfahren zur Bewertung und zum Entwurf von Fuzzy-Systemen vorgestellt, anhand eines Benchmarkdatensatzes evaluiert und die Leistungsfähigkeit demonstriert. Der vorgeschlagene Ansatz ist in der Lage, verschiedene Kostenarten - Entscheidungskosten und Merkmalskosten - in den Entwurfsprozess von Entscheidungssystemen zu integrieren. Weiterhin kommt der Ansatz weitgehend ohne aufwendige Vorverarbeitung der Daten aus und ist damit benutzerfreundlicher als vergleichbare Verfahren.

Bei vielen Problemen ist es sinnvoll, dass das Entscheidungssystem bei unzureichender Information keine Entscheidung trifft. Der vorgestellte Ansatz ist über ein zusätzliches evidenztheoretisches Inferenzverfahren in der Lage, diese Aufgabe zu erfüllen.

Literatur

- [1] Frank, M.: Diagnoseverfahren in der Automatisierungstechnik. *at-Automatisierungstechnik* 2/94 (1994), S. 47–64.
- [2] Neumann, U.: *Integrierte Instandhaltungsplanung für elektrische Energieübertragungssysteme*. VDI Verlag GmbH. 1998.
- [3] Nyberg, M.: Automatic Design of Diagnosis Systems with Application to an Automotive Engine. *Control Engineering Practice* (7)8 (1999), S. 993–1005.
- [4] Haendel, L.: Benchmark: Qualitätskontrolle bei KFZ-Aggregaten. In: *Proc. 12. Workshop Fuzzy Systeme* (Mikut, R.; Reischl, M., Hg.), S. 119–126. FZK Wissenschaftliche Berichte FZKA 6767. 2002.
- [5] Mikut, R.: Fuzzy-Modellbildung für den Benchmarkdatensatz Kfz-Aggregate. In: *Proc. 12. Workshop Fuzzy Systeme* (Mikut, R.; Reischl, M., Hg.), S. 127–134. FZK Wissenschaftliche Berichte FZKA 6767. 2002.

- [6] Loose, T.: *Modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse*. Universitätsverlag Karlsruhe. 2004.
- [7] Sboner, A.; Eccher, C.; Blanzieri, E.; Bauer, P.; Cristofolini, M.; Zumiani, G.; Forti, S.: A Multiple Classifier System for Early Melanoma Diagnosis. *Artificial Intelligence in Medicine* 27 (2003), S. 29–44.
- [8] Bitz, M.: *Entscheidungstheorie*. Verlag Vahlen. 1981.
- [9] Laux, H.: *Entscheidungstheorie*. Springer-Verlag. 2003.
- [10] Raiffa, H.: *Einführung in die Entscheidungstheorie*. R. Oldenbourg Verlag München Wien. 1973.
- [11] Rommelfanger, H.: *Fuzzy Decision Support-Systeme*. Springer-Verlag. 1993.
- [12] Turney, P.: Types of Cost in Inductive Concept Learning. In: *Proc. Workshop On Cost-Sensitive Learning at the 17th Int. Conference on Machine Learning (WCSL at ICML-2000)*, S. 15–21. Stanford University, California. 2000.
- [13] Borgelt, C.; Timm, H.; Kruse, R.: Unsicheres und Vages Wissen. In: *Handbuch der Künstlichen Intelligenz* (Görz, G.; Rollinger, C.-R.; Schneeberger, J., Hg.). München: Oldenbourg. 2000.
- [14] Bouchon-Meunier, B.: Uncertainty Management in Medical Applications. In: *Nonlinear Biomedical Signal Processing, Vol. 1, Fuzzy Logic, Neural Networks, and New Algorithms* (Akay, M., Hg.), S. 1–26. Wiley. 2000.
- [15] Klir, G. J.: Generalized Information Theory: Aims, Results, and Open Problems. *Reliability Engineering & System Safety* 85 (1-3) (2004), S. 21–38.
- [16] Adams, N. M.; Hand, D. J.: Comparing classifiers when the misallocation costs are uncertain. *Pattern Recognition* 32 (7) (1999), S. 1139–1147.
- [17] Bamberg, G.: *Statistische Entscheidungstheorie*. Physica Verlag. 1972.
- [18] Mikut, R.; Jäkel, J.; Gröll, L.: Automatic Design of Interpretable Membership Functions. In: *Proc. 8th Zittau Fuzzy Colloquium, Sep. 6-8, 2000*, S. 103–111. Hochschule Zittau/Görlitz. 2000.
- [19] Bandemer, H.: *Ratschläge zum mathematischen Umgang mit Ungewißheit*. B.G. Teubner Verlagsgesellschaft Stuttgart. 1997.
- [20] Zadeh, L.: Probability Measures of Fuzzy Events. *Journal of Mathematical Analysis and Applications* 23 (1968), S. 421–427.
- [21] Mikut, R.; Jäkel, J.; Gröll, L.: Inference Methods for Partially Redundant Rule Bases. In: *Fuzzy Control: Theory and Practice* (Hampel, R.; Wagenknecht, M.; Chaker, N., Hg.), *Advances in Soft Computing*, S. 177–185. Heidelberg: Physica. 2000.
- [22] Mikut, R.; Jäkel, J.; Gröll, L.: Interpretability issues in data-based learning of fuzzy systems. *Accepted Paper: Fuzzy Sets and Systems* (2004).
- [23] Priber, U.: Ergebnisse der Untersuchung von Benchmarkdaten mittels FUCS - Fuzzy Classification System. In: *Proc. 11. Workshop Fuzzy Control des GMA-FA 5.22* (Mikut, R.; Reischl, M., Hg.), S. 100–105. Forschungszentrum Karlsruhe, FZKA 6660. 2001.
- [24] Slawinski, T.: *Analyse und effiziente Generierung von relevanten Fuzzy-Regeln in hochdimensionalen Suchräumen*. VDI Verlag Düsseldorf, Fortschrittsberichte Reihe 10, Nr. 686. 2001.

Temporal erweiterte Prädikate der Fuzzy-Logik zur Überwachung und Wartung

Thorsten W. SCHMIDT, Dominik HENRICH

Universität Bayreuth

Fakultät für Mathematik, Physik und Informatik

Lehrstuhl Angewandte Informatik III

D-95440 Bayreuth

E-Mail: {thorsten.w.schmidt, dominik.henrich}@uni-bayreuth.de

[Http://ai3.inf.uni-bayreuth.de](http://ai3.inf.uni-bayreuth.de)

1. Einleitung

1.1. Motivation

Seit vielen Jahren schon beschäftigt sich die Forschung mit Fuzzy-Logik und ihrer Anwendung in Fuzzy-Reglern. Seit einigen Jahren gibt es auch schon industriell eingesetzte Fuzzy-Regler. Diese werden zum Beispiel in Waschmaschinen oder anderen Geräten des häuslichen Gebrauchs verwendet. Die Vorteile der Fuzzy-Logik sind, dass vorhandenes Wissen über ein Regelungsprozess sehr leicht zur Modellierung eines Reglers verwendet werden kann und im weiteren Verlauf der Entwicklung und Verbesserung des Reglers dieses Wissen immer transparent bleibt und somit nicht verloren geht. Durch die klare Lesbarkeit der Fuzzy-Logik bleibt das Wissen auch wartbar. Diese Fuzzy-Regler können aber nicht als Wartungssystem eingesetzt werden, da sie nicht in der Lage sind zeitliche Abhängigkeiten von Ereignissen untereinander oder überhaupt Zeit zu modellieren, wie dies beim Modell Checking möglich ist, welches Temporale Logik verwendet [Karjoth87]. Aus diesem Grund ist eine Erweiterung der Fuzzy-Logik um zeitliche Aspekte nötig. Es existieren ein paar Ansätze, um die Zeit in Fuzzy-Logik einzubringen. In Kapitel 2 werden vier solcher Ansätze vorgestellt.

1.2. Problemstellung

Ein Problem bei Fuzzy-Reglern ist, dass es keine sachlich motivierten Ansätze gibt, durch welche klar ersichtlich wird, wie zeitliche Aspekte in Fuzzy-Logik gehandhabt werden können. Model Checking mit Temporaler Logik ist eine in sich geschlossene Methodik. Die temporalen Prädikate können zur Beschreibung von Szenarien genutzt werden um Prozesszustände mit zeitlichen Abhängigkeiten zu modellieren. Das Problem, welches im folgenden gelöst wird, beschäftigt sich nun damit, Prädikate der Temporalen Logik in die Fuzzy-Logik zu übersetzen und zwar so, dass zum Einen die so genannte Temporale Fuzzy-Logik genauso mächtig ist wie die Temporale Logik und zum Anderen eine solide mathematische Basis geschaffen wird, so dass die Bedingungen, welche an Fuzzy-Prädikate und Zugehörigkeitsfunktionen gestellt werden auch erfüllt sind. Gesucht ist demnach die Vereinigung der Temporalen Logik und Fuzzy-Logik zur *Temporalen Fuzzy-Logik*.

Im weiteren Verlauf dieses Artikels beschreibt das zweite Kapitel den Stand der Forschung während das dritte Kapitel die verwendeten Abkürzungen und Definitionen für die späteren Kapitel beschreibt. Kapitel 4 zeigt die Einreihung der Temporalen Fuzzy-Logik im Vergleich zur temporalen/atemporalen beziehungsweise scharfen/unscharfen Logik. Das nächste, fünfte Kapitel definiert alle Prädikate der Temporalen Fuzzy-Logik. Kapitel 6 beschreibt eine Anwendung der Temporalen Fuzzy-Logik an einem kon-

kreten Beispiel. Kapitel 7 fasst die Arbeit zusammen und beschreibt die daraus gewonnenen Schlussfolgerungen.

1.3. Abgrenzung

Ein *Überwachungssystem* ist ein von einem zu überwachenden Prozess unabhängiges System, welches diesen Prozess in seinem Verhalten mittels Sensoren überwacht. Prozesskennzahlen informieren dabei über die internen, nicht zwingenderweise bekannten, Zustände des Prozesses. Bei den Aktuatoren, ist nicht immer bekannt, welchen quantitativen Einfluss diese auf den Prozess haben, somit ist eine Steuerung des Prozesses nicht möglich. Ist jedoch der qualitative Einfluss bekannt, so kann der Prozess geregelt werden. Werden nun in dem Verhalten des Prozesses Abweichungen zu den gewünschten benutzerdefinierten Vorgaben erkannt, kann ein *Regler* in das Verhalten des Prozesses eingreifen und Parameter so verändern, dass das Verhalten des Prozesses sich dem Verhalten nähert, welches von einem Benutzer gewünscht wird [Castillo02]. Ein einfaches Beispiel für ein Überwachungssystem ist die Überwachung der Helligkeit in einem Raum. Sinkt die Helligkeit unter einen angegebenen Schwellwert, so erkennt dies das Überwachungssystem und gibt dem Prozess den Auftrag den Raum stärker zu beleuchten, indem mehr Lampen angeschaltet werden.

Ein *vorausschauendes Überwachungssystem* benutzt nicht nur aktuelle Sensordaten aus dem Prozess, sondern auch mögliche Sensordaten aus der Zukunft [Fantoni00], [Palit00]. Natürlich ist es nicht möglich diese zukünftigen Daten zu messen. Sie müssen mit geeigneten Methoden aus dem bekannten vergangenen Signalverlauf vorhergesagt werden. Werden diese zukünftigen Sensorwerte an ein Überwachungssystem gegeben, so kann dieses eine zukünftige Abweichung vom gewünschten, benutzerdefinierten Verhalten feststellen. Das Eintreten der Abweichung ist dabei nicht garantiert; sie muss nicht eintreten. Im Gegenteil, dadurch, dass dem Überwachungssystem bekannt ist, was bei den aktuellen Parameterwerten in der Zukunft passieren würde, können schon frühzeitig Maßnahmen ergriffen werden, um ein anderes Verhalten herbei zu führen. Wenn nun im obigen Beispiel Lampen verwendet werden, welche eine lange Zeit benötigen, um ihre maximale Helligkeit zu erreichen (Neonröhren mit einem sehr hohen Wirkungsgrad: ca. 15 Minuten, Energiesparlampen: ca. 10 Minuten oder ähnliche), so genügt ein nicht vorausschauendes Überwachungssystem nicht mehr. Das vorausschauende Überwachungssystem kann jedoch feststellen, dass es im Raum immer dunkler wird. Bevor es im Raum zu dunkel ist, also die Helligkeit den angegebenen Schwellwert unterschreitet, schaltet das Überwachungssystem weitere Lampen ein.

Ein *Wartungssystem* baut meistens, nicht immer, auf einem Diagnosesystem auf [Althoff92]. Das in dieser Arbeit vorzustellende Wartungssystem baut jedoch auf einem (vorausschauenden) Überwachungssystem auf. Dann, wenn ein Überwachungssystem durch Veränderung der Prozessparameter keine Verbesserung mehr erreichen kann und sich die Prozesskennzahlen nicht innerhalb eines tolerierbaren Bereiches befinden, liegt ein *Fehler* im System vor, welcher nicht ausgeglichen werden kann. Dieser Fehler kann zum Beispiel eine defekte Teilkomponente sein, welche ersetzt werden muss. Das Wartungssystem generiert in diesem Fall einen Wartungsauftrag für einen Benutzer und teilt diesem mit, welche Teilkomponente einen Fehler verursacht haben könnte. Durch den vorausschauenden Aspekt eines Wartungssystems können Ausfälle dieser Art frühzeitig vorhergesagt und hierfür Wartungsaufträge generiert werden. Die Wartungsaufträge können zeitlich in der Zukunft datiert sein, da der vorhergesagte Ausfall nicht unmittelbar, sondern in der Zukunft eintritt. Das System wird so lange betrieben, wie es funktionsfähig bleibt, also seine Prozesskennzahlen in einem tolerierbaren Bereich liegen. Die Abstände zwischen verschiedenen Wartungen, bei welchen ein

Bediener die Maschine anhält und sie repariert sind maximal. Dadurch, dass die Wartungsaufträge auch in der Zukunft liegen können und es so möglich ist mehrere Wartungsaufträge zu sammeln und zu einem Zeitpunkt alle Wartungsarbeiten parallel auszuführen, wird der Prozess nur einmal angehalten und somit die Standzeiten (*Wartungszeiten*) verringert. So entsteht kein außerplanmäßiger Produktionsausfall, denn die Wartungsarbeiten können eventuell zu Zeiten geringer Auslastung durchgeführt werden. Im obigen Beispiel entspricht dies dem Überwachungssystem, welches versucht den Raum durch Einschalten weiterer Lampen zu erhellen. Da aber defekte Lampen existieren, liegt die Helligkeit auch dann noch unter dem gegebenen Schwellwert, wenn alle Lampen angeschaltet sind. In diesem Fall generiert das Wartungssystem einen Wartungsauftrag, in welchem es dem Benutzer mitteilt, defekte Lampen im Raum auszutauschen. Eine andere Möglichkeit, dass ein Wartungssystem einen Wartungsauftrag generiert, ist wenn festgestellt wird, dass Lampen Anzeichen für einen baldigen Defekt aufweisen, so dass dann nicht mehr genügend Licht produziert werden kann.

Als Endziel, welches über die Temporalen Fuzzy-Prädikate dieses Papers hinaus geht, möchten wir ein Wartungssystem, wie oben beschrieben entwickeln. Das Wartungssystem generiert Wartungsaufgaben für einen zu wartenden Prozess. Die Wartungsaufgaben werden dabei durch Fuzzy-Logik Regeln beschrieben. Natürlich kann das Wartungssystem auch als Überwachungssystem oder Regelungssystem verwendet werden. Das Hauptaugenmerk liegt jedoch auf dem Wartungssystem, welches Wartungsaufträge generiert. Die automatisch generierten Wartungsaufträge müssen dabei zeitlich so geplant sein, dass eine möglichst ökonomische Abarbeitung ohne größere Standzeiten des Prozesses möglich ist.

2. Stand der Forschung

In der Forschungsabteilung der Firma Flender Service GmbH werden spezielle Sensoren zur Überwachung entwickelt [Flender01], [Flender02]. Diese Sensoren dienen der Überwachung von beispielsweise Fräsmaschinen, welche Lager, Zahnräder und andere Komponenten mit hohem Verschleiß beinhalten. Zur Beobachtung der Komponenten mit hohem Verschleiß werden Schwingungssensoren verwendet, denn die Erfahrung hat gezeigt, dass sich das Schwingungsprofil dieser Komponenten charakteristisch mit dem Erreichen des Endes der Lebensdauer verändert. Das heißt die Schwingungen werden aufgenommen, zur Steuereinheit übertragen und dort Fouriertransformiert. Aus den Signalverläufen werden unter Verwendung von mit diesen Maschinen gesammelten Wissen Rückschlüsse auf die weitere Lebensdauer der Komponenten gezogen, so dass diese ausgetauscht werden können bevor es zu einem Ausfall kommt. Einsetzbar ist dieses System in allen Maschinen, an welchen über Jahre hinweg der Verschleiß dieser Komponenten gemessen und protokolliert wurde. Die Einschränkung bei diesem Ansatz liegt dabei, Einzelteile einer Maschine zu beobachten, welche sich bewegen beziehungsweise hohen mechanischen Kräften ausgesetzt sind und durch diese Bewegung verschleifen beziehungsweise durch die Kräfte ermüden. Dies stellt den Überwachungsanteil des Systems dar. Wartungsarbeiten werden vorgenommen, wenn das Überwachungssystem eine Wartungsfirma über das Internet über einen baldigen Ausfall informiert. Dies ist ein Wartungssystem, welches gezielt auf eine Aufgabe zugeschnitten ist. Unser Wartungssystem soll aber allgemeiner und flexibel einsetzbar sein. Außerdem ist unser Ansatz intuitiver und näher an der menschlichen Ausdrucksweise. Durch Verwendung von Fuzzy-Logik ist er mathematisch formalisiert und dadurch nachvollziehbar. Im Folgenden werden vier Ansätze vorgestellt, welche den Begriff der Zeit in Fuzzy-Logik einbringen.

Unter anderem in [Fick00] werden Takagi Sugeno Regeln [Takagi85] in einem Fuzzy-Regler zeitlich abhängig gemacht, indem die Zeit t als weitere Eingabevariable in den Bedingungsteil der Regeln mit aufgenommen wird. Zum Beispiel „**IF** *expression* **AND** $t_0 - t$ **IS** *now* **THEN** *action*“. Dies hat zur Folge, dass Regeln nur zu gewissen Zeiten feuern oder wie bei [Fick00] neu dargestellt, jede Bedingung einer Regel mit einem Zeitintervall konjunktiv verknüpft werden muss. Die Regelaktivierung wird um so höher sein, je näher der aktuelle Zeitpunkt am gegebenen Zeitpunkt liegt. Solch ein Konstrukt ist dann nützlich, wenn Regeln nur zu einer bestimmten Tageszeit feuern dürfen. Zum Beispiel „ $(t_0 - t)$ modulo Tag **IS** *Mittagszeit*“. In diesem Ansatz ist die Zeit zwar unscharf, aber es gibt keine Möglichkeit, Ereignisse untereinander auf zeitliche Abhängigkeiten zu vergleichen, da nur eine Zeitangabe pro Regel vorgesehen ist. Diese Vergleichbarkeit wird jedoch von uns für ein Wartungssystem gefordert.

In [Bovenkamp97] wird ein neuer Ansatz zum temporalen Schließen mit Fuzzy-Logik vorgestellt. So genannte Fuzzy-Zeit-Objekte werden definiert, um Unschärfe in Fakten und Zeit zu repräsentieren. Fuzzy-Zeit-Objekte sind zweistellige, einwertige Zugehörigkeitsfunktionen der Form $\mu_{f \times z}(x, t) := \min(\mu_f(x), \mu_z(t))$. Für diese Zugehörigkeitsfunktionen muss für den Fakt $\max_x \mu_f(x) = 1$ und die Zeit $\max_t \mu_z(t) = 1$ gelten.

Damit gilt auch die Separierbarkeit $\forall x \forall t \in \{t | \max_t \mu_z(t) = 1\} : \mu_f(x) = \mu_{f \times z}(x, t)$ beziehungsweise $\forall t \forall x \in \{x | \max_x \mu_f(x) = 1\} : \mu_z(t) = \mu_{f \times z}(x, t)$. Man kann also ein

Fuzzy-Zeit-Objekt genau dann wenn die Separierbarkeit gilt in Fakt und Zeit zerlegen, so dass man die ursprünglichen Fuzzy Terme $\mu_f(x)$ beziehungsweise $\mu_z(t)$ wieder erhält, ohne dass die Zeit die Unschärfe des Fakts beeinflusst. Die Beziehung zwischen Fakt und Zeit wird durch das temporales Schließen gebildet. Auch bei zeitlich oder faktisch eingeschränkten Fuzzy-Zeit-Objekten gilt dieser Zusammenhang, wobei Fuzzy-Zeit-Objekte nicht gemischt werden dürfen. Die Fuzzy-Zeit-Objekte sind nicht intuitiv verwendbar und erlauben keine UND beziehungsweise ODER Verknüpfungen von Fuzzy-Zeit-Objekten, aber es ist möglich ein Fuzzy-Zeit-Objekt in einer Konklusion zu verwenden. Dieser Ansatz reicht noch nicht ganz um unsere geforderte Mächtigkeit, beliebige Verwendung von zeitlichen Abhängigkeiten in Bedingung und Konklusion einer Fuzzy-Regel, zu erfüllen.

In [Lamine01] wird die Lineare Temporale Logik (LTL) genutzt um zeitliche Abhängigkeiten in Programme einzubringen. Die Syntax der LTL ist ähnlich zu dem der Programmiersprache C. Die LTL ist unscharf in der Auswertung von Bedingungen, denn es wird mit Wahrscheinlichkeiten beziehungsweise Wahrscheinlichkeitsverteilungen für ihr Eintreten gerechnet. Aber es werden keine natürlichsprachlichen Konstrukte verwendet (auch keine Fuzzy-Inferenz). Somit steht die LTL also zwischen der Temporalen Logik und der Fuzzy-Logik. Die Zeit in einer Bedingung ist durch das Prädikat **always** mit dem Zeitintervall $[0, ?]$ gegeben. Das angegebene Zeitintervall steht für „von Jetzt (0) bis in alle Ewigkeit (?)“. Das vorgestellte Anwendungsgebiet, ist ein Roboter, welcher auf einem Straßennetz fährt, wobei die Wahrscheinlichkeitsverteilungen für den weiteren Straßenverlauf bekannt sind. Die einzelnen Regeln werden dann dazu genutzt, um in einer Regelschleife die Programmteile zu aktivieren, welche bei den aktuell vorliegenden und vermuteten zukünftigen Bedingungen mit hoher Wahrscheinlichkeit das gewünschte Ergebnis liefern. Es sind UND beziehungsweise ODER Verknüpfungen von Zeitintervallen möglich, aber es wird hierzu keine Fuzzy-Logik beziehungsweise eine Inferenz darauf verwendet, da die Zeitintervalle nur darüber entscheiden, ob eine eine Regel überhaupt in Betracht gezogen wird.

In [Cárdenas02] wird die Fuzzy Temporal Constraint Logic (FTCL), eine Erweiterung von Prolog mit Horn-Klauseln um Fuzzy-Logik und temporale Prädikate, präsentiert. Die temporalen Prädikate **before**, **after** und **at_the_same_time** ermöglichen es, zeitliche Abhängigkeiten von Ereignissen untereinander auszudrücken. Die Zeit ist dabei als unscharf anzusehen. Außerdem sollen alle Ereignisse, welche zu einer gegebenen Zeit eintreten, nur mit Ereignissen im gleichen Zeitintervall verglichen werden, da Relationen zu unterschiedlichen Zeiten nicht möglich sind. Ein Beispiel, welches zeigt, dass der in [Cárdenas02] vorgestellte Ansatz nicht für ein Wartungssystem verwendet werden kann, ist ein Ereignis, welches kausal von einem anderen Ereignis abhängt, wie zum Beispiel: „Gestern war es heiß und heute leckt das Rohr, dann existiert heute ein Folgeschaden durch die Hitzeinwirkung“.

Wie an den gezeigten Beispielen zu sehen ist, gehen die verwandten Arbeiten in die von uns angestrebte Richtung, Zeit beziehungsweise zeitliche Abhängigkeiten von Bedingungen in Fuzzy-Regler einzubinden. Jedoch gibt es noch keine Lösung, welche dies anhand von der schon ausgereiften Temporalen Logik gemacht hat. Auch wurde noch kein Fuzzy-Regler zur Wartung verwendet. Dies ist durch [Giron02] ersichtlich, wo mehr als 200 Fuzzy-Regler in mehr als 20 Kategorien klassifiziert werden. Ein Fuzzy-Regler mit Temporaler Fuzzy-Logik oder ein Fuzzy-Regler als Wartungssystem existieren demnach noch nicht. Diese Entwicklung ist unser Ziel, welches wir schrittweise erreichen möchten. Ein Schritt für diese Entwicklung ist die hier vorgestellte Erweiterung der Fuzzy-Logik zur Temporalen Fuzzy-Logik.

3. Abkürzungen und Definitionen

- t_A Zeitpunkt in der Vergangenheit, bis zu welchem Aufzeichnungen von Sensordaten eines Prozesses vorhanden sind. Sensordaten vor diesem Zeitpunkt sind nicht bekannt.
- t_o Zeitpunkt in der Zukunft, bis zu welchem Aufzeichnungen von Sensordaten eines Prozesses vorhanden sind, oder bis zu welchem Zeitpunkt eine Vorhersage noch Sinn macht, denn Vorhersagen werden immer ungenauer, je weiter in die Zukunft geschaut wird.
- t_c Aktueller Zeitpunkt eines Prozesses $t_c \in]t_A, t_o[$. Es existiert immer ein Teilintervall mit Sensordaten des Prozesses für die Vergangenheit $[t_A, t_c]$ und für die Zukunft $]t_c, t_o]$.
- t_i Ein beliebiger Zeitpunkt aus dem Intervall $t_i \in [t_A, t_o]$, welcher wenn nicht anders genannt sowohl in der Zukunft, als auch in der Vergangenheit beziehungsweise Gegenwart liegen kann.
- T_i Zeitintervall von t_c bis t_i , wobei t_c größer oder kleiner als t_i sein kann. T_i ist demnach definiert als:
$$T_i := \begin{cases} [t_c, t_i] & , t_c < t_i \\ [t_i, t_c] & , t_i > t_c \end{cases}$$
- $h^i(t)$ Datum h , welches vom Sensor S^i zum Zeitpunkt $t \leq t_c$ aufgezeichnet wurde.
- $p^i(t)$ Datum p , welches für den Sensor S^i für den Zeitpunkt $t \geq t_c$ vorhergesagt wird.
- $S^i(t)$ Sensorwert des i -ten Sensors zum Zeitpunkt t des Prozesses. Liegt der Zeitpunkt $t > t_c$ in der Zukunft, so wird eine Vorhersage zurückgeliefert. Liegt er jedoch in der Vergangenheit $t < t_c$ oder Gegenwart $t = t_c$, dann wird ein aufgezeichneter

Wert zurückgeliefert. $S^i(t)$ ist definiert als:

$$S^i(t) := \begin{cases} h^i(t) & , t \leq t_C \\ p^i(t) & , t > t_C \end{cases}$$

A^i Aktuator i des Prozesses, durch welchen der Regler mit seinen Ausgaben Einfluss auf den Prozess nimmt.

ft Fuzzy-Term $ft = \{(x_i, y_i) | x_i < x_j, 0 \leq y_i \leq 1, i < j, 0 \leq i, j < n\}$, welcher durch die Punkte (x_i, y_i) eines Polygonzugs der Länge n dargestellt wird. In den meisten Fällen wird $n = 3$ und $y_0 = y_{n-1} = 0$ gelten und so, mit dem Fuzzy-Term, ein gleichschenkeliges Dreieck dargestellt. Da ein Fuzzy-Term beliebig viele Stützstellen n besitzen kann, können beliebige stetige Funktionen durch lineare Interpolation approximiert werden.

$\mu_{ft}(x)$ Ist eine Zugehörigkeitsfunktion für den Fuzzy-Term ft , für welche die Bedingungen nach [Bothe95] gelten:

- $\forall x \in \mathbb{R} : \mu_{ft}(x) \geq 0$
- $\mu_{ft}(x)$ umso größer, je besser x ein Bewertungskriterium eines Experten erfüllt
- Einheitsintervall-Normalisierung, so dass für $\mu(x) : x \rightarrow [0,1]$
- Normalisierung auf ein Bezugselement x_0 mit $\mu_{ft}(x_0) = 1$

Während die ersten beiden Bedingungen harte Bedingungen sind, welche in jedem Fall erfüllt sein müssen, sind die dritte und vierte Bedingung weiche Bedingungen, welche nicht unbedingt erfüllt sein müssen. Sind sie nicht erfüllt, so sprechen wir von einem unpräzisen, ungenauen oder auch unsicherem Fuzzy-Term, ansonsten von einem präzisen, genauen, oder sicheren Fuzzy-Term. Wenn nicht anders genannt, ist mit einem Fuzzy-Term immer ein präziser Fuzzy-Term gemeint. Die Einheitsintervall-Normalisierung impliziert nicht notwendigerweise die Normalisierung auf ein Bezugselement, denn die Zugehörigkeitsfunktionen müssen keine surjektiven Abbildungen sein.

Zugehörigkeitsfunktionen sind hier stetige, stückweise lineare, nicht notwendigerweise monotone Funktionen deren Stützstellen durch den Fuzzy-Term ft gegeben sind. Sie sind im gesamten Definitionsbereich definiert und es gilt folgende Definition:

$$\mu_{ft}(x) = \begin{cases} y_i + (y_{i+1} - y_i) \cdot \frac{(x - x_i)}{(x_{i+1} - x_i)} & , x_i \leq x < x_{i+1} \\ y_0 & , x < x_0 \\ y_{n-1} & , x \geq x_{n-1} \end{cases}$$

Alternativ kann auch eine höherdimensionale Interpolation der Stützstellen verwendet werden. Wir beschränken uns aber bewusst auf stückweise lineare Funktionen, denn mit diesen berechnet sich die Vereinigung von Fuzzy-Termen, zum Beispiel $ft_A \cup ft_B$, und die Akkumulierung mit der Schwerpunktmethode nach [Watanabe86] (Berechnung von Integralen) sehr schnell und effizient und führt so zu effizienteren und schneller berechenbaren Fuzzy-Reglern.

Man spricht bei $\mu_{ft}(x)$ von der Zugehörigkeit von x zum Fuzzy-Term ft oder von der Aktivierung des Fuzzy-Terms ft durch x .

4. Temporale Fuzzy-Logik

Betrachtet man die Logik mit der Unterteilung, wie in Tabelle 1 dargestellt, in atemporale und temporale Logik auf der Zeitachse beziehungsweise in scharfe und unscharfe Logik auf der Schärfeachse, so repräsentiert die Prädikatenlogik, welche den Wertebereich $\{0, 1\}$ besitzt und zeitlich konstant ist, die Gruppe der scharfen, atemporalen Logik. Die Prädikatenlogik kann nun zum Einen zeitlich erweitert werden oder zum Anderen unscharf gemacht werden.

Die zeitliche Erweiterung ist für die Modellierung von dynamischen Systemen wie Zustandsautomaten, physikalischen Prozessen und ähnlichem nötig. Also erweitert man die Prädikatenlogik um Operatoren, welche es erlauben zeitliche Abhängigkeiten zu beschreiben. Diese temporalen Operatoren beschreibt [Karjoth87] für Zustandsautomaten beziehungsweise für Prozesse mit zeitlicher Diskretisierung. Zu den Operatoren werden Kalküle und Herleitungen (Beweise), welche in der Prädikatenlogik gültig sind, so angepasst, dass sie auch in der Temporalen Logik gültig sind. Somit ist die Prädikatenlogik ein Spezialfall der Temporalen Logik ohne Verwendung der temporalen Operatoren. Die temporalen Operatoren sind unten links in Tabelle 1 erläutert.

Die unscharfe Erweiterung der Prädikatenlogik kommt Anwendungsgebieten mit ungenauem oder unscharfem Wissen zu gute. Denn bei diesen Anwendungen existiert Wissen von einem oder mehreren Experten, welches in der Regel unscharf formuliert ist. Da die Fuzzy-Logik dazu gedacht ist unscharfes Wissen darzustellen und zu verarbeiten, kann das unscharfe Expertenwissen einfacher in Fuzzy-Logik als in Prädikatenlogik ausgedrückt werden. Die Fuzzy-Logik erweitert wie die Temporale Logik ebenfalls die Prädikatenlogik, führt aber keine neuen Operatoren ein, sondern weicht den Wertebereich von $\{0, 1\}$ beziehungsweise {falsch, wahr} zum Intervall $[0,1]$ auf. Wobei $x = 0$ falsch und $x = 1$ wahr entspricht. Dass dadurch ebenfalls alle Beweise und Kalküle aus der Prädikatenlogik gelten wurde schon mehrfach in der Literatur gezeigt [Bothe95], [Karjoth87]. Die Prädikatenlogik ist demnach auch ein Spezialfall der Fuzzy-Logik.

Um nun die Temporale Fuzzy-Logik, welche einerseits unscharf und andererseits temporal ist, zu erhalten gibt es zwei Möglichkeiten. Entweder wird der Wertebereich der Temporalen Logik unscharf gemacht, oder die Fuzzy-Logik erhält wie die Temporale Logik temporale Operatoren, um zeitliche Abhängigkeiten zu beschreiben. Vergleicht man in Tabelle 1 die Temporale Logik mit der Temporalen Fuzzy-Logik, so sieht man, dass für jeden temporalen Operator ein Fuzzy-Prädikat existiert. Ganz offensichtlich ist die Temporale Fuzzy-Logik (TFL) eine Mischung der Temporalen Logik (TL) und Fuzzy-Logik (FL), da von beiden die Erweiterungen zur Prädikaten Logik (PL) eingeflossen sind. So erhält man durch Weglassen der Unschärfe von der TFL die TL und durch Weglassen der temporalen Prädikate von der TFL die FL.

Die Darstellung in der Tabelle lässt durch die Prädikate **WILL_EXIST_NEXT** und **PREEXIST_PREVIOUS** den Schluss zu, dass die Temporale Fuzzy-Logik wie die Temporale Logik nur für zeitlich diskrete Prozesse geeignet ist. Dass dem nicht so ist, wird im nächsten Abschnitt erläutert. Dort wird gezeigt, wie die unscharfen temporalen Prädikate nach dem menschlichen Empfinden intuitiv modelliert sind und dass sie auch zum Beschreiben für zeitlich kontinuierliche Prozesse eingesetzt werden können.

	Scharf	Unscharf
Atemporal	<u>Prädikatenlogik (PL)</u> $P(x) \rightarrow \{0,1\}$ $P(x)$: x ist wahr	<u>Fuzzy-Logik (FL)</u> $P(x, ft) \rightarrow [0,1]$ x IS ft : x gehört zu μ_{ft}
	<u>Temporale Logik (TL)</u> $P(x) \rightarrow \{0,1\}$ $P(x, y) \rightarrow \{0,1\}$ $\Box x$ x wird immer wahr sein $\Box x$ x war immer wahr $\Diamond x$ x wird mindestens einmal wahr sein $\Diamond x$ x war mindestens einmal wahr Ox x wird zum nächsten Zeitpunkt wahr Θx x war zum letzten Zeitpunkt wahr $x \mathcal{U} y$ x bleibt wahr bis y wahr wird $x \mathcal{S} y$ x gilt seit y zum letzten mal wahr geworden ist	<u>Temporale Fuzzy-Logik (TFL)</u> Einfach: $P(x, ft, t_i, t_j) \rightarrow [0,1]$ Komplex: $P(x, y, ft_A, ft_B) \rightarrow [0,1]$ x WILL_BE ft x wird immer zu μ_{ft} gehören $\forall t > t_C: \mu_{ft}(S^i(t)) > 0$ x WAS ft x gehörte immer zu μ_{ft} $\forall t < t_C: \mu_{ft}(S^i(t)) > 0$ x WILL_EXIST ft x wird min. einmal zu μ_{ft} gehören $\exists t > t_C: \mu_{ft}(S^i(t)) > 0$ x PREEXIST ft x gehörte min. einmal zu μ_{ft} $\exists t < t_C: \mu_{ft}(S^i(t)) > 0$ x WILL_EXIST_NEXT ft x gehört im nächsten Zeitpunkt zu μ_{ft} $\mu_{ft}(S^i(t+\Delta t)) > 0$ x PREEXIST_PREV. ft x gehörte im letzten Zeitpunkt zu μ_{ft} $\mu_{ft}(S^i(t-\Delta t)) > 0$ x IS ft_A UNTIL y IS ft_B x gehört zu μ_{ft} bis y zu μ_{ft} gehört x IS ft_A SINCE y IS ft_B x gehört zu μ_{ft_A} seit y mal zu μ_{ft_B} gehörte

Tabelle 1: Einordnung der Temporalen Fuzzy-Logik in Bezug zur scharfen/unscharfen beziehungsweise atemporalen/temporalen Logik.

5. Temporale Fuzzy-Prädikate

Prädikate bestimmen den Zugehörigkeitsgrad von Sensordaten zu gegebenen Fuzzy-Termen. Die Anzahl der Sensoren beziehungsweise Sensordaten, welche von einem Prädikat betrachtet werden ist n . *Einfache* temporale Prädikate (**WILL_BE**, **WAS**, ...) besitzen einen Fuzzy-Term und sind $(n+3)$ -stellige Abbildungen $P(\vec{x}, ft, t_i, t_j) \rightarrow [0,1]$, wenn ein Zeitintervall gegeben ist beziehungsweise $(n+1)$ -stellige Abbildungen $P(\vec{x}, ft) \rightarrow [0,1]$, wenn kein Zeitintervall gegeben ist. *Komplexe* temporale Prädikate (**SINCE**, **UNTIL**) besitzen zwei Fuzzy-Terme und sind $(2n+2)$ -stellig Abbildungen $P(\vec{x}, \vec{y}, ft_A, ft_B) \rightarrow [0,1]$. Für alle Prädikate müssen folgende Bedingungen erfüllt sein:

- Normierung: $\forall \vec{x}, ft: 0 \leq P(\vec{x}, ft) \leq 1$ (1)
- Stetigkeit: für beliebige, aber feste ft gilt:
 $\forall \vec{x}_0 \forall \epsilon > 0 \exists |\vec{\delta}| > 0: \forall \vec{x} \text{ mit } |\vec{x} - \vec{x}_0| < |\vec{\delta}| \text{ folgt } |P(\vec{x}, ft) - P(\vec{x}_0, ft)| < \epsilon$ (2)
- Komplement (Standardnegation): $\neg P$ ist definiert durch: $\neg P(\vec{x}, ft) = 1 - P(\vec{x}, ft)$ (3)

Definition: Prädikat IS

$$S^i \text{ IS } ft := P_v(i, t_C, ft) := \mu_{ft}(S^i(t_C))$$

Erläuterung: Die Prädikatenfunktion P_v steht für *now* und das Prädikat **IS** entspricht dem schon bekannten Prädikat **IS** der Fuzzy-Logik. Die Semantik ist, dass je genauer der zuletzt aufgezeichnete, also aktuelle Sensorwert $S^i(t_C)$ des i -ten Sensors im gegebenen Fuzzy-Term ft liegt, desto höher ist dessen Aktivierungsgrad, welcher durch die

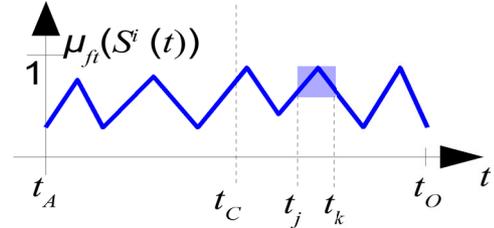
Zugehörigkeitsfunktion μ_{ft} bestimmt ist. Durch die Einschränkung auf präzise Fuzzy-Terme (siehe Definition von μ_{ft} in Kapitel 3) ist die Zugehörigkeitsfunktion und somit auch das Prädikat normiert (1). Die Zugehörigkeitsfunktion liefert einen Wert zwischen 0 (Datum aktiviert nicht den Fuzzy-Term) und 1 (Datum aktiviert den Fuzzy-Term) und entspricht dem Aktivierungsgrad des Fuzzy-Terms ft . Der Aktivierungsgrad entspricht in diesem Fall auch dem Wahrheitsgehalt des Prädikates. Des weiteren ist das Prädikat eine stetige Funktion, denn die Zugehörigkeitsfunktion ist als stückweise lineare und somit stetige Funktion definiert (2). Für die Negation des Prädikates gilt (3):

$$\begin{aligned} \neg P_v(i, t_C, ft) &= \neg \mu_{ft}(S^i(t_C)) = \mu_{\neg ft}(S^i(t_C)) \\ &= \mu_{\{(x_i, y_i) | x_i \leq x_j, 0 \leq y_i \leq 1, i < j, 0 \leq i, j < n\}}(S^i(t_C)) \\ &= \mu_{\{(x_i, 1-y_i) | x_i \leq x_j, 0 \leq y_i \leq 1, i < j, 0 \leq i, j < n\}}(S^i(t_C)) \\ &= 1 - \mu_{ft}(S^i(t_C)) = 1 - P_v(i, t_C, ft) \end{aligned}$$

Definition: Prädikat WILL BE

$$S^i \text{ WILL_BE}[t_j, t_k] ft := P_\Phi(S^i, t_j, t_k, ft) := \frac{1}{t_k - t_j} \int_{t_j}^{t_k} P_v(i, t, ft) dt$$

Erläuterung: Die Prädikatenfunktion P_Φ steht für *future-whole*. Das Prädikat **WILL_BE** berechnet die Aktivierung des Fuzzy-Terms ft für Sensordaten aus dem Zeitintervall $[t_j, t_k]$. **WILL_BE** ist umso aktiver, je genauer die vorhergesagten Sensordaten $p^i(t)$ mit $t \in [t_j, t_k]$ im gegebenen Fuzzy-Term ft liegen. Streifen die



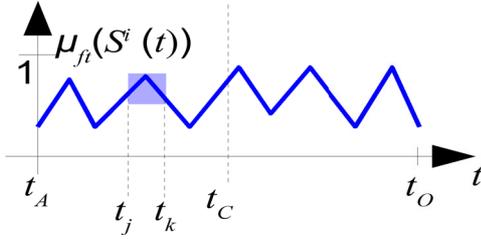
Sensordaten den Fuzzy-Term nur kurz, so ist die Aktivierung geringer, als wenn die Sensordaten während der gesamten Intervalldauer im Fuzzy-Term liegen würden. Das gegebene Zeitintervall liegt dabei vollständig in der Zukunft, demnach gilt $t_j > t_C$. Die Aktivierung wird durch das Mittel über alle Sensorwerte im gegebenen Zeitintervall mit dem Prädikat **IS** gebildet. Da P_v normiert ist, also immer kleiner gleich 1 und größer 0 ist, ist das Integral maximal $t_k - t_j$. Das Integral wird durch $t_k - t_j$ geteilt und ist so normiert auf das Intervall $[0,1]$ (1). Das Prädikat P_Φ ist stetig, denn P_v ist auch stetig und das Integral einer stetigen Funktion ist wiederum stetig. Der konstante Vorfaktor $1/(t_k - t_j)$ hat auch keinen Einfluss auf die Stetigkeit (2). Für die Negation des Prädikates gilt (3):

$$\begin{aligned} \neg P_\Phi(S^i, t_j, t_k, ft) &= \frac{1}{t_k - t_j} \int_{t_j}^{t_k} \neg P_v(i, t, ft) dt \\ &= \frac{1}{t_k - t_j} \int_{t_j}^{t_k} 1 - P_v(i, t, ft) dt = 1 - \frac{1}{t_k - t_j} \int_{t_j}^{t_k} P_v(i, t, ft) dt \\ &= 1 - P_\Phi(S^i, t_j, t_k, ft) \end{aligned}$$

Definition: Prädikat WAS

$$S^i \text{ WAS}[t_j, t_k] ft := P_\Pi(S^i, t_j, t_k, ft) := \frac{1}{t_k - t_j} \int_{t_j}^{t_k} P_v(i, t, ft) dt$$

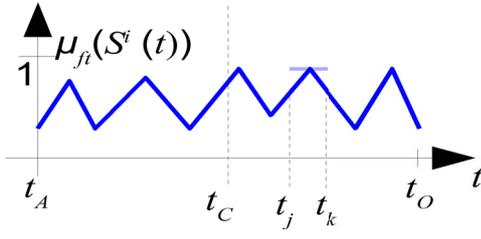
Erläuterung: Die Prädikatenfunktion P_{Π} steht für *past-whole*. Das Prädikat **WAS** berechnet die Aktivierung des Fuzzy-Terms ft für Sensordaten aus dem Zeitintervall $[t_j, t_k]$. **WAS** ist umso aktiver, je genauer die vergangenen Sensordaten $h^i(t)$ mit $t \in [t_j, t_k]$ im gegebenen Fuzzy-Term ft liegen. Streifen die Sensordaten das Intervall nur kurz, so ist die Aktivierung geringer, als wenn die Sensordaten während der gesamten Intervalldauer im Fuzzy-Term liegen würden. Das gegebene Zeitintervall liegt dabei vollständig in der Vergangenheit, demnach gilt $t_k < t_c$. Die Aktivierung wird durch das Mittel über alle Sensorwerte im gegebenen Zeitintervall mit dem Prädikat **IS** gebildet. Die Bedingungen (1) – (3) gelten für das Prädikat P_{Π} analog zu P_{Φ} mit den selben Begründungen.



Definition: Prädikat WILL EXIST

$$S^i \text{ WILL_EXIST}[t_j, t_k] ft := P_{\varphi}(S^i, t_j, t_k, ft) := \max_{t \in [t_k, t_k]} P_v(i, t, ft)$$

Erläuterung: Die Prädikatenfunktion P_{Φ} steht für *future-single*. Das Prädikat **WILL_EXIST** bewertet, ähnlich wie das Prädikat **WILL_BE**, die Aktivierung des Fuzzy-Terms ft für Sensordaten aus dem Zeitintervall $[t_j, t_k]$. **WILL_EXIST** prüft jedoch, ob überhaupt ein vorhergesagter Sensorwert $p^i(t)$ mit $t \in [t_j, t_k]$ im gegebenen Fuzzy-Term ft liegt. Dabei ist es für die Aktivierung des Fuzzy-Terms egal, ob ein oder mehrere Sensorwerte im Fuzzy-Term ft liegen. Für die Aktivierung zählt lediglich der Sensorwert mit der maximalen Aktivierung. Die Aktivierung wird über die Maximumsuche aller Aktivierungen von Sensorwerten im gegebenen Zeitintervall mit dem Prädikat **IS** gebildet. Da P_v normiert ist, ist auch das Maximum von P_v normiert (1). Die Komposition von stetigen Funktionen mit der Maximumfunktion liefert immer stetige Funktionen, und da P_v stetig ist, ist somit auch P_{Φ} stetig (2). Für die Negation des Prädikates gilt (3):

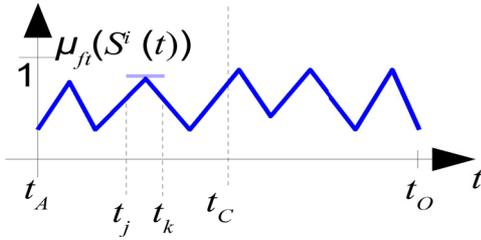


$$\begin{aligned} \neg P_{\varphi}(S^i, t_j, t_k, ft) &= \max_{t \in [t_k, t_k]} \neg P_v(i, t, ft) \\ &= \max_{t \in [t_k, t_k]} 1 - P_v(i, t, ft) = 1 - \max_{t \in [t_k, t_k]} P_v(i, t, ft) \\ &= 1 - P_{\varphi}(S^i, t_j, t_k, ft) \end{aligned}$$

Definition: Prädikat PREEXIST

$$S^i \text{ PREEXIST}[t_j, t_k] ft := P_{\pi}(S^i, t_j, t_k, ft) := \max_{t \in [t_k, t_k]} P_v(i, t, ft)$$

Erläuterung: Die Prädikatenfunktion P_{π} steht für *past-single*. Das Prädikat **PREEXIST** bewertet, ähnlich wie das Prädikat **WAS**, die Aktivierung des Fuzzy-Terms ft für Sensordaten aus dem Zeitintervall $[t_j, t_k]$. **PREEXIST** jedoch prüft, ob überhaupt ein vergangener Sensorwert $h^i(t)$ mit $t \in [t_j, t_k]$ im gegebenen Fuzzy-Term ft liegt. Dabei ist es für die Aktivierung des Fuzzy-Terms egal, ob ein oder mehrere Sensorwerte im Fuzzy-Term ft liegen. Für die Aktivierung zählt lediglich der Sensorwert mit



der maximalen Aktivierung. Die Aktivierung wird über die Maximumsuche aller Aktivierungen von Sensorwerten im gegebenen Zeitintervall mit dem Prädikat **IS** gebildet. Die Bedingungen (1) – (3) gelten für das Prädikat P_π analog zu P_φ mit den selben Begründungen.

Definition: Prädikat WILL_EXIST_NEXT

$$S^i \text{ WILL_EXIST_NEXT } ft \equiv S^i \text{ WILL_EXIST}[t_C, t_C + \Delta t] ft \equiv P_\varphi(S^i, t_C, t_C + \Delta t, ft)$$

Erläuterung: Das Prädikat **WILL_EXIST_NEXT** ist ein Makro für das Prädikat **WILL_EXIST** und entstammt aus der direkten Übersetzung des temporalen Operators Ox (wird zum nächsten Zeitpunkt beziehungsweise im nächsten Zustand wahr, siehe Tabelle 1, Kapitel 4) und kann in der Temporalen Fuzzy-Logik mit kontinuierlicher und diskreter Zeit bei einer geeigneten Wahl von Δt verwendet werden. Bei der Auswertung des Prädikates interessiert nur ein kleiner Verlauf Δt in der Zukunft. Es ist zu beachten, dass Δt so gewählt werden muss, dass immer $\Delta t \leq t_o - t_C$ gilt. Bei Prozessen, welche mit einer festen Frequenz f getaktet sind und bei jedem Takt ihren Zustand ändern können, wird $\Delta t = 1/f$ auf die Dauer eines Taktes gesetzt. Da **WILL_EXIST_NEXT** nur ein Makro für **WILL_EXIST** ist, gelten auch hier die schon gezeigten Bedingungen (1) – (3).

Definition: Prädikat PREEEXIST_PREVIOUS

$$S^i \text{ PREEEXIST_PREVIOUS } ft \equiv S^i \text{ PREEEXIST}[t_C - \Delta t, t_C] ft \equiv P_\pi(S^i, t_C - \Delta t, t_C, ft)$$

Erläuterung: Das Prädikat **PREEEXIST_PREVIOUS** ist ein Makro für das Prädikat **PREEEXIST** (siehe Abschnitt 4.5) und entstammt aus der direkten Übersetzung des Temporalen Operators Θx (war zum letzten Zeitpunkt beziehungsweise im letzten Zustand wahr, siehe Tabelle 1, Kapitel 4) und kann in der Temporalen Fuzzy-Logik mit kontinuierlicher und diskreter Zeit bei einer geeigneten Wahl von Δt verwendet werden. Bei der Auswertung des Prädikates interessiert nur ein kleiner Verlauf Δt in der Vergangenheit. Es ist zu beachten, dass Δt so gewählt werden muss, dass immer $\Delta t \leq t_C - t_A$ gilt. Bei Prozessen, welche mit einer festen Frequenz f getaktet sind und bei jedem Takt ihren Zustand ändern können, wird $\Delta t = 1/f$ auf die Dauer eines Taktes gesetzt. Da **PREEEXIST_PREVIOUS** nur ein Makro für **PREEEXIST** ist, gelten auch hier die schon gezeigten Bedingungen (1) – (3).

Definition: Prädikat UNTIL

$$S^i \text{ IS } ft_A \text{ UNTIL } [t_j, t_k, t_l] S^h \text{ IS } ft_B := P_{\text{UNTIL}}(S^i, S^h, t_j, t_k, t_l, ft_A, ft_B)$$

mit:

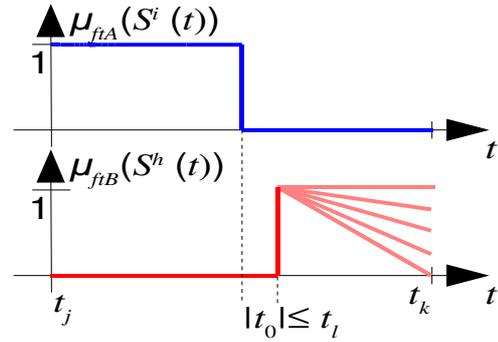
$$P_{\text{reset}}(t_0, S^i, t_j, t_k, ft_A) = \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} P_v(i, t, ft_A) dt + \int_{t_0}^{t_k} 1 - P_v(i, t, ft_A) dt \right)$$

$$P_{\text{edge}}(t_0, S^h, t_j, t_k, ft_B) = \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 - P_v(h, t, ft_B) dt + (t_k - t_0) \cdot P_v(h, t_0, ft_B) \right)$$

$$t_0 \in \{t_0 \mid P_{\text{reset}}(t_0) = \max_t P_{\text{reset}}(t), t_0 \in [t_j, t_k]\}$$

$$P_{\text{UNTIL}}(S^i, S^h, t_j, t_k, t_l, ft_A, ft_B) = \min \left(P_{\text{reset}}(t_0, \dots), \max_{|t_0 - t_m| \leq t_l} P_{\text{edge}}(t_m, \dots) \right)$$

Erläuterung: Das Prädikat **UNTIL** testet mit dem Prädikat P_{reset} im Zeitintervall t_j bis t_k , ob ein Signalverlauf S^i , welcher zu Anfangs gültig war (in einer Zugehörigkeitsfunktion μ_{ftA} lag) durch das gültig werden des Signalverlaufes S^h (getestet mit P_{edge}) ungültig wurde und ungültig bleibt. Dabei ist bei dem Signalverlauf von S^h nur wichtig, dass er ungefähr zur selben Zeit ($\pm t_l/2$) gültig wird, wie S^i ungültig wird. Der weitere Verlauf von S^h spielt für den Aktivierungsgrad des Prädikates keine Rolle. Um P_{UNTIL} aus P_{reset} und P_{edge} zu berechnen kann entweder das Minimum oder das Produkt der beiden Prädikate genutzt werden. Gehen wir davon aus, dass die Zugehörigkeitsfunktionen Dreiecksfunktionen mit den Fuzzy Termen $ft_A = ft_B = \{(0,0), (0.5,1), (1,0)\}$ sind und die Zugehörigkeitsfunktionen demnach wie folgt definiert sind:



$$\mu_{ftA}(x) = \mu_{ftB}(x) = \begin{cases} 2x, & 0 < x \leq 0.5 \\ 1 - 2x, & 0.5 < x < 1 \\ 0, & \text{sonst} \end{cases}$$

Das Integral über die zweistelligen Funktionen für das Produkt $\mu_{ftA}(x) \cdot \mu_{ftB}(y)$ beziehungsweise für das Minimum $\min(\mu_{ftA}(x), \mu_{ftB}(y))$ der beiden Zugehörigkeitsfunktionen ist $1/4$ beziehungsweise $1/3$. Berechnung der Integrale:

$$\begin{aligned} P_{Produkt} &= \int_0^1 \int_0^1 \mu_{ftA}(x) \cdot \mu_{ftB}(y) dy dx = 4 \cdot \int_0^{1/2} \int_0^{1/2} 2x \cdot 2y dy dx \\ &= 4 \cdot \int_0^{1/2} 2x \cdot 2y \Big|_0^{1/2} dx = \int_0^{1/2} 2x dx = 1/4 \end{aligned}$$

$$\begin{aligned} P_{Minimum} &= \int_0^1 \int_0^1 \min(\mu_{ftA}(x), \mu_{ftB}(y)) dy dx \\ &= 4 \cdot \int_0^{1/2} \int_0^{1/2} \min(2x, 2y) dy dx = 8 \cdot \int_0^{1/2} \int_{0, x \geq y}^{1/2} y dy dx + 8 \cdot \int_0^{1/2} \int_{0, y \geq x}^{1/2} x dy dx \\ &= 16 \cdot \int_0^{1/2} \int_{0, x \geq y}^{1/2} y dy dx = 16 \cdot \int_0^{1/2} \int_0^x y dy dx = 8 \cdot \int_0^{1/2} x^2 dx = 1/3 \end{aligned}$$

Für das Prädikat liefert somit das Integral über die Zugehörigkeitsfunktionen mit dem Produkt eine kleinere Aktivierung als das mit dem Minimum. Aus diesem Grund favorisieren wir die Berechnung über das Minimum, da so das Prädikat **UNTIL** eher auf Signalverläufe mit den oben beschriebenen Eigenschaften P_{reset} und P_{edge} reagiert.

Da P_v normiert ist, also immer $0 \leq P_v \leq 1$ gilt, ist das erste Integral von P_{reset} maximal $t_0 - t_j$ und das zweite Integral maximal $t_k - t_0$ groß. Da die Summe der beiden Integrale durch $t_k - t_j$ geteilt wird, ist P_{reset} maximal 1 und somit auch normiert. Für P_{edge} gilt dies auch, denn das erste Integral addiert mit $t_k - t_0$ mal dem normierten Prädikat P_v geteilt durch $t_k - t_j$ ist maximal 1. Da nun P_{edge} und P_{reset} normiert sind, ist dies auch deren Minimum beziehungsweise Produkt (1). Die Komposition von stetigen Funktionen mit der Minimumfunktion liefert immer stetige Funktionen, und da P_{edge} und P_{reset} stetig ist, ist somit auch P_{UNTIL} stetig (2). Für Bedingung (3) soll $\neg P_{UNTIL} = 1 - P_{UNTIL}$ gelten.

Diese Bedingung ist erfüllt, wenn Für $x:=P_{reset}$ und $y:=P_{edge}$ gilt, dass $\neg \min(x, y) = 1 - \min(x, y)$ ist. Der Beweis:

$$\begin{aligned}\neg \min(x, y) &= \neg \begin{cases} x, & x \leq y \\ y, & \text{sonst} \end{cases} = \begin{cases} \neg x, & x \leq y \\ \neg y, & \text{sonst} \end{cases} \\ &= \begin{cases} 1-x, & x \leq y \\ 1-y, & \text{sonst} \end{cases} = 1 - \begin{cases} x, & x \leq y \\ y, & \text{sonst} \end{cases} = 1 - \min(x, y)\end{aligned}$$

Bleibt noch zu zeigen, dass $\neg P_{reset} = 1 - P_{reset}$ und $\neg P_{edge} = 1 - P_{edge}$ gilt.

$$\begin{aligned}& \neg P_{reset}(t_0, S^i, t_j, t_k, ft_A) \\ &= \neg \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} P_v(i, t, ft_A) dt + \int_{t_0}^{t_k} 1 - P_v(i, t, ft_A) dt \right) \\ &= \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} \neg P_v(i, t, ft_A) dt + \int_{t_0}^{t_k} 1 - \neg P_v(i, t, ft_A) dt \right) \\ &= \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 - P_v(i, t, ft_A) dt + \int_{t_0}^{t_k} 1 - (1 - P_v(i, t, ft_A)) dt \right) \\ &= \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 dt - \int_{t_j}^{t_0} P_v(i, t, ft_A) dt + \int_{t_0}^{t_k} 1 dt - \int_{t_0}^{t_k} 1 - P_v(i, t, ft_A) dt \right) \\ &= \frac{t_k - t_j}{t_k - t_j} + \frac{1}{t_k - t_j} \left(- \int_{t_j}^{t_0} P_v(i, t, ft_A) dt - \int_{t_0}^{t_k} 1 - P_v(i, t, ft_A) dt \right) \\ &= 1 - \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} P_v(i, t, ft_A) dt + \int_{t_0}^{t_k} 1 - P_v(i, t, ft_A) dt \right) \\ &= 1 - P_{reset}(t_0, S^i, t_j, t_k, ft_A)\end{aligned}$$

$$\begin{aligned}& \neg P_{edge}(t_0, S^h, t_j, t_k, ft_B) \\ &= \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 - \neg P_v(h, t, ft_B) dt + (t_k - t_0) \cdot \neg P_v(h, t_0, ft_B) \right) \\ &= \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 - (1 - P_v(h, t, ft_B)) dt + (t_k - t_0) \cdot (1 - P_v(h, t_0, ft_B)) \right) \\ &= \frac{1}{t_k - t_j} \left((t_0 - t_j) + (t_k - t_0) + \int_{t_j}^{t_0} -1 + P_v(h, t, ft_B) dt + (t_k - t_0) \cdot (-P_v(h, t_0, ft_B)) \right) \\ &= 1 - \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 - P_v(h, t, ft_B) dt + (t_k - t_0) \cdot P_v(h, t_0, ft_B) \right) \\ &= 1 - P_{edge}(t_0, S^h, t_j, t_k, ft_B)\end{aligned}$$

$\neg P_{edge} = 1 - P_{edge}$ gilt jedes beliebige t_0 , insbesondere auch für $\max_{|t_0 - t_m| \leq t_l} P_{edge}(t_m, \dots)$.

Somit ist Bedingung (3) bewiesen.

Definition: Prädikat SINCE

$$S^i \text{ IS } ft_A \text{ SINCE } [t_j, t_k, t_l] S^h \text{ IS } ft_B := P_{SINCE}(S^i, S^h, t_j, t_k, t_l, ft_A, ft_B)$$

mit:

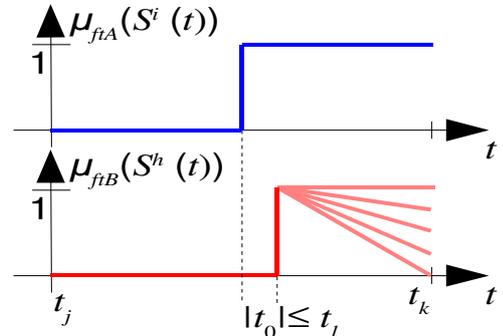
$$\neg P_{reset}(t_0, S^i, t_j, t_k, ft_A) = \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 - P_v(i, t, ft_A) dt + \int_{t_0}^{t_k} P_v(i, t, ft_A) dt \right)$$

$$P_{edge}(t_0, S^h, t_j, t_k, ft_B) = \frac{1}{t_k - t_j} \left(\int_{t_j}^{t_0} 1 - P_v(h, t, ft_B) dt + (t_k - t_0) \cdot P_v(h, t_0, ft_B) \right)$$

$$t_0 \in \{t_0 | \neg P_{reset}(t_0) = \max_t \neg P_{reset}(t), t_0 \in [t_j, t_k]\}$$

$$P_{SINCE}(S^i, S^h, t_j, t_k, t_l, ft_A, ft_B) = \min \left(\neg P_{reset}(t_0, \dots), \max_{|t_0 - t_m| \leq t_l} P_{edge}(t_m, \dots) \right)$$

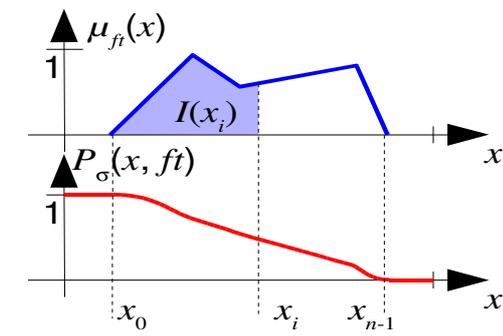
Erläuterung: Das Prädikat **SINCE** testet im Zeitintervall t_j bis t_k , ob ein Signalverlauf S^i , welcher zu Anfangs ungültig war (nicht in einer Zugehörigkeitsfunktion μ_{ft} lag) durch das gültig werden des Signalverlaufes S^h auch gültig wurde und gültig bleibt. Dabei ist bei dem Signalverlauf von S^h nur wichtig, dass er ungefähr zur selben Zeit $\pm t_l/2$ wie S^i gültig wird. Der weitere Verlauf von S^h spielt für den Aktivierungsgrad des Prädikates keine Rolle. Die Bedingungen (1) – (3) gelten für das Prädikat **SINCE** analog zu **UNTIL** mit den selben Begründungen.



Definition: Prädikat SMALLER

$$S^i \text{ SMALLER } ft := P_\sigma(x, ft) = 1 - \int_{-\infty}^x \mu_{ft}(y) dy / \int_{-\infty}^{\infty} \mu_{ft}(y) dy$$

Erläuterung: Die Prädikatenfunktion P_σ steht für *smaller*. Das Prädikat soll entscheiden, ob ein Wert x kleiner ist als ein gegebener Fuzzy-Term ft . Der Fuzzy-Term beginnt bei $(x_0, y_0) \in ft$ und endet bei $(x_{n-1}, y_{n-1}) \in ft$. Ist x echt kleiner als ft , also gilt $x < x_0$, so ist das Prädikat 1. Ist x jedoch echt größer als der ft , also gilt $x > x_{n-1}$, so ist das Prädikat 0. Liegt x jedoch innerhalb von ft , so wird das Prädikat umso kleiner, je näher man sich dem rechten Rand des Fuzzy-Terms nähert.



Zur genaueren Erläuterung definieren wir uns die Funktion $I(x) = \int_{-\infty}^x \mu_{ft}(y) dy$, welche eine Teilfunktion des Prädikates **SMALLER** ist. Sie liefert eine monoton steigende Funktion, denn die Zugehörigkeitsfunktion μ_{ft} ist nie negativ. $I(x)$ wird nun normiert und negiert, indem sie durch $I(\infty)$ geteilt und von 1 abgezogen wird. So entspricht die neu entstehende Funktion dem Prädikat **SMALLER**.

Es ist zu beachten, dass der Vergleich mit nur einem Fuzzy-Term gemacht wird. Es wird also keine Aussage über andere Fuzzy-Terme getroffen. Demnach müssen die Fuzzy-Terme nicht in geordneter Form gegeben sein. Sollten die Fuzzy-Terme jedoch geordnet sein, so lässt sich, wenn S_i **SMALLER** ft wahr ist, genau sagen, für welche Fuzzy-Terme S_i **SMALLER** ft auch noch wahr ist (und zwar für alle Fuzzy-Terme, welche größer als ft sind). Dies kann dazu genutzt werden um Bedingungen zu vereinfachen. Existieren zum Beispiel die Fuzzy-Terme *small*, *medium* und *big* für eine Variable x , so kann die Bedingung „(x **IS** *small*) **AND** (x **IS** *medium*) **AND** (x **IS NOT** *big*)“ zu „(x **SMALLER** *big*)“ abgekürzt werden. In der Regel besitzt eine Variable mehr als drei Fuzzy-Terme, wodurch die Ersparnis umso größer wird.

Für jedes beliebige x gilt immer, dass $0 \leq \int_{-\infty}^x \mu_{ft}(y) dy \leq \int_{-\infty}^{\infty} \mu_{ft}(y) dy$ ist, also ist der

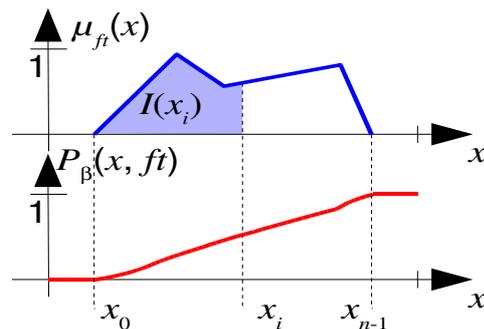
Bruch $\int_{-\infty}^x \mu_{ft}(y) dy / \int_{-\infty}^{\infty} \mu_{ft}(y) dy$ immer kleiner gleich 1 und größer gleich 0. Dies gilt

auch für P_σ welches somit normiert ist (1). Auch gilt wiederum, dass die Komposition von stetigen Funktionen wieder stetige Funktionen liefert (2). Dass Bedingung (3) gilt wird bei der Definition des Prädikates **BIGGER** gezeigt.

Definition: Prädikat **BIGGER**

$$S_i \text{ **BIGGER** } ft := P_\beta(x, ft) = \int_{-\infty}^x \mu_{ft}(y) dy / \int_{-\infty}^{\infty} \mu_{ft}(y) dy$$

Erläuterung: Die Prädikatenfunktion P_β steht für *bigger*. Das Prädikat soll entscheiden, ob ein Wert x größer ist als ein gegebener Fuzzy-Term ft . Der Fuzzy-Term beginnt bei $(x_0, y_0) \in ft$ und endet bei $(x_{n-1}, y_{n-1}) \in ft$. Ist x echt kleiner als ft , also gilt $x < x_0$, so ist das Prädikat 0. Ist x jedoch echt größer als der ft , also gilt $x > x_{n-1}$, so ist das Prädikat 1. Liegt x jedoch innerhalb von ft , so wird das Prädikat, genau wie $I(x)$, umso größer, je näher man sich dem rechten Rand des Fuzzy-Terms nähert.



Eine genauere Erläuterung des Prädikates **BIGGER** kann analog beim Prädikate **SMALLER** nachgelesen werden, da sich die beiden Prädikate in ihrer Definition kaum unterscheiden. Die Bedingungen (1) und (2) gelten mit den selben Begründungen für das Prädikat **BIGGER** analog zum Prädikat **SMALLER**. Wie an den Graphen zu sehen ist, sind die Prädikate **BIGGER** und **SMALLER** bei einem gegebenem ft symmetrisch zur Achse $y = 1/2$. Die Bedingung (3) besagt, dass $\neg P_\beta = 1 - P_\beta = P_\sigma$ ist beziehungsweise dann auch $\neg P_\sigma = 1 - P_\sigma = P_\beta$ ist, also die Negation von **BIGGER** gleich **SMALLER** beziehungsweise die Negation von **SMALLER** gleich **BIGGER** ist. Beweise:

$$\begin{aligned}
\neg P_{\beta}(x, ft) &= 1 - P_{\beta}(x, ft) \\
&= 1 - \int_{-\infty}^x \mu_{ft}(y) dy / \int_{-\infty}^{\infty} \mu_{ft}(y) dy \quad \text{und} \\
&= P_{\sigma}(x, ft)
\end{aligned}
\qquad
\begin{aligned}
\neg P_{\sigma}(x, ft) &= 1 - P_{\sigma}(x, ft) \\
&= 1 - \left(1 - \int_{-\infty}^x \mu_{ft}(y) dy / \int_{-\infty}^{\infty} \mu_{ft}(y) dy \right) \\
&= \int_{-\infty}^x \mu_{ft}(y) dy / \int_{-\infty}^{\infty} \mu_{ft}(y) dy \\
&= P_{\beta}(x, ft)
\end{aligned}$$

6. Anwendung

6.1. Wartungsbeispiel

Dieses Kapitel beschreibt ein komplettes Wartungsbeispiel geschrieben in der Sprache Temporal Fuzzy Control Language (TFCL), welche auf der Sprache Fuzzy Control Language (FCL, von [IEC97]) basiert. In einem Büroraum soll immer eine minimale Helligkeit vorherrschen, jedoch soll ein Maximalwert auch nicht überschritten werden. Eine Beschränkung ist, dass die Lampen etwa 15 Minuten benötigen, bis sie ihre maximale Helligkeit erreicht haben. Ist es im Büroraum zu dunkel, so müssen rechtzeitig zusätzliche Lampen eingeschaltet werden, denn wenn die minimale Helligkeit erreicht ist, dann ist es zu spät um weitere Lampen einzuschalten. Ist es jedoch zu Hell, weil die Sonneneinstrahlung zu stark ist, dann werden genügend Jalousien geschlossen um den Raum abzudunkeln. Aus diesen Gründen muss der Helligkeitsverlauf vorhergesagt werden. Die Aussenhelligkeit ändert sich im Laufe des simulierten Tages durch Sonnenauf- beziehungsweise Sonnenuntergang oder durch Wolken, welche die Sonnen verdunkeln. Wenn alle acht Lampen angeschaltet werden, wird mehr als genügend Licht produziert, auch wenn es Außen komplett dunkel ist. Auch genügen die drei Jalousien um die maximal mögliche Sonneneinstrahlung genügend zu verringern. Das TFCL Beispiel beinhaltet neben dem Wartungsteil auch noch einen Regelteil, um die Helligkeit im Büroraum auf einem konstanten Level zu halten.

Tabelle 2 beinhaltet die Beschreibung des Wartungsbeispielles geschrieben in TFCL. Die Systemvariablen *numberOfLamps* und *shutterClosed* repräsentieren die Anzahl der Lampen welche an zuschalten und Anzahl der Jalousien welche zu schließen sind und sind somit spezielle Ausgabevariablen. In der Eingabevariable *brightness* steht die gemessene Helligkeit im Büroraum. Die Ereignisvariablen *maintenance* und *pWarning* beinhalten Ereignisse, welche eintreten können. Die gemessene Helligkeit im Büroraum wird fuzzyfiziert. Es gibt hierzu fünf Fuzzy-Terme: *veryLow*, *low*, *med*, *high* und *veryHigh*. Dabei soll die Helligkeit größer als *veryLow* und *low* und kleiner als *high* und *veryHigh* sein, also ist ein Helligkeitswert von *med* erwünscht. Dies alles ist im ersten Abschnitt von Tabelle 2 beschrieben (alles oberhalb des Regelblockes **RULEBLOCK**).

Nach der Deklaration der Variablen und Fuzzy-Terme, werden die Regeln für den Fuzzy-Regler angegeben. Die *monitoring* und *prediction* Regel (Nummer 0) überprüft, ob die Helligkeit in der Zukunft einmal *veryLow* oder *low* wird. Wenn ja, wird das vorhergesagte Warnungsereignis *brightness* für die Helligkeit generiert. Die *maintenance* Regel (Nummer 1) überprüft, ob die Helligkeit schon eine viertel Stunde *low* ist und mehr als sieben Lampen eingeschaltet sind. Wenn ja, dann kann man daraus schließen, dass die Lampen nicht mehr genügend Helligkeit liefern, oder dass es defekte Lampen gibt. Es wird dann ein Wartungsereignis *replaceLamps* generiert, um den Benutzer anzuhalten, defekte Lampen auszutauschen. Die *control* Regeln (Nummer 2-6) werden genutzt um Lampen an oder auszuschalten oder um Jalousien zu öffnen oder

zu schließen. Im Detail, schaltet Regel 2 Lampen an, wenn es in einer viertel Stunde zu dunkel sein würde und wenn alle Jalousien offen sind. Sollten Jalousien geschlossen sein, so möchten wir diese zuerst öffnen, bevor wir anfangen Lampen an zuschalten. Regel 3 öffnet eine Jalousie, wenn die Helligkeit in einer viertel Stunde zu dunkel ist und wenn mit der Bedingung „*shutterClosed PREEXIST last quarter_hour shutterClosed*“ überprüft wurde, ob die Anzahl der geschlossenen Jalousien in der letzten viertel Stunde konstant war. Es wird also keine Jalousie geöffnet, wenn schon vor einer viertel Stunde eine geöffnet oder geschlossen wurde. Dadurch wird die Schaltfrequenz der Jalousien herunter gesetzt und so vermieden, dass ständig eine Jalousie geöffnet oder geschlossen wird. Wenn die Helligkeit zu hoch ist, dann werden mit Regel 4 und 5 Lampen ausgeschaltet, oder wenn alle Lampen ausgeschaltet sind, Jalousien geöffnet.

<i>Wartungsbeispiel in TFCL</i>
<pre> VAR_SYSTEM numberOfLamps actuator: Range: 0..8 REAL; shutterClosed actuator: Range: 0..2 REAL; END_VAR VAR_INPUT brightness: REAL; END_VAR VAR_EVENT maintenance; pWarning; END_VAR_EVENT FUZZIFY brightness TERM <i>veryLow</i> := (0, 1)(43, 1)(112, 0); TERM <i>low</i> := (43, 0)(112, 1)(128, 0); TERM <i>med</i> := (112, 0)(128, 1)(170, 0); TERM <i>high</i> := (128, 0)(170, 1)(213, 0); TERM <i>veryHigh</i> := (170, 0)(213, 1)(255,1); RANGE := (0 .. 255); END_FUZZIFY EVENT maintenance TASK replaceLamps; END_EVENT EVENT pWarning EVENT brightnessWarning; END_EVENT RULEBLOCK AND:MIN; OR:MAX; ACCU:MAX; ACT:MIN; PREDICTION:LINEARITY; RULE_0: IF (brightness WILL_EXIST <i>veryLow</i>) OR (brightness WILL_EXIST <i>veryHigh</i>) THEN (pWarning (brightnessWarning)); RULE_1: IF (brightness WAS last quarter_hour <i>veryLow</i>) AND (numberOfLamps > 7) THEN (maintenance (replaceLamps)); RULE_2: IF ((brightness WILL_BE next quarter_hour <i>veryLow</i>) OR (brightness WILL_BE next quarter_hour <i>low</i>)) AND (shutterClosed < 0.5) THEN (numberOfLamps ++); RULE_3: IF ((brightness WILL_BE next quarter_hour <i>veryLow</i>) OR (brightness WILL_BE next quarter_hour <i>low</i>)) AND (shutterClosed PREEXIST last quarter_hour shutterClosed) THEN (shutterClosed --); RULE_4: IF (brightness WILL_BE next quarter_hour <i>high</i>) OR (brightness WILL_BE next quarter_hour <i>veryHigh</i>) THEN (numberOfLamps --); RULE_5: IF (brightness WILL_BE next quarter_hour <i>high</i>) OR (brightness WILL_BE next quarter_hour <i>veryHigh</i>) AND (numberOfLamps < 0.5) THEN (shutterClosed ++); END_RULEBLOCK </pre>

Tabelle 2: Komplettes Wartungsbeispiel geschrieben in der Sprache Temporal Fuzzy Control Language (TFCL) um die Helligkeit in einem Büroraum zu regeln und Wartungsaufträge zu generieren.

6.2. Experiment

Kapitel 6.1 führt ein Wartungsbeispiel ein, welches nun in einem simulierten Experiment eingesetzt wird. Die Randbedingungen sind, dass die Helligkeit im Büroraum nicht unter *low* oder *veryLow* und nicht über *high* oder *veryHigh* liegen. Die Helligkeit ist aus einem Intervall von 0 (Minimum) bis 255 (Maximum). Der Schwellwert, ab welcher Regelaktivierung die Regeln für die Systemvariablen *numberOfLamps* und *shutterClosed* feuern, ist auf 50% gesetzt. Das heißt, die Regeln 2-5 aus Tabelle 2 feuern nur bei einer Aktivierung der Bedingungen von mehr als 50%. Betrachtet man nun die Randbedingung an die Helligkeit, die Fuzzifizierung der Helligkeit und den Schwellwert der Regelaktivierung, so bedeutet dies, dass die Helligkeit zwischen 120 und 149 liegen sollte und nicht die Helligkeit 170 überschreiten beziehungsweise 112 unterschreiten darf.

Die Helligkeit o von Außen, welche durch das Sonnenlicht verursacht ist und in den Büroraum einstrahlt hängt von der Tageszeit t ab und wird durch folgende Gleichung angenähert:

$$o(t) = 192 \cdot \sin\left(\frac{t-5h}{14h} \cdot \pi\right) + \text{noise}$$

Wird eine Lampe eingeschaltet, so steigt deren abgestrahlte Helligkeit innerhalb von 15 Minuten linear von 0 auf 20 an. Wird die Lampe ausgeschaltet, so verringert sich die Helligkeit sofort auf 0.

Experimente mit der beschriebenen Simulation ergeben, dass die Helligkeit während des gesamten Tages innerhalb der geforderten Randbedingungen liegen. Bei der Helligkeit ist der Durchschnitt 135,4, das Maximum 163,4, das Minimum 113,8 und die Standardabweichung 6,9. Der Verlauf der Helligkeit ist in Abbildung 1 durch das Symbol X dargestellt.

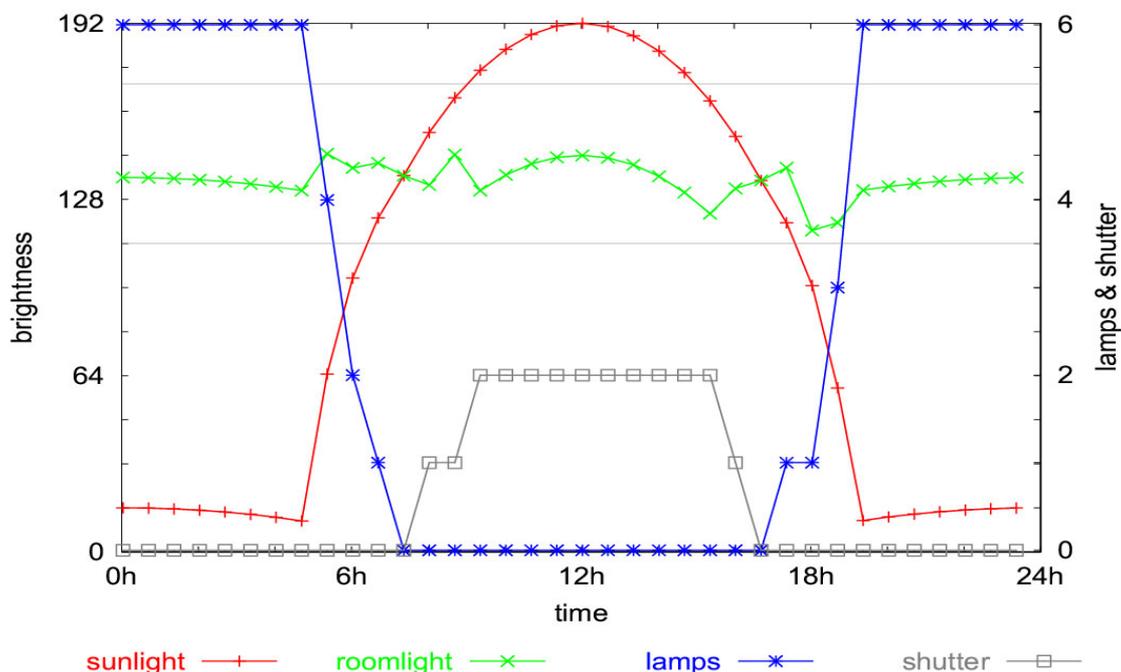


Abbildung 1: Experimentelle Ergebnisse für einen kompletten 24 Stunden Tag für den Helligkeitsverlauf im Büro (x), welcher durch Sonneneinstrahlung (+) gestört wird. Die Anzahl der angeschalteten Lampen (*) und geschlossenen Jalousien (□) regelt die Helligkeit zwischen dem maximal und minimal gewünschten Wert. Die maximal und minimal gewünschten Werte sind durch horizontale Linien bei Helligkeit gleich 112 und 149 dargestellt.

7. Schlussfolgerung

Wir haben gezeigt, dass es möglich ist, die Fuzzy-Logik mit temporalen Prädikaten zu erweitern, so dass wir die so genannte Temporale Fuzzy-Logik erhalten. Diese ermöglicht die Modellierung von zeitlichen Abhängigkeiten von Ereignissen und kann in einem Fuzzy-Regler zur Überwachung, Regelung und Wartung eingesetzt werden. Dieser Einsatz wird in einem Wartungsbeispiel gezeigt, in welchem ein Benutzer über defekte Lampen informiert wird.

Außerdem wurde im Hauptteil die mathematische Basis zu den temporalen Fuzzy-Prädikaten geschaffen. Die temporalen Fuzzy-Prädikate sind eine eins zu eins Abbildung der temporalen Prädikate, welche nach [Karjoth87] vollständig sind, das heißt mit ihnen können Bedingungen über den gesamten Zeitbereich erstellt werden. Somit sind die temporalen Fuzzy-Prädikate ebenfalls vollständig. Des weiteren kann deren Auswertung effizient ausgeführt werden.

Es ist möglich im Bedingungsteil einer Regel **AND** beziehungsweise **OR** verknüpfte Prädikate zu verwenden. Wie die Auswirkung von den angegebenen Zeiten auf die Schlussfolgerungen sind, ist zu diesem Zeitpunkt noch nicht bekannt, denn der Einfluss der Zeitangaben der Regelbedingung auf die Regelfolgerung ist noch nicht untersucht, aber er ist dennoch zu beachten. Im folgenden Beispiel werden die Daten vom Sensor S^i in der Vergangenheit aufgezeichnet und damit ein Aktuator A^i in der Gegenwart gesteuert:

$$\mathbf{IF } S^i \mathbf{ WAS}_{[t_j, t_k]} ft \mathbf{ THEN } A^i \mathbf{ IS } ft$$

Eine mögliche Interpretation ist, dass das Prädikat **IS** in einer Folgerung am wenigsten restriktiv ist, das heißt es ignoriert das Zeitintervall welches in der Bedingung beim Prädikat **WAS** gegeben ist und aktiviert die Regel, wenn die Bedingung erfüllt ist. Ersetzt man in der Folgerung **IS** durch **WAS**, so lässt man nur noch Zeitintervalle in der Vergangenheit zu und filtert Zeitintervalle in der Zukunft heraus, indem die entsprechenden Regeln nicht mehr aktiviert werden. **EXISTED** beziehungsweise **WILL_EXIST** lassen nur einen fixen Zeitpunkt in der Vergangenheit beziehungsweise Zukunft zu. Somit kann durch die Angabe eines temporalen Prädikates in der Folgerung einer Regel eine Beschränkung der Zeitintervalle der Bedingung bewirkt werden. Eine andere Möglichkeit der Interpretation ist, dass ein temporales Prädikat bei der Folgerung den Zeitpunkt der Aktivierung angibt. So dass ein Aktuator erst zu einem späteren Zeitpunkt gesetzt wird. Dies sind nur angedachte Ideen, welche vor allem bei der Verwendung mehrerer Prädikate pro Regelbedingung noch weiter untersucht werden müssen.

8. Literatur

- [Althoff92] K.-D. Althoff, *Eine fallbasierte Lernkomponente als integrierter Bestandteil der MOLTKE-Werkbank zur Diagnose technischer Systeme*, Dissertation, Kaiserslautern, September 1992
- [Bothe95] H.-H. Bothe, *Fuzzy-Logik – Einführung in Theorie und Anwendungen*, Springer-Lehrbuch, 2. Auflage, Berlin Heidelberg, 1995
- [Bovenkamp98] E. G. P. Bovenkamp, J. C. A. Lubbe, *Temporal Reasoning with Fuzzy-Time-Objects*, 4th International Workshop on Temporal Representation and Reasoning, Daytona Beach, Florida, May 10-11, 1997
- [Cárdenas02] M. A. Cárdenas Viedma, R. Martín Morales, *Syntax and Semantics for a Fuzzy Temporal Constraint Logic*, Annals of Mathematics and Artificial Intelligence, Volume 36, 2002

- [Castillo02] O. Castillo, P. Melin, *A New Approach For Plant Monitoring Using Type-2 Fuzzy Logic and Fractal Theory*, Proceedings of the 5th International FLINS Conference, Belgium, Belgium, September 2002
- [Fantoni00] P.F. Fantoni, M. Hoffmann, B. H. Nystad, *Integration of sensor validation in modern control room alarm systems*, Proceedings of the 4th International FLINS Conference, Belgium, August 2000
- [Fick00] A. Fick, H. B. Keller, *Modellierung des Verhaltens Dynamischer Systeme mit erweiterten Fuzzyregeln*, Proceedings 10. Workshop Fuzzy Control des GMA-FA 5.22, Dortmund, Germany, 18. - 20. Oktober 2000
- [Flender02] Flender Service GmbH, *Condition Monitoring for the highest Availability of Power Technology*, http://www.flender-cm.de/images/pdf/Leistungskurzbeschreibung_GB.pdf, Juni 2002
- [Giron02] J. M. Giron-Sierra, G. Ortega, *A Survey of Stability of Fuzzy Logic Control with Aerospace Applications*, IFAC Proceedings of the 15th Triennial World Congress, Bachelona, Spain, 2002
- [Haslum01] P. Haslum, *Models for Prediction*, IJCAI 2001 workshop on Planning under Uncertainty
- [IEC97] IEC TC65/WG 7/TF8, *Fuzzy Control Programming*, International Technical Electrical Commission (IEC), 1997
- [Karjoth87] G. Karjoth, *Prozeßalgebra und temporale Logik – angewandt zur Spezifikation und Analyse von komplexen Protokollen*, Diss. Mathematik/Informatik, Universität Stuttgart, 1987
- [Lamine01] K. B. Lamine, F. Kabanza, *Reasoning About Robot Actions: A Model Checking Approach*, Revised Papers from the International Seminar on Advances in Plan-Based Control of Robotic Agents, Springer Verlag, 2001
- [Palit00] A. K. Palit, *Artificial Intelligent Approaches to Times Series Forecasting*, Dissertation, Bremen, Januar 2000
- [Škrjanc02] I. Škrjanc, D. Matko, *Fuzzy Predictive Functional Control in the State Space Domain*, *Journal of Intelligent and Robotic Systems*, Kluwer Academic Publishers, 2002
- [Takagi85] T. Takagi, M. Suego, *Fuzzy identification of systems and its application to modeling and control*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 15, No. 1, pp. 116-132, 1985.
- [Watanabe86] H. Watanabe, *Schwerpunktmethode*

Fuzzy Decision Trees - A New CI-Method for the Automatic Data Analysis Platform SPIDA

Xiaomeng Wang¹, Detlef D Nauck², Martin Spott², Rudolf Kruse¹

¹ Faculty of Computer Science, University of Magdeburg
Universitaetsplatz 2, D-39106 Magdeburg, Germany
E-Mail: {xwang, rkruse}@iws.uni-magdeburg.de

² BT, Research and Venturing, Intelligent Systems Research Centre
Adastral Park, Orion Bldg. pp1/12
Ipswich IP5 3RE, UK
E-Mail: {detlef.nauck, martin.spott}@bt.com

1 Introduction

Intelligent data analysis is an interdisciplinary science that has emerged as the product of the union of statistics, soft computing, machine learning, artificial intelligence and others, and aims at problems like classification, segmentation, concept description, prediction, trend analysis, dependence/association analysis etc. Here we focus on the classification problem.

Different classification models have individual strengths and weaknesses: sometimes a classifier is good for prediction but difficult to understand, like neural networks. Or a model works well in crisp domains, but cannot model vagueness, like classical decision trees. In order to learn a classification model, which is comprehensible and able to handle vagueness, we have combined fuzzy theory with classical decision trees.

We begin our paper with the introduction of automatic data analysis and the platform SPIDA that was developed in BT's Intelligent Systems Research Centre, for which our fuzzy decision tree module was investigated and implemented. In Section 4 we present our algorithm, examine the core problem of how to compute the information measure in the attribute selection step, and discuss how to treat missing values. In addition we consider how to extract a fuzzy rule base from the induced fuzzy tree by which the classification is finally performed and we study heuristics to simplify the rule base. In Section 5 we present our experimental results obtained with an implementation of our algorithm and compare them to those of some popular classifiers. Section 6 concludes the work and points out some further work.

2 Approaches to Automatic Data Analysis

Previous approaches towards automating data analysis or knowledge discovery in databases were based on AI techniques. Analysis methods were broken down into formal blocks and user requirements were also represented in a formal language. Then a search algorithm would identify suitable blocks and arrange them in a way to carry out an analysis process [16]. These approaches had to face the problem of formalising mainly heuristic methods and that it is usually not feasible to formally compute all necessary parameters to execute an analysis method.

Other authors discussed mainly architectural features of systems that could automate data analysis or data mining and avoided discussing how to automatically select and execute analysis methods [7, 6].

A more recent system uses an ontology based approach and simply describes analysis methods and pre-/post-processing methods as input/output blocks with specific interfaces [2, 1]. The system is built on top of the data analysis package Weka [17]. If the interfaces between two blocks match, they can be concatenated in an analysis process. If a user wants to analyse a data set, all possible analysis processes are created and executed. Once a suitable analysis process has been identified, it can be stored, re-used and shared. The authors suggest a heuristic ranking of analysis processes in order to execute only the best processes. However, they only use speed as a ranking criterion, which can be easily determined as a feature of an algorithm. More useful features about the quality of the analysis like accuracy are obviously dependent on the analysis process as well as the analysed data and are much more difficult to determine up front. Therefore, the reported results in this area are not very encouraging.

In the Computational Intelligence Group (CI Group) of BT's Intelligent Systems Research Centre (ISR Centre) we have followed a different approach. Especially in industrial settings it is important to empower non-expert users in coping with daily data analysis tasks. From our experience take-up of traditional data analysis software is very slow. We believe that typical business users cannot use technology or method-oriented software. Without detailed knowledge of the analysis methods such tools are basically useless. In order to push advanced data analysis technology algorithms into businesses we need software that empowers users and hides complexity from them.

From our point of view users need a tool where they can specify a data source and requirements on the solution of an analysis process. The tool can select appropriate methods and apply them automatically. The results are checked against the user requirements. If the requirements are not sufficiently met, the analysis is repeated with different parameters. At the end, the user is presented with a set of possible solutions and their descriptions in terms of user requirements. After selecting a solution this will be wrapped in a software module that can be readily applied in the user's application domain.

The approach to automating IDA at BT's ISR Centre is based on the following premises:

- User requirements are fuzzy. User's demand *fast, accurate, simple, inexpensive* solutions. All these terms are fuzzy in nature. We have therefore decided to use fuzzy systems for modelling user requirements.
- Most advanced IDA technologies like neural networks, decision trees, neuro-fuzzy system, cluster analysis etc. are all based on heuristics. Based on fuzzy user requirements, it is basically impossible to formally determine exact values for parameters of those methods. We have therefore decided to use fuzzy systems to determine parameters of analysis methods.
- Properties of analysis methods are fuzzy. Some methods are *fast*, some can produce *accurate* results, others are *interpretable*. Such properties are relevant for their application and are best described by fuzzy sets, since many of them are inherently fuzzy.

- Expert knowledge on how to use which analysis method is vague (fuzzy). Data analysis experts have in addition to their formal knowledge about analysis methods vague intuitive knowledge on how to select parameters and methods and how to run an analysis in a certain scenario. We have therefore decided to use fuzzy systems to model such expert knowledge, because this enables us to encode both exact and vague knowledge.

3 SPIDA – A Platform for Automatic IDA

Based on the requirements set above the CI Group at BT's ISR Centre developed SPIDA (Soft computing Platform for Intelligent Data Analysis) [12]. Essentially, SPIDA is a data analysis tool comprising a set of data analysis methods mainly from the area of soft computing and related areas (neural networks, neuro-fuzzy systems, support vector machines, decision trees etc.), data filters for pre- and post-processing, visualisation capabilities and access to different data sources (text files, databases).

The main user group targeted by SPIDA are domain experts. They typically are familiar with their data, they know the processes that produce the data, and are usually keen to review these processes in order to improve or understand them. Furthermore, gained knowledge can also be applied to other problems that are related to the data like using information gained from customer data for marketing purposes. Domain experts are usually no data mining experts, but they can specify their data analysis problem and their requirements for the solution at a high level. Based on this information and the data, the SPIDA Wizard selects and runs data analysis methods automatically. To achieve this, SPIDA uses fuzzy knowledge bases to match fuzzy user requirements against method features and to automatically determine runtime parameters. Each analysis methods that is available in SPIDA is described by fuzzy and non-fuzzy features. For example, a multi-layer perceptron can be described as follows.

- Non-Fuzzy Features
 - suitable for function approximation and classification
 - requires numeric input/output data
 - ...
- Fuzzy-Features
 - high number of parameters
 - very low interpretability
 - can potentially achieve high accuracy
 - ...

SPIDA is implemented in an open client/server architecture. The server runs IDA processes and the client functions as a graphical user interface (GUI) and can connect both to local and remote SPIDA servers. A SPIDA server provides a plug-in API that can be used to connect basically any data analysis method or software to SPIDA. Connection is possible, for example, by direct method invocation, or by automatically creating control

files that are then executed by a connected software tool in a separate thread. After a new method has been described in the knowledge base, SPIDA is ready to use it.

SPIDA consists of the following function blocks

- Wizard: user interface for non-expert users to run SPIDA in automatic mode.
- Automatic pre-processing: detection of data types, handling of missing values, normalisation, scaling, re-coding etc.
- Automatic method selection: depending on the user requirements and the data to be analysed methods are selected. If a method requires changes to the data format pre-processing is invoked.
- Automatic method execution: IDA processes are configured and started. Method execution is monitored by the SPIDA knowledge base and methods are automatically re-configured or re-run if necessary.
- Adaptive user profiling: for each user a profile is maintained. Profiles comprise user-specific requirements for IDA processes. They are adapted according to the acceptance or rejection of IDA results.
- Automatic result evaluation. This module compares results of IDA processes to (fuzzy) user requirements. If no sufficient match is obtained, the processes are re-configured and re-run to obtain a better match.
- Automatic Solution Generation. This module wraps the selected IDA results into executable objects with standardised interface, which can be used in user applications directly.
- Expert interface: experts can use SPIDA in a non-automatic expert mode, where IDA processes are graphically constructed and can be fully controlled.

For a more detailed description of the SPIDA wizard that drives the automatic data analysis process see [12, 13]

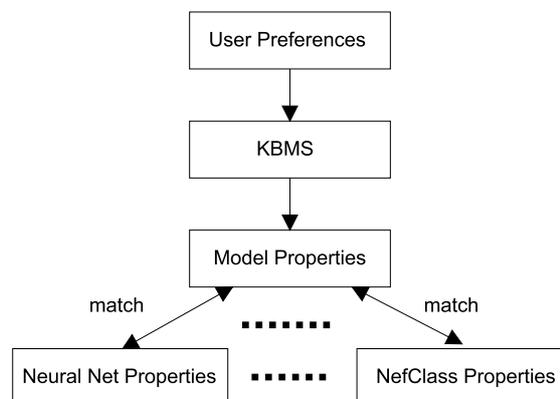


Figure 1: Automatic selection of methods. Using the KBMS, wanted model properties are inferred from user preferences. Properties of available models are matched against wanted properties, which results in a ranking of models.



Figure 2: Choosing preferences in the Wizard. Left: the dialog for an explanation facility of methods. Right: ranking of data analysis methods according to user preferences before model creation

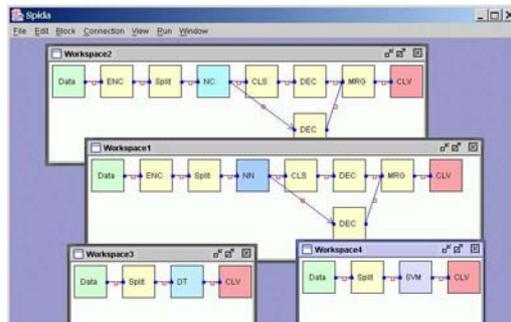


Figure 3: Workspaces automatically created by the wizard for four different data analysis methods. The workspaces are usually not shown by the wizard. This view also represents the GUI an IDA expert would use to configure data analysis processes manually.

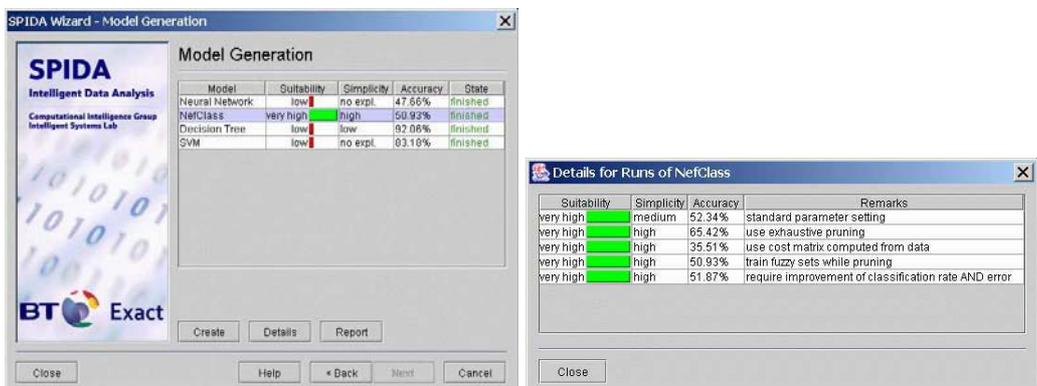


Figure 4: Left: results of the model creation process. Right: automatic re-runs of the NEFCLASS model.

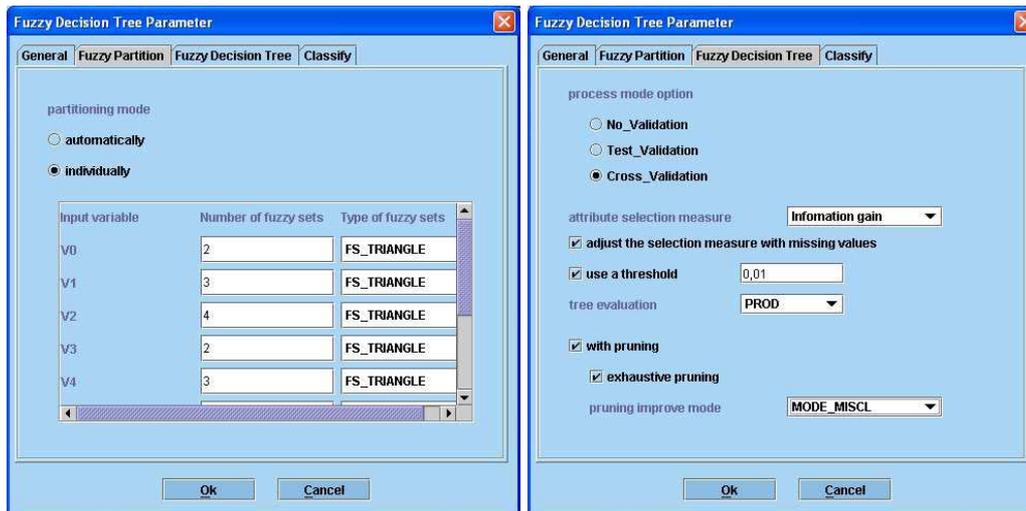


Figure 5: Left: Fuzzy partitioning of the input attributes. Right: Fuzzy decision tree configuration

4 Fuzzy Decision Trees

In the past several variants of fuzzy decision trees were introduced by different authors. Boyen and Wenkel [8] presented the automatic induction of binary fuzzy trees based on a new class of discrimination quality measures. Janikow [9] adapted the well-known ID3 algorithm so that it works with fuzzy sets. In this paper, we also adapted the ID3 algorithm to construct fuzzy decision trees and borrowed some basic ideas from [9]. Going beyond Janikow's work we consider how to extract a fuzzy rule base from the resulted fuzzy tree and use this rule base to perform the classification.

The fuzzy decision tree module here has been developed for SPIDA. In this implementation, the configuration for the underlying program is done through the GUI (some of it will be shown in the following), which is running on the client side; and the algorithm for model learning is resident on the server side.

4.1 Tree Induction

In this paper we focus on the induction of a fuzzy decision tree (FDT) on continuous attributes. Before the tree induction a fuzzy partition has to be created for each attribute (see Figure 5). The fuzzy sets of these partitions will be used as fuzzy tests in the nodes of the fuzzy tree. To initialize these fuzzy partitions we adopted an existing algorithm, which creates them either completely automatically based on a given data set (called "automatic partitioning"), or based on a user specification of the shape and number of the membership functions (called "individual partitioning"). In the latter case the fuzzy sets are distributed evenly over the entire domain of each attribute. Here we assume the fuzzy partitions of the input variables are given.

Like classical decision trees with the ID3 algorithm, fuzzy decision trees are constructed in a top-down manner by recursive partitioning of the training set into subsets. From the algorithm presented in Figure 6, we can see some particular features of fuzzy tree learning as follows:

FDT($examples\{e_k\}, training\ set\{\chi_k\}, class_variable, attributes$) /* $1 \leq k \leq s$, where s is the number of examples */

- Create a *Root* node for the tree
 - If all examples belong to one class, return *Root*
 - If *attributes* is empty, return *Root* with the membership degrees to the classes
 - Otherwise begin
 - Calculate the information measure for each attribute (details see below)
 - $A \leftarrow$ the “best” attribute according to the highest information measure, if this information measure is below the threshold specified by the user, return root; else continue
 - The test attribute for *Root* $\leftarrow A$
 - For each fuzzy set a_p of A
 - * Add a new branch below *Root* connecting with the test $A = a_p$
 - * Update training set $\{\chi_k\}$ into $\{\chi_k^{a_p}\}$ according to the membership degree to a_p
 - * If all elements of $\{\chi_k\}$ are equal 0, go to the next fuzzy set a_{p+1} , else below this new branch add the subtree FDT($examples\{e_k\}, training\ set\{\chi_k^{a_p}\}, class_variable, (attributes-\{A\})$)
 - Return *Root*
-

Figure 6: FDT learning algorithm

1. The membership degree of examples

The training set $\chi = \{\chi_1, \dots, \chi_s\}$ (in node N denoted as χ^N) is the fuzzy set over a set of examples $E = \{e_1, \dots, e_s\}$ defined by membership degrees. The membership degree of an example to an example set lies in the interval $[0, 1]$. In each node, an example has a different membership degree to the current example set, and this degree is calculated from the conjunctive combination of the membership degrees of the example to the fuzzy sets along the path to the node, where different t -norms (\top) can be used for this combination.

2. Selection of test attributes

A standard method to select a test attribute in classical decision tree induction is to choose the attribute that yields the highest information gain. In node N , we will calculate the information gain $\text{Gain}(\chi^N, A) = I(\chi^N) - I(\chi^N|A)$ for each candidate attribute A . However due to fuzzy logic, we must extend the information measures (we also consider gain ratio here) to ensure a correct ranking of the candidates in FDT. We will discuss this point in detail below.

3. Fuzzy tests

As mentioned above, in the inner nodes fuzzy tests are used instead of crisp tests.

A fuzzy test of an attribute means to determine the membership degree of the value of an attribute to a fuzzy set.

4. Stop criteria

Usually classical tree learning is terminated if all attributes are already used on the current path; or if all examples in the current node belong to the same class. Here we add a novel condition (see Figure 5), namely whether the information measure is below a specified threshold. In FDT any example can occur in any node with any membership degree. Thus in general more examples are considered per node and fuzzy trees are usually larger than classical trees. The threshold defined here enables a user to control the tree growth, so that unnecessary nodes are not added. The experiments prove that an adequate threshold helps not only to avoid overfitting, but also to decrease the complexity of the tree.

4.2 Information Measures: The Problems And Extensions

In this subsection we discuss the problems that occur if we apply information gain, which is a standard method for the selection of a test attribute in classical trees, directly in fuzzy decision tree induction.

The definition of information gain is based on probability theory. Its value, as we know, can never be negative (a prove can be found, for instance, in [5]). However, depending on the computation, in FDT negative information gain is possible. This phenomenon occurs due to the following two reasons, both of which can lead to a situation in which the sum of the weights of the example cases before and after a split and thus the class frequency distributions differ.

1. In fuzzy logic, the sum of the membership degrees of a value to the fuzzy sets of its variable can differ from 1, depending on how the fuzzy sets overlap. A simple example to illustrate how membership degrees influence information measures, can be found in [15].
2. Probability theory prescribes to use the product to express a (conditional) conjunction (i.e., $P(X \wedge Y) = P(X | Y) \cdot P(Y)$), whereas fuzzy logic offers other possibilities besides the product, for example $\top_{\min}(a, b)$ and $\top_{\text{Łuka}}(a, b)$.

A negative information gain, although it has no real meaning, can still yield a correct ranking of the candidate test attributes. But if information gain ratio is used, a negative value for the information gain can produce an inappropriate answer. To see this, let us consider a simple example: suppose for the current tree node N we have two candidate attributes A and B with information gain $\text{Gain}(\chi^N, A) > \text{Gain}(\chi^N, B)$ and split information $\text{SplitI}(\chi^N, A) \gg \text{SplitI}(\chi^N, B)$, where χ^N is the fuzzy example set (a fuzzy set over a reference set E of examples) in N , which acts actually as the training set.

1. In classical decision trees it is always $\text{Gain}(\chi^N, A) > \text{Gain}(\chi^N, B) \geq 0$, and then it may be that

$$0 \leq \frac{\text{Gain}(\chi^N, A)}{\text{SplitI}(\chi^N, A)} < \frac{\text{Gain}(\chi^N, B)}{\text{SplitI}(\chi^N, B)}. \quad (1)$$

This is desired, because it reduces the well-known bias of information gain towards many-valued attributes.

2. In the fuzzy domain, however, we can also have the situation $\text{Gain}(\chi^N, A) > 0 > \text{Gain}(\chi^N, B)$. In such a case, attribute A is always favored by the information gain ratio, because

$$\frac{\text{Gain}(\chi^N, A)}{\text{SplitI}(\chi^N, A)} > 0 > \frac{\text{Gain}(\chi^N, B)}{\text{SplitI}(\chi^N, B)}, \quad (2)$$

independent of the relationship between $\text{SplitI}(\chi^N, A)$ and $\text{SplitI}(\chi^N, B)$, and this contradicts our intuition.

In our investigation we find that negative information gain can occur in FDT induction (e.g. in the version developed in [9]) if we use the entropy ($I(\chi^N)$) computed from the membership degrees of the examples (interpreted as case weights) in the current node N , in which a test attribute is to be chosen, as the first term for the computation of the information gain. This was caused by the fact, that in fuzzy domain the sum of the case weights in the training set has changed after splitting according to the fuzzy partition of the candidate attribute, and accordingly the class frequency distribution as well, which therefore causes the premiss of the classical information gain to be violated.

In order to eliminate the problems with negative information measures we suggested a different way of the computing: by using the entropy $I(\hat{\chi}^N)$ of the examples which implicitly includes the information of the test attribute, i.e., the entropy derived from the case weights $\hat{\chi}^N$ of the examples in the whole set after split, we have $\widehat{\text{Gain}}(\chi^N, A) = I(\hat{\chi}^N) - I(\chi^N|A)$. This can guarantee a non-negative information gain and thus make information gain ratio applicable in FDT. More details about the suggested improvements can be found in [15].

Remark: information gain ratio is used in C4.5 [14] to select the test attribute in order to reduce the natural bias of information gain, i.e., the fact that it favors attributes with many values (which may lead to a model of low predictive power). In FDT induction, fuzzy partitions are created for all attributes before the tree induction. To keep the tree simple, usually each partition possesses as few fuzzy sets as possible. Since the outgoing branches are labelled with fuzzy sets instead of crisp values, the problem mentioned above is mitigated, because continuous values are mapped to few fuzzy sets and thus the problem of many values is less severe. Therefore the effect of using information gain ratio in FDT may not be so obvious as in classical decision trees.

4.3 Missing Value Handling

Real data often contain missing values. To handle such data we extended the learning algorithm. The first question to be answered is how to assign the examples with missing values of the test attribute to the outgoing branches of a tree node. In this paper a popular method from classical decision trees is adopted: an example e_k is distributed evenly to all children if the value u_k^i for test attribute A_i is unknown. It means that to each fuzzy set (branch) a_p^i of A_i the example e_k is assigned with the membership degree $\mu_{a_p^i}(u_k^i)$:

$$\mu_{a_p^i}(u_k^i) = \frac{1}{|D_i|} \quad \text{if } u_k^i \text{ is unknown,} \quad (3)$$

where $|D_i|$ is the number of the fuzzy sets of A_i .

As we know, the information gain can be interpreted as “the information gained about the classes by ascertaining the value of the test attribute”, a test of an example with a missing value for the test attribute, can obviously provide no information about the class membership of this example. Therefore the assessment of candidate attributes has to be modified accordingly, so that attributes with missing values are penalized.

Suppose we are given a reference set E having missing values for attribute A_i . Then the calculation of the information gain for candidate attribute A_i can, as suggested in [14], be modified as following:

$$\begin{aligned}\widehat{\text{Gain}}(\chi^N, A_i) &= \text{frequency of examples with known } A_i \cdot (I(\hat{\chi}^N) - I(\chi^N|A_i)) \\ &+ \text{frequency of examples with unknown } A_i \cdot 0 \\ &= \alpha \cdot (I(\hat{\chi}^N) - I(\chi^N|A_i)), \text{ where } \alpha = \frac{Z^{N|A_i \text{ known}}}{Z^N}\end{aligned}\quad (4)$$

Due to the factor α , which is computed from the counter for examples without missing values of A_i “ $Z^{N|A_i \text{ known}}$ ” and the counter for the entire examples “ Z^N ”, the real information gain is only dependent on those examples with known values for the test attribute.

The information gain ratio can be amended in a similar way:

$$\text{GainR}(\chi^N, A_i) = \alpha \cdot \frac{I(\hat{\chi}^N) - I(\chi^N|A_i)}{\text{SplitI}(\chi^N, A_i)} \quad (5)$$

Since the split information $\text{SplitI}(\chi^N, A_i)$ is the entropy of the frequency distribution over the values of attribute A_i , the split information is increased artificially by evenly splitting the examples with missing values, and the information gain ratio is decreased accordingly (since $\text{SplitI}(\chi^N, A_i)$ appears in the denominator). This effect is desired, because an attribute having missing values should be penalized. Since the increased split information already penalizes the measure, one may consider making the use of the factor α (see above) optional. That is, it is added only when a user explicitly requests it.

4.4 Fuzzy Rule Base

An important goal of this work is to generate a comprehensible classification model, here a fuzzy rule base, which can be generated from the fuzzy decision tree by transforming each path to a leaf of the tree into a rule. The antecedent of each rule consists of the fuzzy sets along that path leading to the leaf and the consequent of each states membership degrees for the classes which can be read directly from that leaf.

A simpler model or a model with better predictive power cannot be produced by such rewriting of the tree. To achieve this an optimization of the rule base is necessary. We optimize the rule base by rule pruning, where three heuristic strategies are used, which are adapted from [11]:

1. Pruning by information measure: the attribute having the smallest influence on the output should be deleted.

2. Pruning by redundancy: the linguistic term, which yields the minimal membership degree in a rule in the least number of cases, should be deleted.
3. Pruning by classification frequency: The rule, which yields the maximal fulfillment degree in the least number of cases, should be deleted.

These three strategies are called successively by the pruning algorithm. At each pruning step only one attribute or one fuzzy set or one rule is deleted from the system. Before each step the current rule base is to be saved. Once the execution of one strategy is finished, we should check, whether this pruning step improved the rule base successfully. If the rule base simplified by this step has a better performance than the original one, the changed rule base is kept and then is to be pruned further with the same strategy. Otherwise the rule base should be reset with the previously saved version, and the pruning continues then with the next strategy. The pruning process can work automatically without any user interactions.

Since the comprehensibility of a fuzzy system can be defined by the number of the rules, the number of attributes used in a rule and the number of fuzzy sets per attribute, the heuristics used in the strategies above are plausible. We will not present how the fuzzy rule base classifies new data here, as it basically works in the same manner as standard fuzzy systems.

5 Experiments

In this section, we report some results obtained from experiments run with the program FDT¹, which was written by the first author of this paper, the well-known decision tree learner C4.5 (Release 8)² [14], a neural network training program [4], and NEFCLASS [10], which can generate a fuzzy rule based classifier by coupling neural networks with fuzzy systems. We compare the models generated by these programs w.r.t. precision, complexity, and the ability of dealing with missing values. For the tests we used five data sets from the UC Irvine Machine Learning Repository [3]. Table 1 shows general information about these data sets.

All experiments were run with 10-fold cross validation. C4.5 was run with the standard configuration. In NEFCLASS for each attribute a fuzzy partition with three evenly distributed fuzzy sets was created. Fuzzy sets were also optimized during the rule pruning. The neural network program trained a multilayer perceptron (MLP) with one hidden layer (3 neurons) for 1000 epochs.

Since we tried to generate comprehensible classification models, a trade-off between precision and complexity should be found. With this concern in mind, in FDT a threshold of 0.05 for the information measure was chosen. That is, a test is created only if the chosen test attribute yields an information value higher than 0.05.

¹We used the rule base generated by FDT for the experiments.

²The learning result of C4.5 can be both a tree or a rule base. Here we used the generated rule base for the experiments.

<i>data</i>	<i>size</i>	<i>attributes</i>	<i>classes</i>	<i>missing value</i>
iris	150	4	3	no
glass	214	10 (incl. Id)	7	no
thyroid	215	5	3	no
wbc	699	10 (incl. Id)	2	yes
pima	768	8	2	no

Table 1: test data

<i>model</i>		<i>iris</i>	<i>glass</i>	<i>thyroid</i>	<i>wbc</i>	<i>pima</i>
<i>FDT</i> (1)	$\bar{\epsilon}$	4.67%	34.29%	3.33%	2.79%	31.32%
	<i>n</i>	3	11	5	12	2
<i>FDT</i> (2)	$\bar{\epsilon}$	5.33%	31.90%	7.62%	2.64%	18.82%
	<i>n</i>	3	33	10	17	40
<i>C4.5</i>	$\bar{\epsilon}$	4.01%	33.54%	7.03%	4.83%	23.3%
	<i>n</i>	4	14	7	8	8
<i>NEFCLASS</i>	$\bar{\epsilon}$	3.33%	32.19%	11.60%	2.35%	25.89%
	<i>n</i>	3	14	6	21	14
<i>NN</i>	$\bar{\epsilon}$	6.25%	31.27%	3.54%	5.05%	24.67%

Table 2: 10-fold cross validation

5.1 Precision And Complexity

Table 2 shows the average error rate $\bar{\epsilon}$, as well as the number of rules n of the resulting pruned classifiers. The best error rate of the models is printed in bold in the table.

In these experiments, FDT was run with two different initial partitioning of the attributes – the automatic (labelled as *FDT* (1)) and the individual partitioning (labelled as *FDT* (2)) mentioned above. With the individual partitioning each attribute was partitioned with three fuzzy sets, which were evenly distributed over the attribute’s domain, while with automatic partitioning the number of fuzzy sets was determined by the program.

If we consider only the precision of the models, it is very difficult to say which method is the best one, since each method produces the best result at least once. C4.5 never yields the worst error rate. FDT (1) gives the highest precision (3.33%) for the *thyroid* data and at the same time a very small rule base (5 rules). For the *wbc* data NEFCLASS achieves the best performance of 2.35% with as many as 21 rules, while FDT (1) provides performance only slightly worse (2.79%), for which it needs only about half the rules (12). The neural network fares worst for the *wbc*, which is probably due to the fact that an MLP with three hidden neurons (as used here) is comparable in power to about 3 rules. With so few rules no good performance can be expected for the *wbc* data.

For the *pima* data the worst classification rate is provided by FDT with the automatic partitioning (however, with only 2 rules). The reason is that the partitioning algorithm created for only 2 of the 8 attributes two fuzzy sets and only one fuzzy set for each of

<i>data</i>		5%			10%		
		<i>FDT</i>	<i>C4.5</i>	<i>Nefclass</i>	<i>FDT</i>	<i>C4.5</i>	<i>Nefclass</i>
<i>iris</i>	$\bar{\epsilon}$	4.67%	8.01%	5.33%	10.67%	12.00%	6.67%
	n	4	5	3	3	3	3
<i>glass</i>	$\bar{\epsilon}$	39.52%	34.61%	37.19%	29.04%	40.25%	39.18%
	n	13	11	18	21	10	23
<i>thyroid</i>	$\bar{\epsilon}$	3.81%	8.34%	33.92%	8.57%	10.24%	18.53%
	n	6	7	3	4	6	5
<i>wbc</i>	$\bar{\epsilon}$	5.51%	4.86%	4.87%	7.68%	5.28%	5.58%
	n	23	11	32	20	12	39
<i>pima</i>	$\bar{\epsilon}$	31.45%	26.71%	22.66%	35.00%	27.23%	27.59%
	n	2	8	21	2	6	25

Table 3: Learning from data with missing values

the rest. Therefore the potential number of rules is only four, with which no learning algorithm can do much. In a comparison with the best result of FDT (with individual partitioning, which required as many as 40 rules), we noticed that the attributes of the *pima* data have a relatively strong interrelationship. Therefore the data can be predicted better only by combining several attributes. A finer granularity, which was achieved by FDT with the individual partitioning, enhanced the probability of a combination of attributes, and thus led to a better performance.

The same partitioning like in FDT (2) was also used in NEFCLASS. For the *pima* data NEFCLASS provided a slightly lower precision, but with less than half the rules. We assume that the reason is that the fuzzy sets of NEFCLASS were trained during the learning and pruning phase, so they probably fit the data better. In contrast to this the fuzzy sets used in FDT (2) were created once at the beginning and did not change anymore.

If we compare the two groups of results yielded by FDT — taking not only the precision but also the complexity of the classifiers into account — we conclude that the learning process creates better classifiers if it works with automatic instead of individual partitioning. In particular, the number of rules of the first variant is often clearly less than that of the latter. Presumably the reason is that in the first variant the class information is taken into account, whereas it is neglected in the latter.

5.2 Tests On Imperfect Data

The experiments on the data with missing values, which were generated by randomly deleting values from each data set, demonstrate how well different learning methods can cope with imperfect data. In these tests FDT was only run with automatic partitioning.

Two columns of Table 3³ show the results for data sets with 5% and 10% missing values, respectively. The best results are printed in bold face. As expected, the performance of all methods decreased with the increased portion of missing values. FDT provided for

³The neural network program does not appear in this table, since it cannot work with missing values.

the *thyroid* data (both 5% and 10% missing values) the best result, as well as for the *iris* data (5%) and the *glass* data (10%). However, it is impossible to single out a method that is consistently superior to the others. The properties of the data seem to have stronger influence than the portion of missing values.

In C4.5 the threshold values for tests of continuous attributes are determined dynamically and locally in the nodes; in NEFCLASS, although all attributes are partitioned with fuzzy sets before learning, the fuzzy sets are still optimized afterwards. In contrast to this the fuzzy sets used in FDT are not changed anymore after creation. The lack of such dynamic fitting may be a disadvantage of the resulting fuzzy decision tree.

6 Conclusions

With the widespread availability of fast computing power automatic data analysis has become increasingly feasible and valuable. To empower non-expert users in coping with daily data analysis tasks, BT developed the user-centered, automatic data analysis tool SPIDA. Within this framework, we extended classical decision trees by means of fuzzy methods in order to achieve comprehensible classifiers with the abilities to model vagueness.

The learning of FDT was strongly influenced by fuzzy theory, although the principle is the same as that of classical decision trees. Due to the properties of fuzzy logic, the information measure used in FDT for the test attribute selection can become negative. We introduced amendments for two measures—information gain and information gain ratio—to ensure a correct ranking of the candidates and further modifications for them to deal with missing values. Moreover we suggested a threshold for the information measure in order to better control the complexity of the tree. To optimize the fuzzy rules we extracted from a FDT, we transferred three heuristic pruning strategies from NEFCLASS. Since the rules are expressed linguistically, the classifier is easy to interpret. In our experiments we observed that the approach proposed here often generates smaller and at the same time comparably good rule bases. Hence we conclude that we reached our goal of obtaining comprehensible classifiers.

Future work includes the investigation of other fuzzy partitioning techniques to enhance the quality of the initial partitioning and other measures for test attribute selection.

References

- [1] Abraham Bernstein, Shawndra Hill, and Foster Provost. Intelligent assistance for the data mining process: An ontology-based approach. CeDER Working Paper IS-02-02, Center for Digital Economy Research, Leonard Stern School of Business, New York University, New York, 2002.
- [2] Abraham Bernstein and Foster Provost. An intelligent assistant for the knowledge discovery process. CeDER Working Paper IS-01-01, Center for Digital Economy Research, Leonard Stern School of Business, New York University, New York, 2001.
- [3] C.L. Blake and C.J. Merz. UCI Repository of machine learning databases. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.

- [4] C. Borgelt. `mlp` — multilayer perceptron training (computer software). <http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#mlp>.
- [5] C. Borgelt and R. Kruse. *Graphical Models - Methods for Data Analysis and Mining*. J.Wiley & Sons, Chichester,England, 2002.
- [6] Juan A. Botia, Antonio F. Skarmeta, Juan R. Velasco, and Mercedes Garijo. A proposal for meta-learning through a mas. In Tom Wagner and Omer Rana, editors, *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, number 1887 in LNAI, pages 226–233, Berlin, 2000. Springer-Verlag.
- [7] Juan A. Botia, Juan R. Velasco, Mercedes Garijo, and Antonio F. G. Skarmeta. A Generic Datamining System. Basic Design and Implementation Guidelines. In Hillol Kargupta and Philip K. Chan, editors, *Workshop in Distributed Datamining at KDD-98*, New York, 1998. AAAI Press.
- [8] X.P. Boyen and L. Wenkel. Fuzzy Decision Tree induction for power system security assessment. *IFAC Symposium on control of power plants and power systems*, 1995.
- [9] C.Z. Janikow. Fuzzy Decision Trees: Issues and Methods. In *IEEE Transactions on Systems*, volume 28, pages 1–14, Budapest, 1998. Man and Cybernetics.
- [10] D. Nauck and U. Nauck. Nefclass — neuro-fuzzy classification (computer software). <http://fuzzy.cs.uni-magdeburg.de/~nauck/software.html>.
- [11] D. Nauck, U. Nauck, and R. Kruse. NEFCLASS for JAVA — New Learning Algorithmus. In *In Proc. 18th International Conf. of the North American Fuzzy Information Processing Society (NAFIPS99)*, pages 472–476, New York, NY, 1999. IEEE.
- [12] Detlef Nauck, Martin Spott, and Ben Azvine. SPIDA – a novel data analysis tool. *BT Technology Journal*, 21(4):104–112, October 2003.
- [13] Detlef Nauck, Martin Spott, and Ben Azvine. Fuzzy methods for automated intelligent data analysis. In *Proc. Int. Conf. on Fuzzy Systems FuzzIEEE'2004*, Budapest, 2004. IEEE.
- [14] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.
- [15] X. Wang and C. Borgelt. Information Measures in Fuzzy Decision Trees. In *Proc. Int. Conf. on Fuzzy Systems FuzzIEEE'2004*, Budapest, 2004. IEEE.
- [16] R. Wirth, C. Shearer, U. Grimmer, J. Reinartz, T.P. nd Schloesser, C. Breitner, R. Engels, and G. Lindner. Towards process-oriented tool support for knowledge discovery in databases. In *Principles of Data Mining and Knowledge Discovery. First European Symposium, PKDD '97*, number 1263 in Lecture Notes in Computer Science, pages 243–253, Berlin, 1997. Springer-Verlag.
- [17] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann Publishers, San Francisco, CA, 2000. Software available at <http://www.cs.waikato.ac.nz/~ml/>.

Gewinnung von Merkmalen für die Clusteranalyse anhand eines selektiven Gassensors für Hochtemperaturanwendungen bis 900°C

Norman Bauersfeld, Klaus-Dietrich Kramer

Fachbereich AI,
Hochschule Harz, Wernigerode
Friedrichstr. 57-59, 38855 Wernigerode
Tel.: (03943) 659362
Fax: (03943) 659399
E-Mail: nbauersfeld@hs-harz.de

Kurzbeschreibung

Der "Selektive Gassensor für Hochtemperaturanwendungen bis 900°C" sowie die dafür erforderliche Soft- und Hardware zur Auswertung der durch das Sensorsystem gewonnenen Daten und der Kontrolle des damit verbundenen Prozesses stellen ein hochspezialisiertes, intelligentes Messsystem dar. Die Zieleigenschaften seiner Anwendung liegen in der simultanen Analyse von Gasen, hauptsächlich bestehend aus Wasserstoff (H_2) und Kohlenmonoxid (CO), aber auch Sauerstoff (O_2), Kohlendioxid (CO_2), Methan (CH_4) und Wasserdampf (H_2O). Die Detektion von Gasgemischen und deren Konzentrationsbestimmung erfolgt bei unterschiedlichen Temperaturniveaus der Umgebung, wie zum Beispiel zur Prozesssteuerung von Brennstoffzellen oder Reformern.

Entwickelt wird ein Verfahren, welches qualitative und quantitative Aussagen über spezifische Gase erzeugt. Die Auswertung sensorspezifischer Daten erfolgt mit Methoden der "Computational Intelligence". Ziel ist zudem die Entwicklung einer kostengünstigen Auswerte- und Anzeigeeinheit auf einer spezialisierten MC-/DSP-Architektur.

1. Vorbetrachtungen

Die im Folgenden erläuterten Arbeiten basieren auf den an der TU-Clausthal im Laborbetrieb gewonnenen Signaldaten durch Anordnung eines prototypisch erstellten selektiven Gassensors (Abbildung 1) in gashaltigen Umgebungen im Arbeitsbereich von 10^{-20} bis 10^{-15} bar Sauerstoffpartialdruck. Der Sensor besteht aus einem Resonator und einer gassensitiven Schicht. Über eine spezielle Elektrodengeometrie kann nach elektrischer Anregung bzgl. einer Frequenz der komplexe Widerstand mittels Impedanzspektrometer gemessen werden.

Zunächst wurde die Reaktion auf Wasserstoff (H_2) und Kohlenmonoxid (CO), jeweils mit entsprechendem Sauerstoffanteil untersucht. Das spezielle Design des Sensors

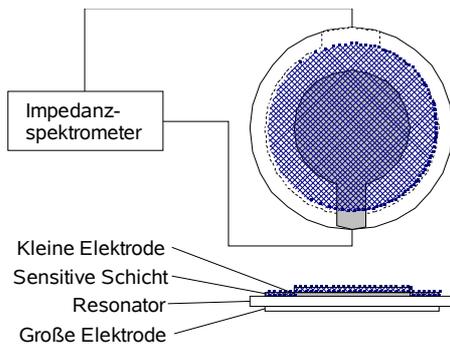


Abbildung 1: Sensorkonzept

ermöglicht eine Auswertung unter Beachtung unterschiedlicher Effekte, wie Masseffekt (Änderung der mechanischen Eigenschaften) und Leitfähigkeits-effekt (Änderung der elektrischen Eigenschaften) des Sensors durch Umschalten zwischen zwei Elektrodengeometrien [1].

Ausgangspunkt der Auswertungen ist die Variation des Verlaufs der charakteristischen Impedanzspektren (Signaldaten $|Z|(f)$; Abbildung 2) im Bereich der Resonanz des beschichteten Resonators durch unterschiedliche Gaskonzentrationen.

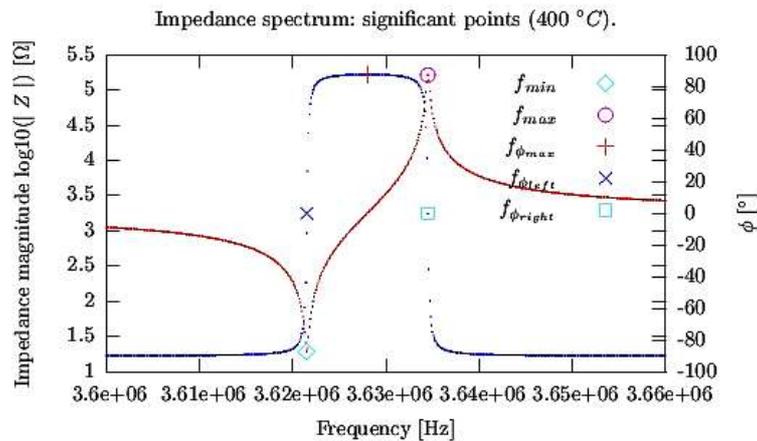


Abbildung 2: charakteristisches Impedanzspektrum

Die Impedanzmessung erfolgt mit Hilfe eines speziellen, für den Anwendungsfall entwickelten, Impedanzanalyzers.

Jeder Punkt des Signals bezeichnet einen komplexen Widerstand Z zu einem Zeitpunkt t . Das vollständige Spektrum besteht aus Grund- und Oberschwingungen. Entsprechende Methoden separieren die Grundschwingung (erster Oberton) und den dritten Oberton. Deren Funktionswerte werden partiell oder im Ganzen mittels Sensormodellbildung ausgewertet. Über das Verhältnis des ersten zum dritten Oberton werden Temperaturkoeffizienten ermittelt, um eine Kompensation der Temperatur einer entsprechenden Größe, z. B. der Serienresonanzfrequenz am Minimum der Impedanz, durchzuführen.

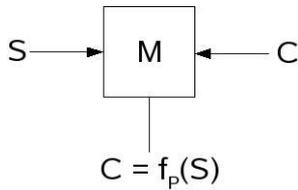
Die S -Dimensionen der Datenpunkte des Sensorspektrums sind auf signifikante Parameter C , deren Änderung bzw. Verlauf äquivalent zu unterschiedlichen, am Sensor vorliegenden Gaskonzentrationen ist, zu reduzieren. Mittels der C -Parameter wird dann ein Modell für die qualitative und quantitative Gaskonzentrationsanalyse erstellt.

2. Modellbildung

2.1. Sensormodell

Eine mögliche Grundlage der Sensormodellbildung ist die Ermittlung von parameterisierten und gewichteten analytischen Werten \underline{C} (Zieldaten) anhand von Signalwerten \underline{S} . Es sei zunächst unerheblich, wie ein Modell funktioniert, welches die Signalwerte auf die Zieldaten abbildet. Mit einem auf den Übertragungsvorgang kalibrierten Modell können die Zieldaten entweder über das inverse oder durch die

rückwärts gerichtete Kalibrierung durch das direkte Modell ermittelt werden. Es sei angenommen, das Modell entspräche dem Übertragungsverhalten eines Systems, welches als Eingabewerte Signaldaten erhält und als Ausgabewerte für die Auswertung entsprechend signifikante Zieldaten erzeugt.



Die Gewinnung der Zieldaten, z. B. von Gaskonzentrationswerten \underline{C} aus Signalwerten \underline{S} , wird mit Hilfe einer mit p_i Attributen parameterisierten Modell-Funktion f_p durchgeführt (Abbildung 3). Die Parameter der Modellfunktion und die Funktion selbst werden während der Analysephase (Lern- bzw. Trainingsphase) über das Sensorverhalten bei verschiedenen Gaskonzentrationen bestimmt. Im Laborbetrieb

Abbildung 3: Modellbildung

können die anhand der Signaldaten und der durch die Modellfunktion f_p ermittelten Konzentrationswerte mit den vorliegenden Konzentrationswerten verglichen und somit die signifikanten Parameter durch ein geeignetes Verfahren gefunden werden. Als signifikant gelten die Parameter, welche einen Rückschluss auf vorliegende reale Gasgemische und deren jeweilige Konzentration erlauben.

Entsprechen die geschätzten Werte \underline{S} nicht den anliegenden realen Signaldaten, werden die geschätzten Zieldaten so lange durch Funktionsbildung und mittels eines Annäherungsverfahrens modifiziert, bis der Fehler zwischen realen und geschätzten Signaldaten hinreichend klein ist.

Der Fehler wird durch die Genauigkeit der Abbildung des Systems durch das Modell bestimmt. Je genauer das Modell, d. h., die analytische Repräsentation ist, desto kleiner ist der Fehler, mit dem die Schätzdaten den Realdaten angenähert werden können. Die Fehlergröße wird so festgelegt, dass durch die Auswertung der Realdaten mittels Schätzdaten eine ausreichende Lösung der Aufgabenstellung, z. B. die Erkennung und Auswahl von Gasgemischen, gegeben ist (Abbildung 4).

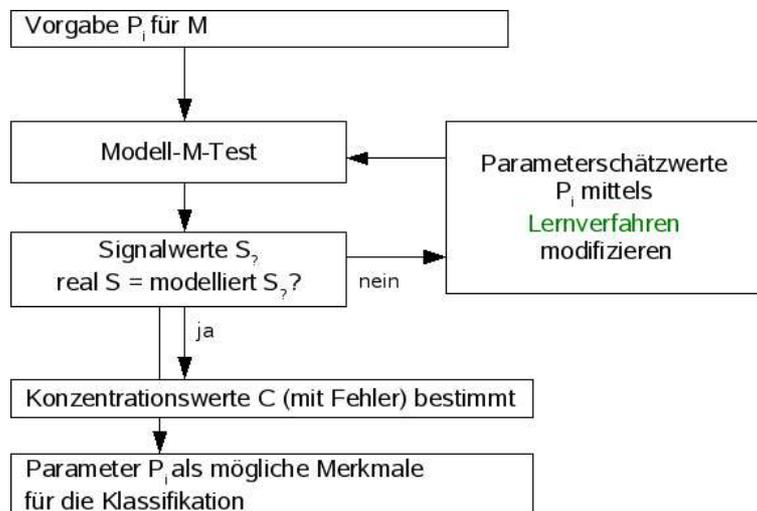


Abbildung 4: Lernverfahren

Die Erstellung eines inversen Modells stellt sich oft als sehr komplex und damit wenig interpretierbar heraus. Es wird daher versucht, den mittels eines Messverfahrens aufgenommenen Funktionsverlauf zu approximieren. Das Modell ist der Approximationsalgorithmus.

Zwischen den Parametern der Modellfunktion und der quantitativen Auswerteeinheit wird im weiteren Verlauf der Arbeiten im Projekt nach einem funktionalen Zusammenhang gesucht.

2.2. Methoden der Sensormodellbildung

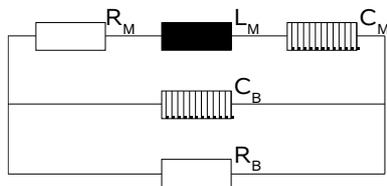
Die Parameter einer Modellfunktion f_p bezeichnen für die Gaskonzentrationsbestimmung signifikante Merkmale durch einen Merkmalsvektor \underline{P} für die Datenauswertung. Als Ansätze für die Bildung von f_p werden genannt:

1. ein erweitertes Butterworth-van-Dyke (BvD) Ersatzschaltbild, welches das elektrische Verhalten des Sensors in der Nähe der Resonanz beschreibt und damit direkte Rückschlüsse auf physikalische Größen erlaubt,
2. die Verhaltensabbildung durch Methoden der "Computational Intelligence", bei der die Sensormodellfunktion zum Beispiel durch ein mehrschichtiges Neuronales Netz erfolgt oder
3. die Abbildung der Signaldaten durch zusammengesetzte Funktionen, z. B. mittels stückweiser Analyse des charakteristischen Funktionsverlaufes (Halbwertsbreiten, Realteilfit der Admittanz).

2.2.1. Gewinnung von Parametern für die Auswertung

BvD-Ersatzschaltbild

Die aktuelle Architektur des Sensors erlaubt eine Analyse des Sensorverhaltens mittels eines erweiterten BvD-Ersatzschaltbildes (Abbildung 5; [1]; [4]). Die im folgenden Schaltbild eingetragenen Elemente aus komplexen Widerständen repräsentieren in analytischer Form die physikalischen Komponenten des Sensors. Durch eine Änderung der Werte der analytischen Ausdrücke sind somit Rückschlüsse auf Änderungen im Sensormaterial möglich.



$$\underline{Y} = \underline{Y}_M + \underline{Y}_B$$

$$\underline{Z}_M = R_m + j \left(\omega L_M - \frac{1}{\omega C_M} \right)$$

$$\underline{Y}_M = \frac{1}{\underline{Z}_M}$$

$$\underline{Y}_B = \frac{1}{R_B} + j \omega C_B$$

Abbildung 5: Modifiziertes BvD-Ersatzschaltbild [1]

Gleichung 1: Berechnung der Admittanz

Die komplexe Admittanz Y errechnet sich nach Gleichung 1 aus der Zusammenfassung der komplexen Leitwerte. Das in [1] beschriebene BvD-Ersatzschaltbild besteht aus Komponenten des Schwingers (m-"motional arm") und der Elektroden (b-"bulk").

Aus Real- und Imaginärteil der Impedanz werden die analytischen Parameter mit einem nichtlinearen Regressionsverfahren mit hinreichender Genauigkeit und bei hinreichend genauer Vorgabe der Startwerte ermittelt. Die Signalwerte der Impedanz (Regressoren) über der Frequenz (Regressanden) bilden die Basis für die in der Levenberg-Marquardt-Methode (LM; nichtlineare Funktionsapproximation, [6]) zu bestimmenden Parameter aus L_M , $C_{M,B}$ und $R_{M,B}$. Über die Ermittlung der minimal möglichen Fehlerquadratsumme zwischen Fit- und Regressionsfunktion und dem Berechnungsaufwand wird im weiteren Verlauf der Bearbeitung eine Verifikation des BvD-Ersatzschaltbildes bzw. des Fit-Algorithmus durchgeführt. Ausgehend von zunächst analytisch bestimmten Startparametern ermittelt der LM-Algorithmus die hinreichend genauen Parameter im Bereich eines bestimmten zugelassenen Fehlerintervalls.

Neuronales Netz

Das charakteristische Impedanzspektrum des Absolutwertes der Impedanz im logarithmischen Maßstab kann mittels eines neuronalen Netzes approximiert werden. Als spezieller "Funktionsapproximator" für die Sensormodellbildung ist die Verwendung eines RBF- bzw. HBF-Netzes (Radiale-Basisfunktion-Netz, Hyper-Basisfunktion-Netz) möglich.

RBF-Netze sind vorwärts gerichtete neuronale Netze, bestehend aus Eingabeschicht, verdeckter Schicht und Ausgabeschicht. Die Neuronen der versteckten Schicht besitzen radialsymmetrische Aktivierungsfunktionen. Diese sind im mathematischen Sinne die Basisfunktionen eines Funktionensystems zur Approximation von (mehrdimensionalen) Funktionen anhand von Werten der Stützstellen. Die Ausgabe (Funktionswert) wird aus der Summe der gewichteten Ausgaben der Aktivierungsfunktionen gebildet (Gleichung 2; [3]).

$$f(X) = \sum_i c_j h_j(q_{i,j}, p_j)$$

Gleichung 2: zu approximierende Funktion

Bei der Approximation mit Zentrumsfunktionen h_j (Gleichung 3) wird unabhängig von der Anzahl der Trainingsmuster eine Anzahl von Zentren, Neuronen der verdeckten Schicht, gewählt. Als Klasse der Zentrums- bzw. radialsymmetrischen Funktion wird eine in der Parameterdimension reduzierte Form der Lorentzfunktion (siehe "Partieller Fit-Lorentzfunktion") eingesetzt:

$$h_j(q_{i,j}, p_j) = \frac{p_j}{2 * \pi} \frac{1}{q_{i,j}^2 + (0.5 p_j)^2}$$

Gleichung 3: modifizierte Lorentzfunktion für RBF-Neuronen

Der Parameter $q_{i,j}$ ist die L_2 -Norm zwischen Zentrum w_j und einem x-Wert x_i . Die Koeffizienten (Gewichte) c_j , Zentren w_j und Amplitudenwerte p_j müssen zunächst hinreichend genau initialisiert und mittels eines Lernverfahren gefunden werden (Kalibrierungsphase des neuronalen Netzes).

Für den Verlauf des Impedanzspektrums werden in der Initialisierung zwei Zentren (w_1, w_2) jeweils beim Minimum und Maximum der Signalwerte der Kurve bestimmt. Die Parameter (p_1, p_2) bestimmen neben den Gewichten (c_1, c_2) die Breite und Höhe der jeweiligen Lorentzkurve und werden mit der Größe des Abstandes der jeweiligen x-Werte, z. B. $(x(y_{MAX}) + x(0.5y_{MAX})) - (x(y_{MIN}) + x(0.5y_{MIN}))$, an denen der y-Wert am Maximum bzw. Minimum auf die Hälfte abgesunken bzw. angestiegen ist, bestimmt. Die Gewichte werden während des Lernverfahrens berechnet.

Das Approximationsproblem kann nach obiger Gleichung 2 in Matrizenformulierung auch formuliert werden als:

$$H * C = Y$$

Gleichung 4: mathematische Formulierung des Approximationsproblems

Das Gleichungssystem ist überbestimmt, da die Dimension der Datenpunkte größer ist als die der zu bestimmenden Koeffizienten und Zentrumsfunktionsparameter. Es kann daher nicht exakt gelöst werden. Der Approximationsalgorithmus versucht die Summe der Quadrate der Fehler aller Trainingsmuster zu minimieren, so dass die Koeffizienten berechnet werden können mit:

$$C = H^{PLUS} * Y$$

Gleichung 5: Berechnung der (Gewichts-)Koeffizientenmatrix

H^{PLUS} ist die "Moore-Penroe-Pseudoinverse" der Matrix H . Sie wird berechnet aus:

$$H^{PLUS} = (H^T H)^{-1} H^T$$

Gleichung 6: Berechnung der "Moore-Penroe-Pseudoinverse"

Ein Lernverfahren nähert die Parameter unter Verringerung des globalen Fehlers (Gleichung 7) an die tatsächlichen Werte an.

$$E = \frac{1}{2} \sum_i (y_i - f_i(x_i, a, \dots))^2$$

Gleichung 7: Globale Fehlerfunktion

Für jeden Parametertyp wird eine eigene Lernrate festgelegt. Das Lernen erfolgt mit einer maximalen Anzahl von Iterationen in Epochen, d. h., nach jeder Epoche mit einer maximalen Anzahl von Iterationen/Lernschritten oder nach Erreichen eines Konvergenzkriteriums werden die für jeden Parameter gesondert einzustellenden Lernraten auf den Anfangswert zurückgesetzt. Damit wurde eine schnellere Konvergenz erreicht. Als Konvergenzkriterien können z. B. das Unterschreiten einer bestimmten Größe des globalen Fehlerwertes oder das Unterschreiten des Betrages der Differenz der Änderungen des globalen Fehlers in zwei aufeinander folgenden Iterationen festgelegt werden.

Die durch die RBF-Approximation gewonnenen Parameter müssen auf Signifikanz überprüft und können ebenfalls als Merkmalsvektoren einer nachfolgenden Klassifikation zu Gaskonzentrationsbestimmung zugeführt werden.

Partieller Fit

Bei der Ermittlung signifikanter Merkmale durch zusammengesetzte Funktionen werden Teilbereiche der Signaldaten (x_n, y_n) für eine vereinfachte Funktionsapproximation verwendet und in den folgenden Abschnitten vorgestellt.

Parabelfunktion

Angenommen wird, dass die Parameter $a...c$ der Parabelfunktion f_{Pa} ,

$$f_{Pa}(x, a, b, c) = ax^2 + bx + c$$

Gleichung 8: Parabelfunktion

die den Teilbereich um das Maximum der Amplitude des Realteils der Admittanz beschreiben kann, signifikante Merkmale bezeichnen.

Es werden die Signaldaten für die Funktionsapproximation von $x_{MAX} - 0.5 * x_{MAX}$ bis $x_{MAX} + 0.5 * x_{MAX}$ extrahiert, bei denen der Wert von y_{MAX} auf jeweils 50% abgesunken ist.

Die Parameter der Parabelfunktion werden entsprechend [2] ermittelt.

Lorentzfunktion

Angenommen wird, dass die Parameter $a...e$ der Lorentzfunktion f_{Lo} ,

$$f_{Lo}(x, a, b, c, d, e) = a \left[1 + \left(\frac{2 * (x - b)}{c} \right)^2 \right]^{-d} + e$$

Gleichung 9: Lorentzfunktion

die den Teilbereich um das Maximum der Amplitude des Realteils der Admittanz

beschreiben kann, signifikante Merkmale bezeichnen.

Die Signaldaten, bei denen der Wert von y_{MAX} auf jeweils 50% abgesunken ist, werden für die Funktionsapproximation von $x_{MAX}-0.5*x_{MAX}$ bis $x_{MAX}+0.5*x_{MAX}$ extrahiert. Die Parameter der nicht linearen Funktion ermittelt der LM-Approximationsalgorithmus innerhalb eines bestimmten Konfidenzintervalls.

Die Anfangsinitialisierung der Parameter in der Nähe der approximierten Werte bedarf einer Vorab-Analyse des Funktionsverlaufs der Signaldaten, da der LM-Algorithmus sonst nicht oder nur langsam konvergiert.

2.2.2. Ergebnisse

Entsprechend der Sensormodellbildung werden im Folgenden die Ergebnisse für Messungen mit Einzelgasen H_2 und CO sowie erste Resultate bei Einstellung eines H_2/CO -Gemisches vorgestellt. Die Einstellung der Gase fand jeweils bei konstantem Sauerstoffpartialdruck durch eine entsprechende Regelung statt. Die Einstellung des Gemisches erfolgte durch kurzzeitige Zuführung von CO . Um eine konstante Betriebstemperatur zu ermöglichen und die Abweichungen von Messungen, die durch eine variable Temperatur hervorgerufen werden, zu vermeiden (Dimensionsreduktion), wird der Labortest in einem Heizofen bei Temperaturstufen von 400, 600 und 800°C durchgeführt. Ein spezielles Impedanzspektrometer ermittelt die Spektren zur Datenauswertung im Arbeitsbereich.

BvD

Durch die Auswertung der Signalwerte kann über eine charakteristische Verschiebung der Parameter ein erster Rückschluss auf das physikalische Verhalten der Sensorschicht bei Vorhandensein entsprechender Gasmische und Gasmischkonzentrationen am Sensor getroffen werden.

Neuronales Netz

Mittels des oben beschriebenen (minimalen) RBF-Netzwerkes wurden die Parameter w_i , c_i und p_i ermittelt. Die folgende Abbildung 6 zeigt einen Fit, der sich aus zwei Lorentzfunktionen zusammensetzt:

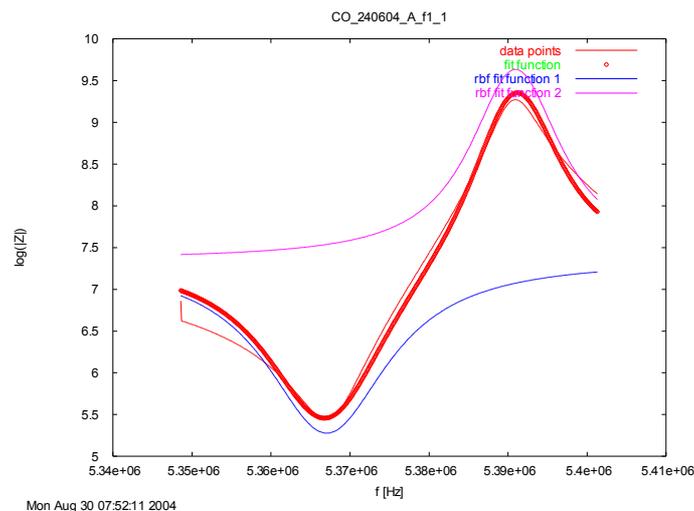


Abbildung 6: RBF-Fit ("fit function")

Interessant sind hierbei zunächst die Parameter der Lage der Zentren w_i (m_1 und m_2 am Minimum und Maximum der Impedanz), mit denen bereits Betrachtungen zur

Selektivität durchgeführt werden können. Die normierte Darstellung in Abbildung 7 zeigt deutlich die Trennung der Einzelgase CO und H₂.

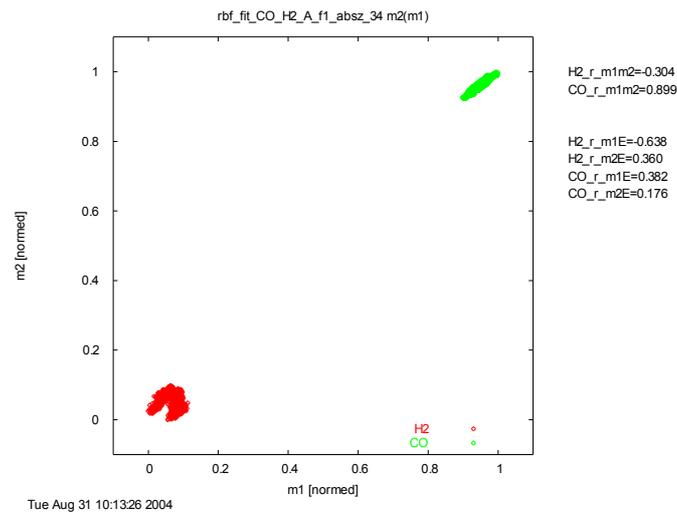


Abbildung 7: Merkmalsraum für zwei (Rein-)Gaskonzentrationen H₂ und CO aus RBF-Fit

Der Funktionsfit mittels RBF-Netz ist im Vergleich zu Methoden des partiellen Fits zeitlich intensiver, so dass auch bzgl. der Gasmischanalyse weitere Untersuchungen zunächst mittels partieller Fit-Methoden durchgeführt werden.

Partieller Fit

Die durch einen Parabel- bzw. Lorentzfunktion-Fit gewonnenen Parameter zeigen eine Selektivität des Sensors für das bisher vorliegenden Datenmaterial zur Auswertung für die jeweilige Erkennung von H₂ oder CO.

Parabelfunktion

Die maximale Amplitude A und die entsprechende Serienresonanzfrequenz f_s eines Parabel-Fits auf den Realteil der Admittanz bezeichnen Parameter in einem zweidimensionalen Analyse- und Parameterraum. In der normierten Darstellung zeigt sich eine deutliche Trennung der Signale für H₂ (links-unten) und CO (rechts-oben) nach Abbildung 8:

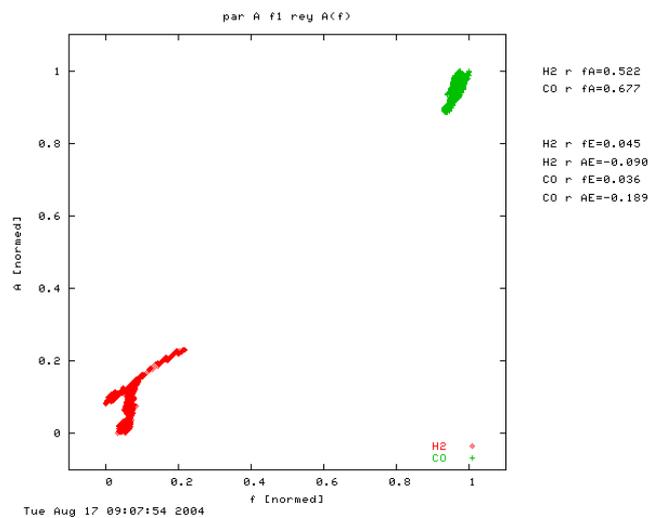


Abbildung 8: Merkmalsraum für zwei (Rein-)Gaskonzentrationen H₂ und CO aus Parabel-Fit

In diesem Beispiel sind die Merkmale m_1 (f) und m_2 (A) für H₂ mit 0.522 und für CO mit 0.677 korreliert.

Aus der Summe der Differenzen der durch die Sensormodellbildung ermittelten Funktionswerte und der tatsächlichen Signalwerte ergibt sich ein globaler Fehler.

Es wurde der jeweilige Korrelationskoeffizient der Merkmale der spezifischen Gaskonzentration zum Fehler E durch einen Parabelfunktionsfit ermittelt:

	H ₂	CO
$f-E$	0,045	0,036
$A-E$	-0,090	-0,189

Tabelle 1: Korrelationskoeffizienten Parabel-Fit

Die Werte zeigen keine lineare Abhängigkeit der als Merkmal angenommenen Parameter zum Fehler E.

Die Gruppierungen sind nun mit Hilfe eines Klassifikationsalgorithmus oder eines anderen funktionalen Zusammenhangs zu untersuchen. Die qualitative Bestimmung des Gases erfolgt z. B. durch Minimalabstandsbestimmung eines Parametervektors zum metrischen Zentrum einer Gruppe.

Die weitergehende quantitative Bestimmung wird durchgeführt, wenn zusätzliche Datensätze bei Gaskonzentrationsgemischen zur Analyse vorliegen.

Es ist zu erwarten, dass für die Detektion mehrerer Gase und Gasgemische einen mehrdimensionalen Merkmalsraum aufspannt. Erste Datenanalysen von Gasgemischen aus O₂, H₂ und CO ergeben die in Abbildung 9 dargestellten Abhängigkeiten mittels der Funktion "Phasenwinkel über Amplitude der Impedanz bei Serienresonanz f_s " bzw. $\Phi(|Z|_{f_s})$ für eine Elektrodengeometrie A ([1]) bereits beim nicht-temperaturkompensierten ersten Oberton fl.

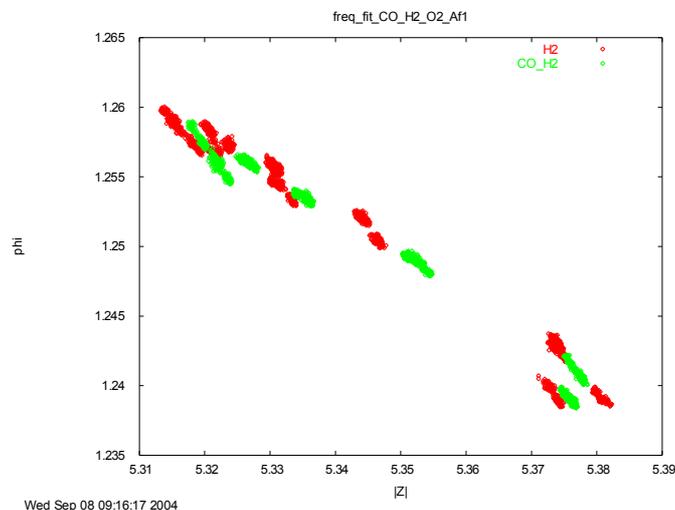


Abbildung 9: Merkmalsraum Gaskonzentrationen aus O₂, H₂ und CO

Auf verschiedenen Stufen von Sauerstoffpartialdruck (10^{-20} bis 10^{-15}) wird jeweils zu einem O₂-H₂-Gemisch CO hinzugegeben. Es erfolgt eine Verschiebung des O₂-H₂-CO-Gemisches. Die Parameter können zumindest als Merkmale für die qualitative Analyse verwendet werden, da sie eine Gemischänderung anzeigen.

Lorentzfunktion

Die Analyse der Daten mit einer Lorentzfunktion bestätigen die durch den Parabelfit gewonnenen Ergebnisse.

3. Clusterbildung und Klassifikation

Die durch die speziellen Sensormodelle ermittelten Parameter können bei genügender Signifikanz als Merkmalsvektoren in den Algorithmus zur Bildung von unscharfen Klassen eingehen (Lernphase). Signifikante Parameter bezeichnen jene, die zur Detektion von Gasgemischen durch den Sensor beitragen. Bei einer ausreichend hohen Anzahl von Vektoren werden im Merkmalsraum Cluster gebildet. Es wird davon ausgegangen, dass im Zeitraum der Lernphase verschiedene (bekannte) Gaskonzentrationen C_L eine variierende Veränderung der Parametervektoren P_i hervorrufen (Gasselektivität) und dies anhand der Signaldaten S_L durch einen Modellvergleich erkennbar wird. Die unterschiedliche Lage der Clusterzentren und das Vorhandensein bekannter Gaskonzentrationsmengen repräsentiert Cluster (im Idealfall) von Einzelgasen bzw. Gasgemischen, die somit bezeichnet ("gelabelt") werden können.

Nach Abschluss der Lernphase besitzen neue, dem Klassenraum (Clusterraum) zugeführte Signalvektoren S_A eine bestimmte Lage zu den Clusterzentren und damit eine bestimmte Zugehörigkeit. So kann beispielsweise prognostiziert werden, dass die Gaskonzentration C_A am Sensor vorhanden ist, zu deren Clusterzentrum der Vektor entsprechend der eingesetzten Metrik am nächsten liegt. Der gewichtete Abstand sollte dabei gleichzeitig eine Aussage über die Konzentrationsmenge zulassen. Die folgende Abbildung 10 zeigt die Klassifikation in der Lern- und Arbeitsphase:

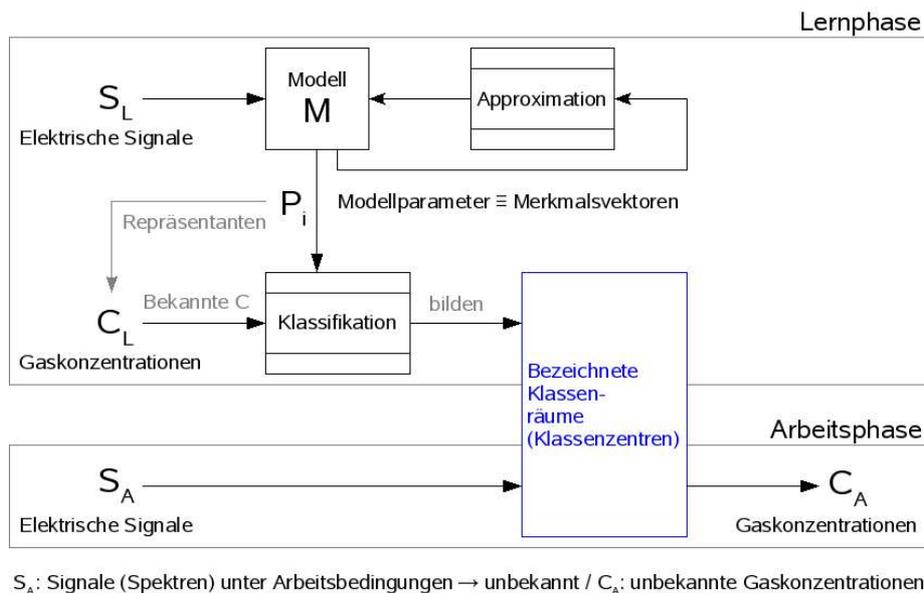


Abbildung 10: Klassifikation in der Lern- und Arbeitsphase

3.1. Methoden

Im Folgenden werden Methoden vorgestellt, die Strukturen/Gruppen in (zweidimensionalen) Daten bei bekannter Anzahl von Gruppierungen finden. Daten, die einer Gruppe zugeordnet werden, sind sich ähnlich, Daten unterschiedlicher Gruppen, möglichst unähnlich. Es wird ein metrisches Maß für die Ähnlichkeit oder Unähnlichkeit von Daten (1 Datensatz \equiv 1 Datum) verwendet ([8]). Eine Gruppierung wird Cluster genannt.

3.1.1. Fuzzy-c-Means (FCM)

Der FCM erkennt kreisförmige Punktwolken im 2-dimensionalen Raum. Die als gleichgroß zu betrachtenden Cluster werden durch ihren Mittelpunkt (Prototyp)

dargestellt. Zwischen Datum und Prototyp wird als Ähnlichkeitsmaß der Euklidische Abstand verwendet, d. h., je kürzer der Abstand zwischen einem Datum und einem Prototyp ist, desto höher ist die Zugehörigkeit des Datums zum Cluster. Bei einer zufälligen Anfangsinitialisierung der Clusterzentren sucht der FCM-Algorithmus nach der optimalen Lage der Zentren im Datenraum durch Minimierung einer Kostenfunktion:

$$J = \sum_{i=1}^C \sum_{k=1}^N u_{i,k}^{FF} d_{i,k}^2$$

Gleichung 10: Kostenfunktion des Clusteralgorithmus

C bezeichnet die Anzahl der zu findenden Cluster, N die Anzahl der Datenpunkte und FF den Fuzzifizier. Um leere Cluster zu verhindern, muss mindestens ein Datum eine Zugehörigkeit größer Null zu diesem besitzen. Die Summe der Zugehörigkeiten jedes Datums zu allen Clustern ist Eins (probabilistisches Clustern).

Bis zur Minimierung der Kostenfunktion J führt der Algorithmus die Schritte in Gleichung 11 aus:

Berechnen der Abstände mit

$$d_{i,k} = \sqrt{(\bar{x}_k - \bar{v}_i)}$$

Berechnen der Zugehörigkeit der Daten zu den Clustern

$$u_{i,k} = \frac{1}{\sum_{j=0}^C \left(\frac{d_{i,k}}{d_{j,k}} \right)^{\frac{2}{FF-1}}}$$

Berechnen der neuen Clusterzentren

$$\bar{v}_i = \frac{\sum_{k=1}^N u_{i,k}^{FF} \bar{x}_k}{\sum_{k=1}^N u_{i,k}^{FF}}$$

Gleichung 11: FCM-Algorithmus

3.1.2. Gustafson-Kessel (GK)

Zur Erkennung ellipsoider Cluster muss die im FCM verwendete euklidische Metrik verändert werden, da mit dem FCM eine automatische Anpassung der einzelnen Clusterformen nicht möglich ist. Jedes Cluster wird zusätzlich über eine positiv-definite Matrix A charakterisiert, die eine eigene Norm induziert ([8]):

$$\|x\|_A := \sqrt{(x^T A x)}$$

Gleichung 12: GK-Matrix-Norm

Es wird ein konstantes Clustervolumen angenommen (Gleichung 13), d. h., es ist lediglich die Clusterform, nicht aber die Größe veränderbar.

$$\det(A) = 1$$

Gleichung 13: Volumenvorschrift

Die Berechnung der Clusterzentren durch den GK-Algorithmus erfolgt unter Minimierung der Kostenfunktion J (Gleichung 10). Die Zugehörigkeitsgradwerte $u_{i,k}$ eines Datums zu einem Cluster werden wie in Gleichung 11 berechnet. Für die Ermittlung der Prototypen führt der GK folgende Schritte aus:

Anfangsinitialisierung der Abstände $d_{i,k}$ und

Zugehörigkeitsgradwerte $u_{i,k}$ mittels einer FCM – Iteration

Berechnen der Fuzzy – Kovarianzmatrix mit

$$A_k = \left(\det(S_k^p) \right)^{-1} S_k^{-1} \text{ und } S_k = \sum_{i=1}^N u_{i,k} (\bar{x}_i - \bar{v}_k) (\bar{x}_i - \bar{v}_k)^T$$

Berechnen der Abstände mit

$$d_{i,k} = (\bar{x}_i - \bar{v}_k)^T A_k (\bar{x}_i - \bar{v}_k)$$

Berechnen der neuen Zugehörigkeitsgradwerte $u_{i,k}$ (s. o. FCM)

Berechnen der neuen Clusterzentren (s.o. FCM)

Gleichung 14: GK-Algorithmus

p ist die Dimension des Datenraumes ($p=2$).

3.1.3. Gath-Geva (GG)

Als Erweiterung des GK beachtet der GG auch die Größe und Dichte der Cluster. Die Daten werden als Realisierungen p -dimensionaler normalverteilter Zufallsvariable interpretiert.

Die Normalverteilung N_k mit dem Erwartungswert p_k und der Kovarianzmatrix A_k wird mit der a-priori Wahrscheinlichkeit P_k zur Erzeugung eines Datums ausgewählt. Die a-posteriori Wahrscheinlichkeit, mit der das Datum x_i durch die Normalverteilung N_k erzeugt wurde, beträgt somit nach [3], S. 48ff.:

$$\frac{P_k}{(2\pi)^{\frac{p}{2}} \sqrt{\det(A_k)}} \exp\left(-\frac{1}{2} (\bar{x}_i - \bar{v}_k)^T A_k^{-1} (\bar{x}_i - \bar{v}_k)\right)$$

Gleichung 15: a-posteriori Wahrscheinlichkeit

Die Wahrscheinlichkeit der Zugehörigkeit eines Datums x_i zu einer Normalverteilung N_k wird nach Gleichung 16 berechnet mit:

$$P_k = \frac{\sum_{i=1}^N u_{i,k}^{FF}}{\sum_{i=1}^N \sum_{l=1}^C u_{i,l}^{FF}}$$

Gleichung 16: a-priori Wahrscheinlichkeit der Zugehörigkeit eines Datums zu einer Normalverteilung

Die Distanzfunktion berechnet sich entsprechend aus:

$$d^2(\bar{x}_i, \bar{v}_k) = \frac{1}{P_k} \sqrt{\det(A_k)} \exp\left(\frac{1}{2} (\bar{x}_i - \bar{v}_k)^T A_k^{-1} (\bar{x}_i - \bar{v}_k)\right)$$

Gleichung 17: Distanzfunktion

Für die Ermittlung der Prototypen unter Minimierung der Kostenfunktion führt der GG die Schritte aus:

Berechnen von P_k

Berechnen der Kovarianzmatrix mit

$$A_k = \frac{\sum_{i=1}^N u_{i,k}^{FF} (\bar{x}_i - \bar{v}_k) (\bar{x}_i - \bar{v}_k)^T}{\sum_{i=1}^N u_{i,k}^{FF}}$$

Berechnen der neuen Prototypen mit

$$v_k = \frac{\sum_{i=1}^N u_{i,k}^{FF} x_i}{\sum_{i=1}^N u_{i,k}^{FF}}$$

Gleichung 18: GK-Algorithmus

Die Distanzfunktion ist umgekehrt proportional zur a-posteriori Wahrscheinlichkeit, da eine geringe Distanz eine hohe Zugehörigkeit und eine große Distanz eine niedrige Zugehörigkeit eines Datums zu einem Cluster bedeutet.

3.1.4. Ergebnisse

Die Datenanalyse der Einzelgastestreihen ergab eine eindeutige Aufteilung in zwei verschiedene Bereiche durch die Funktion $f_{res}(A)$ (siehe Abbildung 8). Im Hinblick auf die Anwendung verschiedener Clusteralgorithmen zum Auffinden von Gruppen in Daten wurde ein Test mit klassischen Clusterverfahren wie Fuzzy-C-Means (FCM), Gustafson-Kessel (GK), Gath-Geva (GG) und deren achsenparalleler Entsprechungen AGK und AGG durchgeführt (siehe [5]). Die Bestimmung der Clusterzentren erfolgte in Abbildung 11 mittels F2M-Algorithmus und 100 Iterationen. Ausgehend von zufällig initialisierten Zentren $S1$ und $S2$ werden somit die Endpunkte in $E1$ und $E2$ ermittelt.

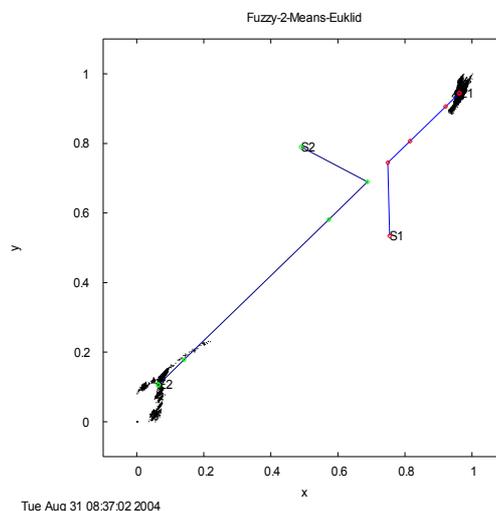


Abbildung 11: F2M-Analyse der Daten aus Abbildung 8

Die tabellarische Übersicht (Tabelle 2) zeigt zudem Ergebnisse für Clusterzentren der Daten mittels Fit durch die Lorentzfunktion und Funktionsapproximation durch das RBF-Netz (s. o.).

Algorithmus	Parabel-Fit	Lorentz-Fit	RBF	Zeit (sec)	P/L/R
FCM	$v_1 = (0.962, 0.945)$	$v_1 = (0.937, 0.921)$	$v_1 = (0.062, 0.048)$	1/1/1	
	$v_2 = (0.064, 0.107)$	$v_2 = (0.063, 0.098)$	$v_2 = (0.945, 0.960)$		
GK	$v_1 = (0.962, 0.947)$	$v_1 = (0.937, 0.921)$	$v_1 = (0.062, 0.048)$	5/1/1	
	$v_2 = (0.064, 0.106)$	$v_2 = (0.062, 0.098)$	$v_2 = (0.945, 0.960)$		
AGK	$v_1 = (0.962, 0.947)$	$v_1 = (0.937, 0.921)$	$v_1 = (0.062, 0.048)$	2/1/1	
	$v_2 = (0.064, 0.106)$	$v_2 = (0.062, 0.098)$	$v_2 = (0.945, 0.960)$		
GG	$v_1 = (0.962, 0.947)$	$v_1 = (0.937, 0.921)$	$v_1 = (0.062, 0.048)$	18/1/1	
	$v_2 = (0.065, 0.112)$	$v_2 = (0.062, 0.098)$	$v_2 = (0.945, 0.960)$		
AGG	$v_1 = (0.962, 0.947)$	$v_1 = (0.937, 0.921)$	$v_1 = (0.062, 0.048)$	16/1/1	
	$v_2 = (0.065, 0.112)$	$v_2 = (0.062, 0.098)$	$v_2 = (0.945, 0.960)$		

Tabelle 2: Clusterzentren bei unterschiedlichen Sensormodellen

Die Sensormodellbildung zeigt in der Lage der für die Clusteranalyse normierten Daten weniger verrauschte Ergebnisse, so dass keine, für die qualitative Analyse wesentliche, veränderte Position der Zentren durch verschiedene Clusteralgorithmen erkennbar wird. Bereits der FCM-Algorithmus findet die entsprechenden Zentren. Die Anwendung von Clusteralgorithmen auf, und eine entsprechende Analyse der, Gaskonzentrationsdaten wird Bestandteil des Vortrages zu dieser Abhandlung sein.

3.2. Diskussion

Die Sensormodelle und damit auch die Parameterbildung müssen in Hinblick auf Aufwand, Rechenzeit, Speicherplatzbedarf usw. untersucht werden.

Die Bildung eines Sensormodells, welches einem nichtlinearen Verlauf folgt, könnte sich als zu komplex, also als zu wenig interpretierbar und überschaubar erweisen. Gleichfalls könnte die Funktionsapproximation nicht der geforderten Genauigkeit entsprechen und damit für eine quantitative Analyse unbrauchbar werden. Mittels entsprechender Transformationen bilden Teilmodelle der Signaldaten über signifikanten Abschnitten der Funktionswerte der Impedanzspektren gleichwertige und auswertbare (Teil-) Merkmalsvektoren. Die aus den transformierten Signaldaten gewonnenen Vektoren müssen einem Eignungstest bezüglich der Merkmalsbildung unterliegen, da resultierende Merkmale einen neuen Merkmalsvektor zur quantitativen und qualitativen Analyse bezeichnen.

Die Funktionsapproximation der Teilmodelle vereinfacht sich durch die Anwendung von Verfahren, wie zum Beispiel der Approximation von Funktionsabschnitten. Das Impedanzspektrum wird segmentiert. Jedes Segment enthält eine Teilkurve, z. B. eine Parabel- bzw. Lorentzfunktion. Ein Merkmalsvektor wird aus den Koeffizienten der durch die Annäherung ermittelten Funktion zusammengesetzt.

Nachteilig bei der Teilmodellbildung ist in erster Linie die fehlende physikalische Interpretierbarkeit der Ergebnisse der Parameter. Für die Clusterbildung, Interpretation und Zuordnung neuer unbekannter Signaldaten zu Gaskonzentrationen über einem gleichbleibenden Modell sind sie zunächst jedoch geeignet.

4. Literaturverzeichnis

- [1] Fritze, H.; Richter, D.: *BMBF-Zwischenbericht 02.2004*. Clausthal-Zellerfeld: TU-Clausthal 2004.
- [2] Voß, Werner u.a.: *Taschenbuch der Statistik*. Leipzig: Fachbuchverlag 2000.
- [3] Zell, Andreas: *Simulation Neuronaler Netze*. Bonn, Paris: Addison-Wesley 1994.
- [4] Fritze, H. u.a.: *High Temperature Bulk Acoustic Wave Properties of Languasite*. Clausthal-Zellerfeld: TU-Clausthal 2003.
- [5] Fritze, H.; Richter, D. u.a.: *BMBF-Zwischenbericht 01.2004*. Clausthal-Zellerfeld: TU-Clausthal 2004.
- [6] Press, W. H. u.a.: *Numerical Recipes in Pascal. The Art of Scientific Computing*. Cambridge: University Press 1996.
- [7] Richter, D.: *Charakterisierung TiO_2 , CeO_2 , und $YBa_2Cu_3O_{7-x}$ beschichteter Hochtemperaturresonatoren*. Diplomarbeit. Clausthal-Zellerfeld 2003.
- [8] Höppner, F.; Klawonn, F.; Kruse, R.: *Fuzzy-Clusteranalyse. Verfahren für die Bilderkennung, Klassifikation und Datenanalyse*. Braunschweig, Wiesbaden: Vieweg 1997.

Optimierung von Aufstellungsentwürfen von Chemieanlagen durch Constraint Programming

Hanna Köpcke

Universität Dortmund, Fachbereich Informatik,
Lehrstuhl für künstliche Intelligenz, 44221 Dortmund

Tel.: (0231) 755-6033

Fax: (0231) 755-5105

E-Mail: koepcke@ls8.cs.uni-dortmund.de

Aufgabe der Aufstellungsplanung ist die Platzierung sämtlicher für eine Chemieanlage relevanten, verfahrenstechnischen Ausrüstungen innerhalb und im Nahbereich außerhalb einer Stahlbaukonstruktion. Dabei müssen Anforderungen an die Lage der Komponenten im Stahlbau und zueinander berücksichtigt werden. Diese Anforderungen sind bestimmt durch eine Vielzahl von Vorgaben der einzelnen Ausrüstungen, die sich u.a. durch die Betriebsweise der Ausrüstung, technische Spezifikationen und Vorgaben des Anlagenbetreibers ergeben. Anforderungen an die Positionierung von Ausrüstungen sind immer geprägt durch eine Reihe von konkurrierenden und widersprüchlichen Bedingungen, die vom Planungsingenieur abgewogen und gelöst werden müssen.

Zur Lösung dieser Problemstellung wurde am Lehrstuhl für Anlagentechnik der Universität Dortmund ein System zur computerunterstützten Aufstellungsplanung entwickelt. Für ein konkretes Planungsszenario (vorgegebenes Baufeld inklusive des Anlagengerüsts und der Anlagenwege sowie eine Anzahl zu platzierender Equipmentmodelle) werden die zu erfüllenden Anforderungen unter Verwendung einer im Programm hinterlegten Regelbasis abgeleitet. Die inferierten Anforderungen, d.h. die zu berücksichtigenden Abhängigkeiten zwischen einzelnen Equipments und dem Stahlgerüst, werden in einem zweiten Schritt in eine numerische Form übertragen. Auf der Basis dieser als Anziehungen zwischen Equipments und Anlagengebäude interpretierbaren numerischen Anforderungsrepräsentation wird nun ein heuristisches Suchverfahren (ein *Simulated Annealing Algorithmus* (SA)) zur Minimierung der Regelverstöße bzw. zur Maximierung der erfüllten Anforderungen verwendet. Die bisherigen Arbeiten haben gezeigt, dass eine Platzierung von Equipments im Anlagengebäude mit dem beschriebenen Verfahren möglich ist. Allerdings ist diese Methode für große Anlagen zu aufwändig und erfordert viel Erfahrung bei der Festlegung der SA-Strategieparameter, insbesondere der Abkühlvorschriften.

Aus diesem Grunde wurde im Rahmen des Teilprojektes C11 „Optimierung von Aufstellungsentwürfen für Chemieanlagen“ des Sonderforschungsbereiches 531 „Design und Management komplexer technischer Systeme mit Methoden der Computational Intelligenz“ (SFB CI) das Problem der Aufstellungsplanung als Konfigurationsproblem formalisiert und Verfahren des Constraint Logic Programming (CLP) als alternative Lösungsstrategien untersucht. Die realisierten Lösungsstrategien beruhen auf Methoden zur partiellen Constraint-Erfüllung und Fuzzy-Constraints.

Optimierung NH₄OH-Eindüsung einer SCR-Anlage mittels Fuzzy Control am Beispiel der MVA Flötzersteig, Wien

Erich Pawelka¹, Marcus Kuhn¹, Ch. Gierend²

¹MVA Flötzersteig, Wien

²IPG GmbH, Grünstadt

1 Motiv der Arbeit

Die MVA Flötzersteig, Fernwärme Wien, war mit der bisherigen Regelung zur Eindüsung von Ammoniakwasser vor Katalysator unzufrieden. Mit Einführung der Fuzzy - Feuerleistungsregelung an allen drei Kesseln der Anlage Flötzersteig wurde mit gesteigerter Leistungsfähigkeit der Kessel neben der Frischdampferzeugung auch im Wesentlichen die Rauchgasmenge angehoben. Die notwendige Ammoniakwassermenge ist im Schnitt von 1,8 kg/Mg (Müll) im Jahresmittel auf bis zu 2,2 kg/Mg (Müll) angestiegen, wobei der Schlupf sehr schwankend zwischen 3 und 5 mg/m³ pendelte. Die sichere Einhaltung der Grenzwerte für das Reingas (Stickoxide und Schlupf) waren gefährdet. Die Überlegungen zur Einführung einer fuzzybasierten Eindüsung gründeten wesentlich auf den nachfolgenden Erläuterungen zum Thema.

Die erfolgreiche Applikation innovativer Technologien in einfachen Regelkreisen, insbesondere die Anwendung von Fuzzy Logic, weckte in der letzten Zeit die Hoffnung, solche Technologien ohne weiteres für Systemlösungen auf komplexe Prozesse zu übertragen. Technische und wirtschaftliche Grenzen der konventionellen Regelungstechnik bei Mehrgrößenproblemen in der Kraftwerkstechnik, speziell in der Rauchgasreinigung, fordern noch effektiver anlagenspezifische Kenntnisse aus den verfahrenstechnisch nachvollziehbaren Reaktionsabläufen in Regelungsentwürfe einzubringen. Dabei bildet verbal vorhandenes Wissen die Basis einer optimal an den Prozess angepassten regelungstechnischen Umsetzung, unter Einbeziehung der verfahrenstechnischen Grundlagen und Zusammenhänge.

Dies gilt vor allem für Prozesse in der thermischen Abfallbehandlung / Abfallverwertung und deren Rauchgasreinigung. Die Einhaltung der gesetzlichen Bestimmungen hinsichtlich der Emissionsgrenzwerte und eine wirtschaftlich nutzbare Auskopplung von Energie in Form von Strom und Dampf sind relevante Kriterien, um Edelbrennstoffe Gas/Öl und Kohle durch die Müllverbrennung zu substituieren. Die Umsetzung bedingt sowohl eine hohe Personalqualifikation zur Betreuung der erweiterten konventionellen Automatisierungstechnik, als auch einen nicht unerheblichen Investitionsumfang. Die Anschaffung einer neuen Automatisierungstechnik zur Verbesserung der Prozessführung und die erwartete Kosteneinsparung bei Personal und Betriebsmitteln sind gegeneinander abzuwägen.

Motiviert zur Einführung von Fuzzy Control, eine gänzlich neue Vorgehensweise zu erproben, entwickelt der Verfahrenstechniker mit eigenem Betriebspersonal der Automatisierungsabteilung eine Regelung zur Ammoniakwasser-Eindüsung (NH₄OH) vor Katalysator, die den anlagenspezifischen Prozessabläufen angepasst wird.

Die Neugestaltung einer so gearteten Regelung zur Ammoniakwasser-Eindüsung wurde in einen entscheidenden Anteil von der Überlegung getragen, auf eigenes Anlagenwissen zurückzugreifen und entscheidend die Anschaffungs- und Folgekosten zu senken. Des Weiteren inspirierte der Gedanke verbal vorhandenes Wissen und bisher verborgenes Wissen zu dokumentieren und anschließend in eine maschinenlesbare Schreibweise zu überführen. Es war Wunsch der Anlagenbetreiber, ein Regelungskonzept in der eigenen Anlage zu entwickeln und anlagenspezifische Besonderheiten jederzeit ohne großen Aufwand anzupassen.

Im Vergleich zu den bisher verwendeten Regelungstechniken hat sich die Fuzzy-Control-Technik besonders bei komplizierten und komplexen Prozessen bewährt, bei denen das Zusammenspiel aller wichtigen verfahrenstechnischen Größen zur Entscheidungsfindung notwendig ist. Der installierte Fuzzy Controller entscheidet dadurch, dass er mehr als fünf bis sieben Prozessgrößen synchron miteinander gewichtet. Hiermit erhält er in einer Art Mittelwertbildung, die der menschlichen Denkweise sehr nahe kommt, den Wert, den die zu regelnde Ausgabegröße annehmen soll. Der Regler greift im Sekundenbereich auf mehr als 500 Regeln zurück, von denen etwa durchschnittlich 25 bis 40 Regeln zum Regelergebnis beitragen.

Ziel der Arbeit: In der Müllverbrennung stellt die außerordentlich inhomogene und zudem zeitlich stark wechselnde Beschaffenheit des Brennstoffs die größte Störgröße dar. Aufgrund großer Schwankungen in der Mülldichte, Müllbeschaffenheit und Müllfeuchte werden viele Müllkessel allein durch Beobachtung des Feuerbetts in der Hauptverbrennungszone manuell gefahren. Die Handfahrweise der Müllkessel erzielte eine minimale Schwankungsbreite der Feuerleistung, besseren Ausbrand und höhere Durchsatzraten gegenüber der konventionellen Regelung aus den Jahren 1991 und 1992. Darüber hinaus konnte das Betriebspersonal das gesamte Prozessverhalten bis hin zur Rauchgasreinigung an einem günstigeren Arbeitspunkt stabilisieren. Der konventionellen Regelungstechnik ist es bis dahin nicht gelungen, diese sehr komplexe, mit mathematischen Algorithmen nicht oder nur unzulänglich zu beschreibende Regelstrecke optimal zu regeln, noch an die Ergebnisse der Handfahrweise anzuknüpfen. Mit Einführung eines jeweils neuen Regelungskonzeptes wurden stets die Erwartungen verbunden zu einer wesentlichen Entlastung des Betriebspersonals und größeren Kosteneinsparungen beitragen zu können. Die Einbindung von Fuzzy verwirklichte dieses Ziel in der Feuerleistungsregelung und wurde den genannten Anforderungen unter Einbeziehung verbal vorhandener Wissensstrukturen gerecht. Gerade ältere Anlagen laufen bei steigenden Müllheizwerten und größeren Heizwertschwankungen Gefahr, die zulässige Feuerraumbelastung, Druckdifferenzüberschreitungen im Kessel und vor Rauchgasreinigung zu überschreiten. Sprunghafte Veränderungen in der Dampfmenge, Rauchgasmenge und Temperaturspitzen sind bei durchschnittlich gestiegenen Müllheizwertenschwankungen deutlich größer. Anlagensicherheitstechnische Umstände zwingen viele Betreiber, die Feuerung unterhalb der thermischen, mechanischen und rauchgastechnischen Auslegung zu fahren [1]. Die mittlere Dampfproduktion hängt entscheidend von der Größe der Frischdampfschwankungen ab. Gelingt es, die mittlere Dampfproduktion durch minimale Schwankungen der Frischdampfmenge zu erhöhen, so führt dies zu einer merklichen Erhöhung des jährlichen Mülldurchsatzes, konstanten Rauchgasvolumen und einer nachweislichen Gewinnoptimierung [2, 3]. Weitere günstige Begleiterscheinungen, abgesehen vom Einsatz an Edeln Brennstoffen für die Stützbrenner, sind nicht zu unterschätzen und finden sich in einem gleichmäßigeren Gesamtrauchgasvolumen wieder, das die Rauchgasreinigung optimiert am Arbeitspunkt

der technischen Auslegung fahren lässt. Dies sollte als primäres Ziel auch für die SCR-Anlage mit Fuzzy umgesetzt werden.

Darstellung der Anlage mit SCR-Katalysator: Die Anlage Flötzersteig ist eine der beiden Hausmüllverbrennungsanlagen der Fernwärme Wien GmbH und verfeuert ca. 200.000 Mg Müll pro Jahr. Sie ist die etwas kleinere und „unbehübschte“ Schwesteranlage der bekannten Müllverbrennung in der Spittelau, verfügt allerdings über die gleiche Technik der Rauchgasreinigung.

Die Anlage ging 1963 in Betrieb und wurde in den 80er und 90er Jahren mit moderner Rauchgasreinigungstechnologie nachgerüstet (Siehe Abbildung 01).

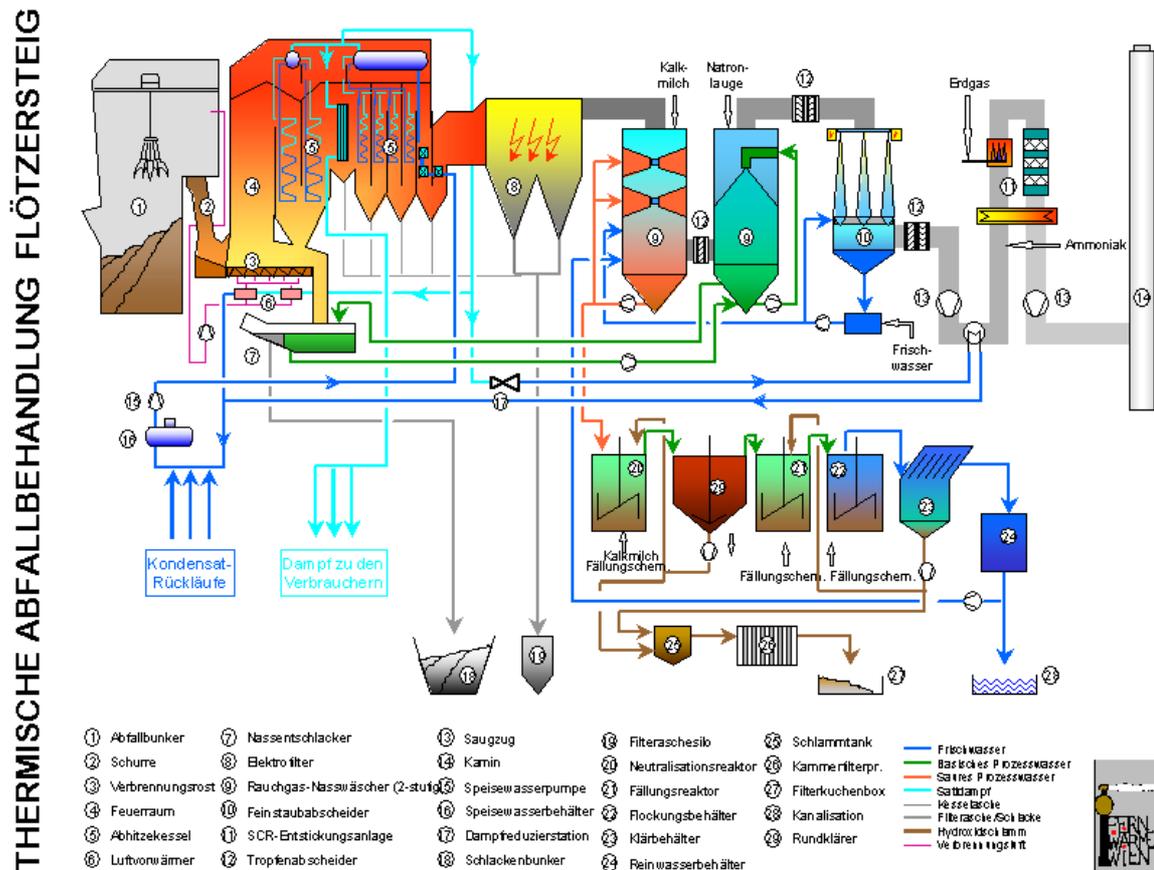


Abbildung 01: Schema der Müllverbrennungsanlage Flötzersteig, Wien

Unter anderem wurden 1991 und 1992 ein neuer Rost und eine neue Feuerleistungsregelung installiert. Die Feuerleistungsregelung kam mit Müll mit konstant gutem Heizwert zufrieden stellend zurecht, reagierte aber schlecht auf Heizwertveränderungen und insbesondere auf Müll mit niedrigem Heizwert, wie er im Sommer durch erhöhten biogenen Anteil und nach Regenperioden vorkam. Sank der Heizwert, kam es zu Überschüttungen am Rost, stieg der Heizwert, traten CO-Spitzen auf [3]. Dies erforderte eine ständige Überwachung des Prozesses durch die Heizer und pro Tag etwa 300 manuelle Eingriffe in die Feuerleistungsregelung. Die Betriebsleitung hat sich in 2001 für eine Fuzzy-Regelung an Kessel 1 und nach erfolgreichem Start in 2002 für Kessel 2 und 3 entschieden. Aufgrund des wesentlich gestiegenen Bedienkomforts, der kaum noch Handeingriffe in die Regelung erforderte und die Heizertätigkeit auf eine reine Überwachungstätigkeit reduzierte, wurde für die Umsetzung der NH_4OH -Eindüsung am Katalysator ebenfalls eine Regelung mit Fuzzy-Technologie umgesetzt.

2 Aufbau und Systemstruktur der Fuzzy-Regelung

Die Systemstruktur beschreibt den Datenfluss in dem Fuzzy System. Eingangsinterfaces fuzzifizieren die Eingangsgrößen. Hierbei werden Analogwerte in Zugehörigkeitsgrade umgesetzt. Dem Fuzzifizieren schließt sich die Fuzzy-Inferenz an: Mit in Regelblöcken festgelegten "wenn-dann" Regeln werden durch die Eingangsgrößen sprachlich beschriebene Ausgangsgrößen festgelegt. Diese werden in den Ausgangsinterfaces durch eine Defuzzifizierung in analoge Größen umgewandelt.

Die folgende Abbildung zeigt die Struktur für dieses Fuzzy-System mit Eingangsinterfaces, Regelblöcken und Ausgangsinterfaces. Die Verbindungslinien symbolisieren hierbei den Datenfluss.

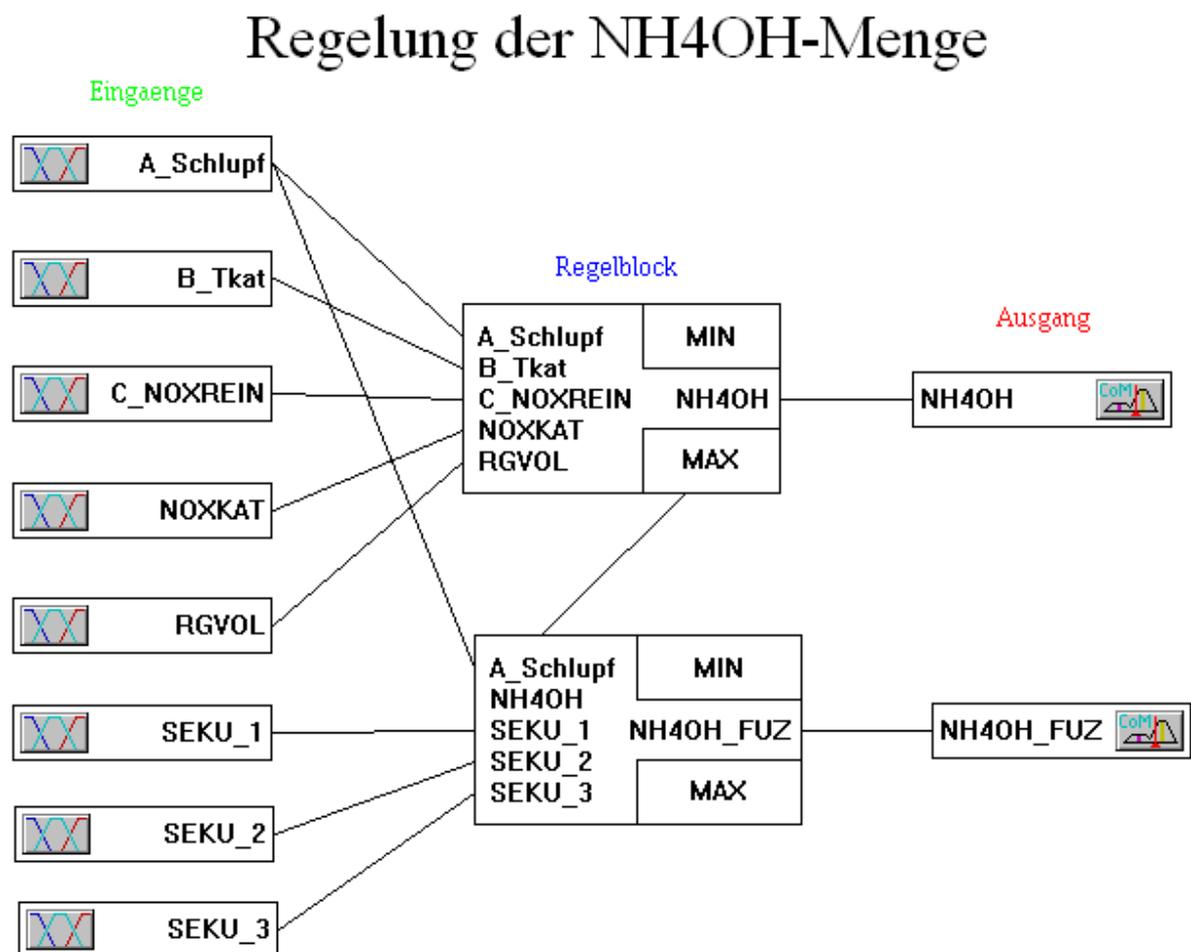


Abbildung 02: Aufbau und Systemstruktur der Fuzzy-Regelung für die NH₄OH-Menge

Aus verfahrenstechnischer Sicht werden die wichtigsten Eingangsgrößen um den Prozess der Ammoniakwassereindüsung (NH₄OH-Mengeneindüsung) einbezogen. Dazu zählen der überschüssige Anteil an Ammoniakwasser (A_Schlupf), die Katalysatortemperatur (B_Tkat), die Konzentration der Stickoxide auf der Reingasseite nach Katalysator (C_NOXREIN), die Konzentration der Stickoxide auf der Rohgasseite vor Katalysator (NOXKAT) und das gesamte Rauchgasvolumen aller drei Müllkessel (RGVOL). Um die Anlage nach den Bedingungen der Feuerungen aller drei Kessel optimal zu fahren, werden die einzelnen Sekundärluftmengenströme (SEKU_1 bis SEKU_3) mit einbezogen. Die Sekundärluftmenge erhöht sich schwerpunktmäßig bei gesteigerter Dampfleistung (Feuerleistung), fallender Sauerstoffkonzentration im Rauchgas und ganz

entscheidend bei Bedingungen, die auf ein erhöhtes Emissionsrisiko von Kohlenmonoxid (CO) hindeuten, aber auch selbstverständlich bei hohen CO-Emissionen der einzelnen Feuerungen. Anhand der Sekundärluftfahrweise kann das gestiegene oder reduzierte Rauchgasvolumen vorzeitig über die Fuzzy-Feuerleistungsregelung der Fuzzy-NH₄OH-Mengen-Regelung mitgeteilt werden. Rauchgasschwankungen wirken sich besonders ungünstig auf die Temperatur vor Katalysator, den Schlupf (Überschuss von Ammoniakwasser im Rauchgas) und die Umsatzrate von Stickstoffoxiden zu molekularem Stickstoff aus.

3 Vergleich des Anlagenverhaltens ohne und mit Fuzzy

Dieses Kapitel beschreibt die Veränderungen, die durch die Fuzzy-Regelung eingetreten sind. Abbildung 03 (Tabelle 1) gibt einen Überblick über die gemessenen Größen, die in die Fuzzy-Regelung eingehen.

Gemessene Größe	Bezeichnung	Einheit
Stickoxide vor KAT	NO _x v. Kat	mg/m ³ (i.N.)
Stickoxide nach KAT	NO ₂ rein	mg/m ³ (i.N.)
Katalysator Temperatur	Kat-Temp.	°C
Rauchgasvolumenstrom	RG-Menge	m ³ /h (i.N.)
Ammoniakwasser	NH ₄ OH	l/min
Ammoniakschlupf (NH ₃)	Schlupf	mg/m ³ (i.N.)
Staub im Rauchgas	Staub	mg/m ³ (i.N.)
Sekundärluft Kessel 1	SEKU_1	m ³ (i.N.)/h
Sekundärluft Kessel 2	SEKU_2	m ³ (i.N.)/h
Sekundärluft Kessel 3	SEKU_3	m ³ (i.N.)/h

Abbildung 03 (Tabelle 1): Übersicht der gemessenen Größen in der Fuzzy-Regelung

Die Erläuterungen beziehen sich nur auf den 3-Linien-Betrieb der Anlage, da es sich hierbei um den weitaus kritischen Zustand handelt. Im 2- oder 1-Linien-Betrieb sind die Umsatzraten wesentlich temperaturunkritischer.

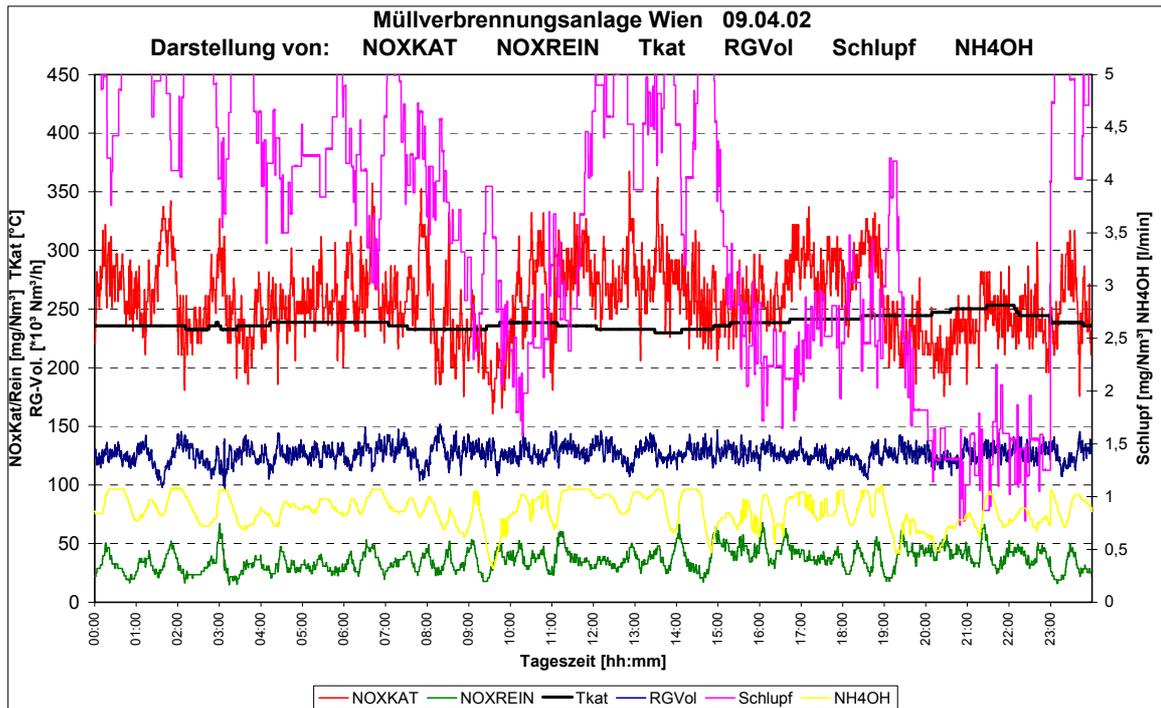


Abbildung 04: Auswertung ohne Fuzzy vom 09.04.2002 im 3-Linien-Betrieb

Das mehrmalige Überschreiten des Grenzwertes für den NH_3 -Schlupf im Reingas (5 mg/m^3 i.N., trocken) kann in Abbildung 04 beobachtet werden. Diese Überschreitungen treten immer kurz nach Absinken der Stickoxide im Reingas auf, da mit einer größeren Ammoniakwassermenge verbleibende Stickoxide im Rauchgas nicht mehr umgesetzt werden können.

Die Katalysatortemperatur erreicht zu keiner Zeit des Tages den angestrebten Wert von 260°C , lediglich in der Zeit zwischen 20.00 Uhr und 22.30 Uhr wird diese Temperatur annähernd erreicht. Die Abhängigkeit der Katalysatortemperatur zum NH_3 -Schlupf ist zu erkennen. Der Ammoniak Schlupf fällt von über 8 mg/m^3 (i.N., trocken) im angegebenen Zeitraum auf unter 2 mg/m^3 (i.N., trocken).

Ein ebenfalls erhöhten Ammoniak-Schlupf ist zwischen 10.00 Uhr und 17.00 Uhr in Abbildung 05 zu erkennen. In diesem Zeitraum hat die Katalysatortemperatur ein Minimum und liegt in diesem Zeitraum bei ca. 230°C . Der Katalysator erreicht, ähnlich wie am 09.04.2002, zu keinem Zeitpunkt seine Betriebstemperatur von 260°C . Auch hier ist die Abhängigkeit der Katalysatortemperatur zum NH_3 -Schlupf zu erkennen. Die unverbrauchte Ammoniakwassermenge kühlt den Katalysator an seiner Oberfläche ab. Normalerweise führt dies zu einer geringeren Umsetzung von Stickoxiden. Man müsste eigentlich eine Zunahme an Stickoxiden feststellen können, was an diesem Beispiel aber nicht eindeutig belegt werden kann.

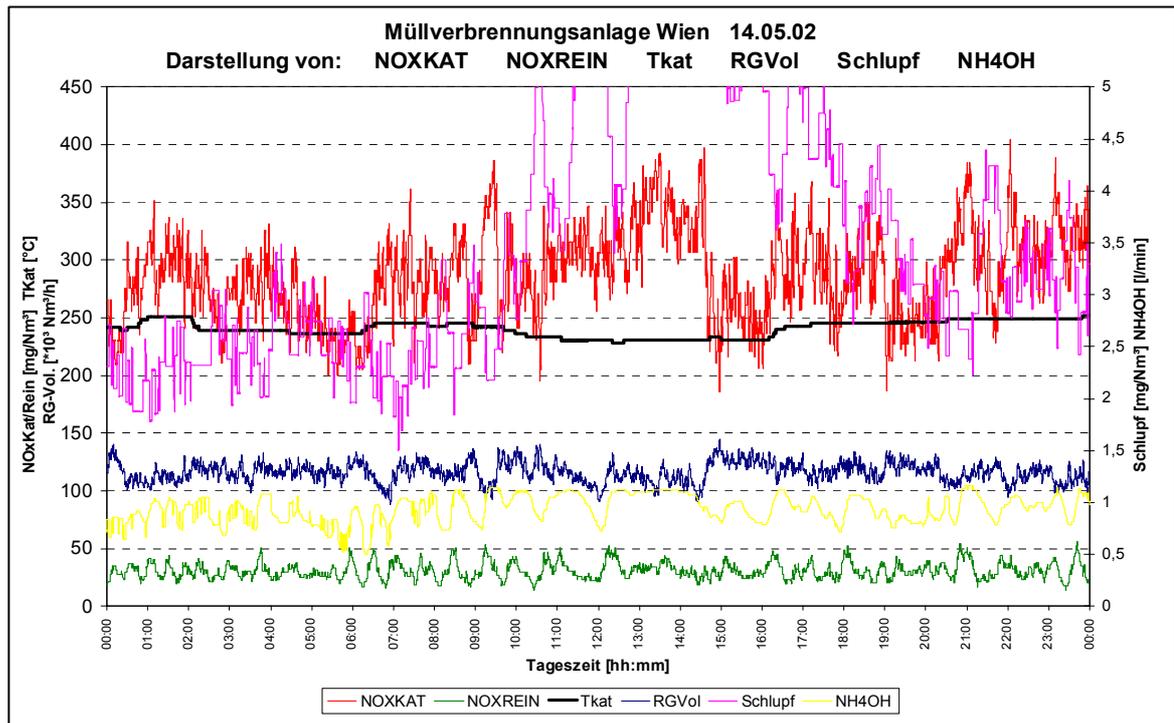


Abbildung 05: Auswertung ohne Fuzzy vom 14.05.2002 im 3-Linien-Betrieb

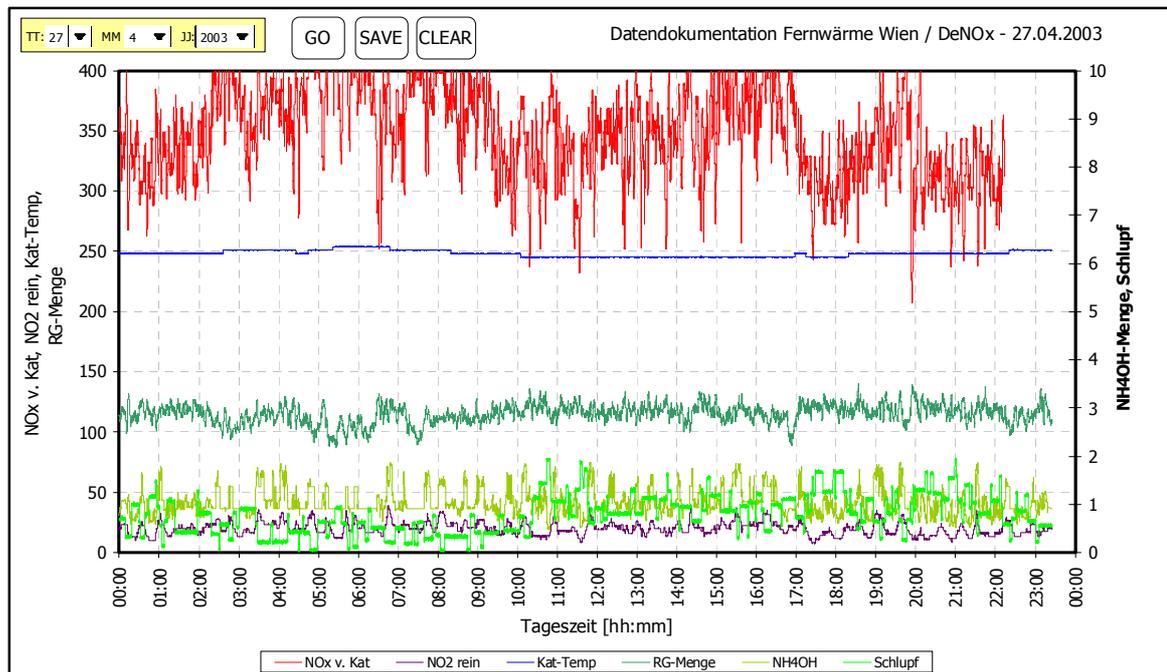


Abbildung 06: Auswertung mit Fuzzy vom 27.04.2003 im 3-Linien-Betrieb

Am Beispiel von Abbildung 06 können Veränderungen aufgezeigt werden, die mit Einführung der Fuzzy-Regelung in der MVA Flötzersteig korrelieren. Deutlich ist im Gesamtbild über den Tagesverlauf von 24 Stunden zu erkennen, dass der NH₃-Schlupf nicht mehr solch hohen Schwankungen unterliegt und sich im Mittel unter 2 mg/m³ i.N., trocken einpendelt.

Die Fuzzy-Feuerleistungsregelungen vermeiden in Summe zu hohe Gesamtvolumina an Rauchgas, was sich deutlich in den Abbildungen 06 aufzeigen lässt und die Temperatur von Katalysator auf 250°C stabilisiert.

Bei Abbildung 06 tritt mit Absenkung der Ammoniakwassermenge kein Überschreiten des Grenzwertes für NO_x im Reingas mehr auf. Die Trägheit des PID-Reglers (ohne Fuzzy-Regelung) führte dazu, dass bei Auftreten von NO_x-Spitzen nicht genügend bzw. zuviel Ammoniakwasser verspätet zudosiert wurde. Die Katalysatortemperatur erreicht am 27.04.2003 auch nicht die optimale Temperatur von 260 °C. Mit Einführung der Fuzzy-Regelung ist das Problem des NH₃-Schlupfes beseitigt, aber die Temperaturproblematik besteht im Volllastbetrieb weiterhin, da mit der Fuzzy-Feuerleistungsregelung und weiteren Veränderungen [3] die Dampfmengenleistung um 9% gesteigert werden konnte, somit auch die Rauchgasvolumina.

Abbildung 07 bestätigt diese Erkenntnisse. Der NH₃-Schlupf liegt ebenfalls unter 2 mg/m³ i.N., trocken. Stickoxide im Reingas liegen deutlich unterhalb von 30 mg/m³(i.N.).

Auch bei einer Katalysatortemperatur, die in diesem Beispiel permanent unter 250°C liegt, werden die Emissionswerte sicher auf niedrigem Niveau eingehalten. Das Temperaturproblem (bei maximaler Last aller Kessel und minimaler Gesamtrauchgasmenge) kann mit Einführung der Fuzzy-Regelung nicht mehr aus der Zudosierung der Ammoniakwassermenge stammen, sondern ermöglicht dem Betrieb aus verfahrenstechnischer Sicht eine Änderung der Strömungsverhältnisse und der Wärmeverschiebesystemtechnik zu analysieren.

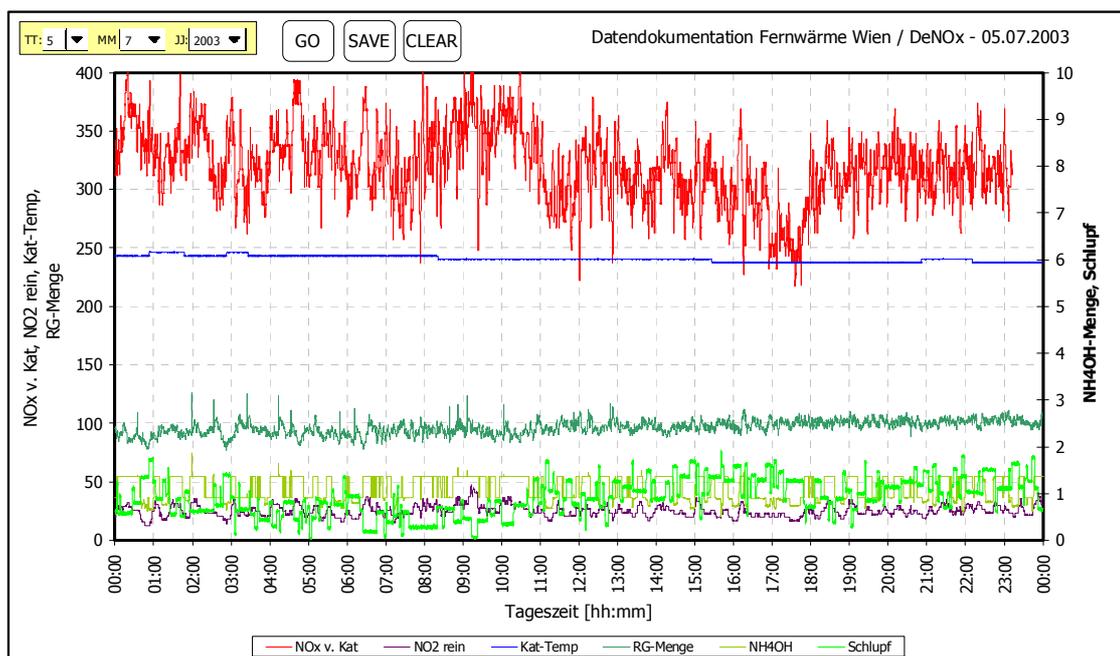


Abbildung 07: Auswertung mit Fuzzy vom 05.07.2003 im 3-Linien-Betrieb

4 Zusammenfassung

Die MVA Flötzersteig, Fernwärme Wien, war mit der bisherigen Regelung zur Eindüsung von Ammoniakwasser vor Katalysator unzufrieden. Die Fahrweise der Ammoniakwassereindüsung war aus verfahrenstechnischer Sicht mit den geänderten Bedingungen am Katalysator (mehr Rauchgas, über die Lebensdauer leicht reduzierte Reaktivität, niedrigere Temperaturen) unzureichend. Mit Einführung der Fuzzy - Feuerleistungsregelung an allen drei Kesseln der Anlage Flötzersteig wurde mit

gesteigerter Leistungsfähigkeit der Kessel neben der Frischdampferzeugung auch im Wesentlichen die Rauchgasmenge und Rauchgastemperatur angehoben.

Die notwendige Ammoniakwassermenge ist im Schnitt von 1,8 kg/Mg (Müll) im Jahresmittel auf bis zu 2,2 kg/Mg (Müll) angestiegen, wobei der Schlupf sehr schwankend zwischen 3 und 5 mg/m³ pendelte. Die sichere Einhaltung der Grenzwerte für das Reingas (Stickoxide und Schlupf) waren gefährdet. Die Einführung von Fuzzy Control hat sich seit dem Frühjahr 2002 in der Müllverbrennungsanlage Flötzersteig bestens bewährt. Die Emissionen an Stickoxiden (im Tagesmittel unter 30 mg/m³) und die verbleibende Menge an unverbrauchtem Ammoniak (Schlupf stabilisiert sich unter 1,5 mg/m³) sind auf ein Minimum gegenüber den Vorjahren zurückgegangen.

Die Feuerleistung aller drei Kessel konnte mit Fuzzy gesteigert werden, wobei die Rauchgasmenge anstieg. Damit kam die bisherige Regelung nicht zurecht. Die Fuzzy-Ammoniakwasser-Regelung kommt nicht nur mit der gestiegenen Gesamtrauchgasmenge zurecht, sondern kann auch im Jahresmittel die eingedüste Ammoniakwassermenge bei einem Wert von 3,5 kg/Mg (Müll) halten. Der Betrieb konnte nicht nur im 3-Linien-Betrieb optimiert werden, es kam ebenso im 2- als auch im 1-Linien-Betrieb zu den genannten Verbesserungen und zu einer wesentlichen Einsparung an Ammoniakwasser.

Besonders positiv hat sich ausgewirkt, dass die Fuzzy-Feuerleistungsregelung ebenfalls rückgekoppelt als Einzelfeuerleistungsregelung auf die maximale Rauchgasmenge und die Temperaturführung am Katalysator Rücksicht nehmen können. Die Einbindung der Sekundärluftmengen aller Kessel begünstigt vorzeitig die Berücksichtigung extremer Einzelercheinungen und gibt damit die Möglichkeit vorausschauend die Eindüsung von Ammoniakwasser vorzunehmen, so dass die Einsparungen an Ammoniakwasser eine nicht unerhebliche Menge am Jahresende ausmachen.

Gleichzeitig können die verbrauchten Gasmengen für die Aufheizung des Rauchgases vor Katalysator nachvollzogen zu einer verbesserten verfahrenstechnischen und/oder strömungstechnischen Überarbeitung der Anlagenkonfiguration vor SCR-Anlage genutzt werden. Das Gesamtzusammenspiel von wichtigen Größen der Feuerleistungsregelung und der Emissionen der Rauchgase vor und nach Katalysator bedingen den Erfolg der Fuzzy-Regelung.

Literaturverzeichnis

- [1] Gierend, Christian
Fuzzy Control in Kraftwerken: Potentiale und Anwendungsmöglichkeiten
VGB Fachtagung „Thermische Abfallbehandlung 1996“
VGB Kraftwerkstechnik 5/98, Heft 5, Essen 1998

- [2] Gierend, Christian
Optimierungspotential bei der Abfallverbrennung durch Fuzzy-Technologie,
Teil 1, VDI Wissensforum BAT- Fachtagung 2001, München

- [3] Locher, Ernst
Erfahrungen mit Fuzzy-Technologie am Beispiel MVA Flötzersteig, Wien, Teil 2
VDI Wissensforum BAT- Fachtagung 2001, München

The fuzzy component assignment problem

Thomas A. Runkler
Siemens AG Corporate Technology
Information and Communications, CT IC 4
81730 München
Tel. 089/636-45372
E-Mail Thomas.Runkler@siemens.com

Abstract

In the optimization of a real world logistic process we have to dynamically assign components from stock to waiting customer orders. This *component assignment problem* (CAP) is a relative of the the *bin packing problem*. For CAP, exhaustive search is infeasible because of its exponential complexity. So, the standard method is *largest order first* (LOF). In this paper, a fuzzified version of CAP is introduced. This *fuzzy component assignment problem* (FCAP) can be efficiently solved by linear programming (using the simplex algorithm). The fuzzy solutions perform at least as good as *any* crisp solution. However, since our real world process requires crisp solutions, we defuzzify the FCAP solutions using the *largest membership first* (LMF) algorithm. FCAP/LMF is about 3 times slower than LOF, but its solutions outperform the LOF solutions. FCAP/LMF solutions are (globally) optimal in 67.1% of the cases, and their average is at 97.34% of the global optimum.

1 Introduction: The Logistic Process

In this paper we consider a logistic (sub-)process at Fujitsu-Siemens-Computers (FSC) that provides computer systems to commercial customers (enterprise products). A schematic view of this logistic process is shown in Fig. 1.

Consider the example that a bank department shall be equipped with a new computer system consisting of several servers, terminals, storage devices, etc. The complete system shall be installed on a specific day, the so-called *desired delivery date*. The logistic process starts when the complete order, together with the desired delivery date, is submitted to FSC. FSC then collects all the arrived orders and requests the necessary components from selected suppliers at a specific time. In order to guarantee just-in-time delivery, the supplier order date is computed from the expected supplier delay and the customer's desired delivery date. The suppliers then deliver the requested components into FSC's stock, which is usually a *virtual stock*, e.g. a stock at an airport or a harbor.

In the ideal case, all components arrive as requested and each order can be delivered in time. In reality, however, the necessary components are often not available as requested. The reasons for this are manifold: the components might not be available at the suppliers, air planes might be delayed, deliveries might go to the wrong locations, the components might be damaged during transportation, or the suppliers might deliver the wrong component types. If the logistic system is disturbed in such a way, an intelligent reaction is required in order to maintain a maximum level of service to the customers, i.e. to minimize the delivery delays.

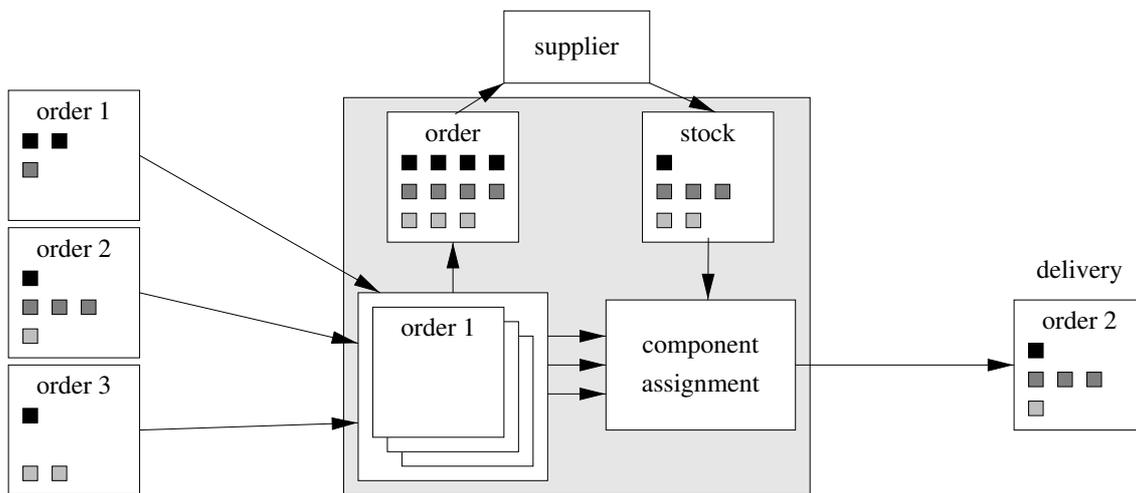


Figure 1: The logistic process at Fujitsu-Siemens-Computers.

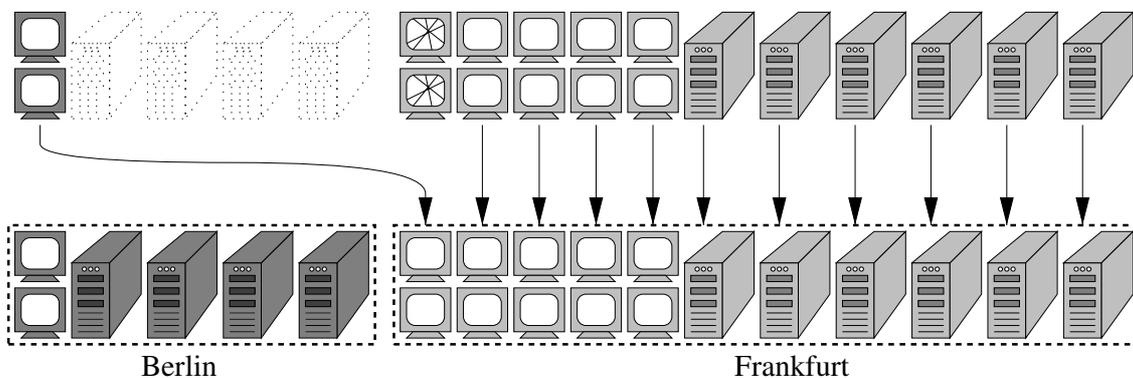


Figure 2: Optimization of the logistic process by exchanging components between orders.

In order to be able to react on disturbances, components between orders have to be exchanged. Consider the following example (Fig. 2): 2 monitors that had been ordered for a bank in Frankfurt are damaged during transportation. 2 monitors for a computer center in Berlin have already arrived, but this order is still waiting for 4 servers that have not arrived yet. So, the order for Frankfurt can be delivered, if 2 monitors for Berlin are sent to Frankfurt, and 2 new monitors are ordered for Berlin. Another option is to send four servers from Frankfurt to Berlin and deliver the order for Berlin instead. This option, however, would release only 6 stock items instead of 16.

Instead of just looking for the optimal way of exchanging components in case of disturbances, we want to *continuously* optimize the assignment of components to orders. This means that components are no longer explicitly requested for specific components, but that each component is (virtually) sent to a common stock and is then available to all the active orders. In this approach, each delivery decision is the result of an optimization problem: Maximize the number of components that can be delivered today. This *component assignment problem* is discussed in the following section.

2 The Component Assignment Problem

Formally, the *component assignment problem (CAP)* can be described as follows: We have a stock of components $c \in \mathbb{N}_0^p$, $p \in \mathbb{N}$, where each c_i , $i = 1, \dots, p$, denotes the number of available components of type i , e.g. the number of 19 inch TFT monitors in stock. We also have a stock of orders $X = \{x_1, \dots, x_n\} \subset \mathbb{N}_0^p$, $n \in \mathbb{N}$, where x_{ik} , the i^{th} component of the k^{th} order, $i = 1, \dots, p$, $k = 1, \dots, n$, denotes the number of components of type i required in order k . The goal in this component assignment problem is to assign the available components to the orders, so that as many components as possible can be delivered, i.e. to select a subset $Y = \{y_1, \dots, y_m\} \subseteq X$ with maximum

$$J(Y) = \sum_{i=1}^p \sum_{k=1}^m y_{ik}. \quad (1)$$

The constraint in this problem is the limited amount of components, i.e.

$$\sum_{k=1}^m y_{ik} \leq c_i \quad (2)$$

for all $i = 1, \dots, p$. An equivalent formulation of CAP is obtained by using a partition vector $u \in \{0, 1\}^n$, where $u_k = 1$ if and only if order k is delivered, $k = 1, \dots, n$. CAP can then be reformulated as the following problem:

Given $X = \{x_1, \dots, x_n\} \subset \mathbb{N}_0^p$, $n \in \mathbb{N}$, $p \in \mathbb{N}$, and $c \in \mathbb{N}_0^p$, find $u \in \{0, 1\}^n$, where

$$\sum_{k=1}^m u_k x_{ik} \leq c_i \quad (3)$$

for all $i = 1, \dots, p$, so that

$$J(u) = \sum_{i=1}^p \sum_{k=1}^m u_k x_{ik} \quad (4)$$

is maximized.

Notice that CAP is a discrete optimization problem. More specifically, it is a variation of a special case of the bin packing problem (BPP) [2]. In particular, the one-dimensional BPP is equivalent to CAP with only a single component type ($p = 1$). BPP tries to pack a subset of a set of items into a number of bins so that the total weight, volume, etc., does not exceed a maximum value. Fig. 3 shows an example for a two-dimensional single-bin packing problem. A subset of the 6 available items with the areas of 1, 14, 18, 25, 40, and 56 has to be placed onto a surface with the area 64. The bottom row of Fig. 3 shows three solutions for this problem which cover the areas of $1 + 18 + 40 = 59$, $1 + 56 = 57$, and $1 + 14 + 18 + 25 = 58$. We have solved this problem using exhaustive search, i.e. we have checked all possible combinations of items, and found that the solution 59 is optimal.

Fig. 4 shows a *one-dimensional* single-bin example of the bin packing problem. A subset of the 6 available bars with the lengths 1, 14, 18, 25, 40, and 56 has to be attached to each other so that the maximum length of 64 is not exceeded. The three bottom views show three solutions to this problem with the lengths $1 + 18 + 40 = 59$, $1 + 56 = 57$, and $1 + 14 + 18 + 25 = 58$. Again, we have performed exhaustive search and found that the solution 59 is optimal.

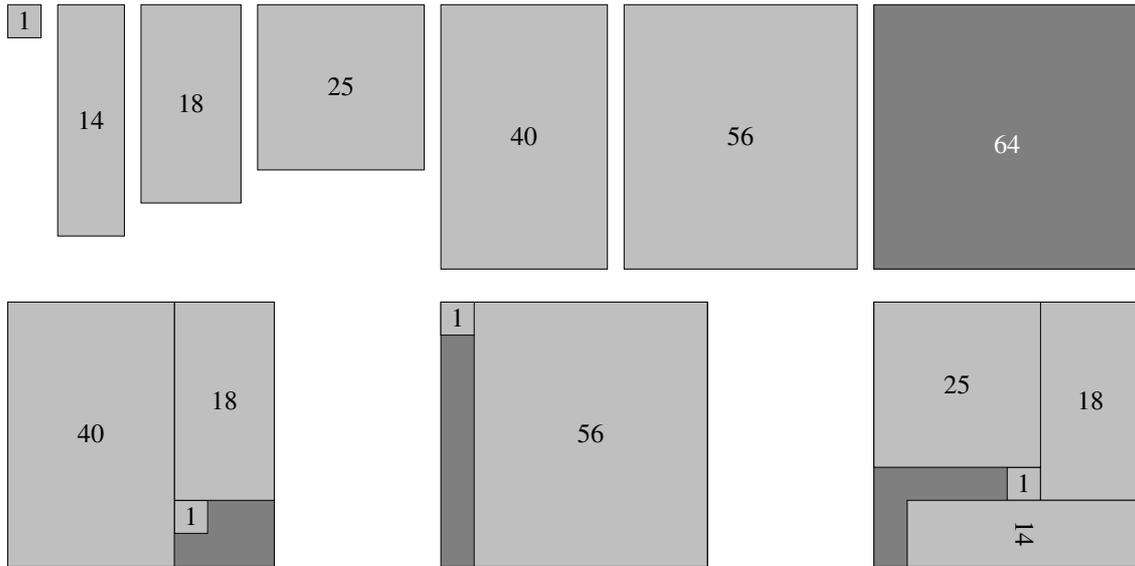


Figure 3: Two-dimensional single-bin example of the bin packing problem. Top row: 6 available items (light grey) with the areas 1, 14, 18, 25, 40, and 56 have to be placed onto a surface with the area 64 (dark grey). Bottom row: 3 solutions; the first solution is optimal and covers the area $1 + 18 + 40 = 59$.

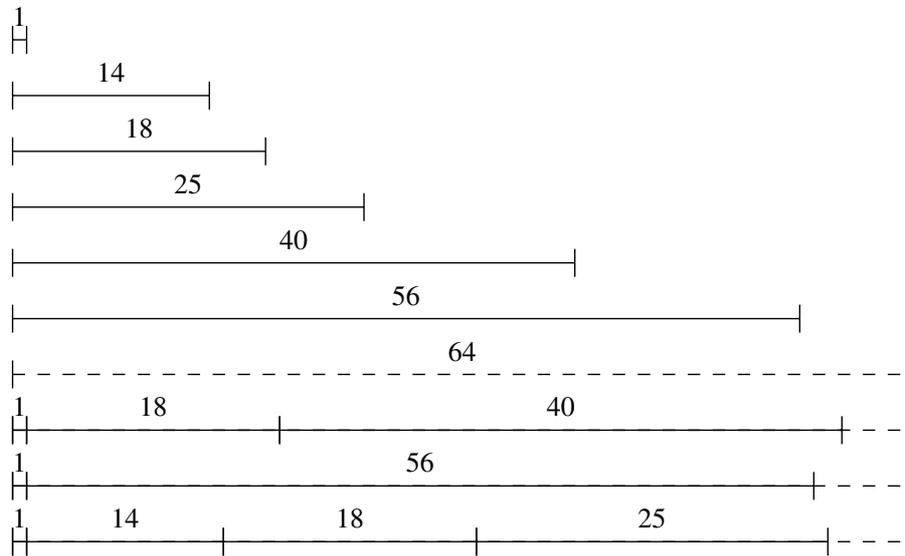


Figure 4: One-dimensional single-bin example of the bin packing problem: from top to bottom: 6 available items with the lengths 1, 14, 18, 25, 40, and 56, the maximum overall length of 64 that the items have to fit in (dashed), and three solutions; the first solution is optimal and extends to the length $1 + 18 + 40 = 59$.

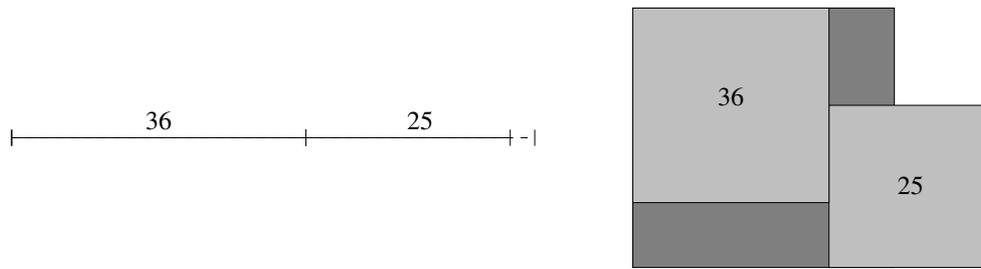


Figure 5: Two items with lengths/areas 36 and 25. In the one-dimensional case (left) they fit into the overall length of 64, In the two-dimensional case (right) they do not fit into the overall area of 64.

As mentioned before, we can equivalently formulate this one-dimensional single-bin BPP as a CAP using

$$c = 64 \quad (5)$$

$$X = \{1, 14, 18, 25, 40, 56\}. \quad (6)$$

This means that we have only *one* type of components, and we have 64 of these on stock; moreover, we have 6 orders waiting: order 1 needs 1 component, order 2 needs 14 components, order 2 needs 18 components, and so on. In this example, the CAP formulation then reads:

Find $u \in \{0, 1\}^6$, where

$$u_1 + 14 \cdot u_2 + 18 \cdot u_3 + 25 \cdot u_4 + 40 \cdot u_5 + 56 \cdot u_6 \leq 64 \quad (7)$$

so that

$$J(u) = u_1 + 14 \cdot u_2 + 18 \cdot u_3 + 25 \cdot u_4 + 40 \cdot u_5 + 56 \cdot u_6 \quad (8)$$

is maximized.

Notice that all the numerical values in the previous two examples at Figs. 3 and 4 are equal. In the two-dimensional example the numbers refer to areas, whereas in the one-dimensional example they refer to lengths. However, in the two-dimensional example the shapes of the items play an important role. For example, consider the two bars with the lengths 25 and 36 shown in the left view of Fig. 5. They cover the overall length of $25 + 36 = 61$, and therefore fit into the limit of 64. However, the squares with the areas 25 and 36 shown in the right view of Fig. 5 do not fit into the limit box with the area of 64. So, multi-dimensional BPP examples are not necessarily equivalent to their one-dimensional correspondents.

The best known algorithm for BPP sorts the orders by size and then successively delivers them, starting with the largest orders, as long as sufficient components are available. We

will call this algorithm *largest order first (LOF)* here. In [3] it was shown that LOF (and any other efficient algorithm) can not be guaranteed to be less suboptimal than 22%. The second solutions shown in Figs. 3 and 4 are the solutions that were obtained using LOF. First, LOF places the largest item (56). Then it places the largest remaining item that fits (1), and achieves the solution 57, which is $1 - 57/59 \approx 3.39\%$ suboptimal.

Recently, several approaches were published to solve CAP using ant algorithms [7], evolutionary algorithms [8], market-based algorithms [6], and fuzzy decision making [9]. This paper proposes a fuzzification of CAP that leads to a very efficient optimization algorithm.

3 The Fuzzy Component Assignment Problem

The *fuzzy component assignment problem (FCAP)* is obtained by fuzzifying CAP in its partitional form. FCAP is then defined as the following problem:

Given $X = \{x_1, \dots, x_n\} \subset \mathbb{N}_0^p$, $n \in \mathbb{N}$, $p \in \mathbb{N}$, and $c \in \mathbb{N}_0^p$, find $u \in [0, 1]^n$, where

$$\sum_{k=1}^m u_k x_{ik} \leq c_i \quad (9)$$

for all $i = 1, \dots, p$, so that

$$J(u) = \sum_{i=1}^p \sum_{k=1}^m u_k x_{ik} \quad (10)$$

is maximized.

Formally, the only difference between CAP and FCAP is the domain of u . The fuzzification of u , however, transforms the discrete (and therefore difficult) problem into a continuous (and therefore simpler) problem. Hence, FCAP (in contrast to CAP) can be solved efficiently by linear programming using the simplex algorithm [1].

The constraints (9) in FCAP form multiple joint simplexes in \mathbb{R}^p . A simplex in \mathbb{R}^p is a figure with $p + 1$ vertices. For $p = 2$ it is a triangle, for $p = 3$ it is a pyramid. In the linear case (as in FCAP), the optimal solutions of the objective function J (10) are on the vertices of the simplexes. The simplex algorithms finds the optimum by consecutively moving from vertex to vertex, where the objective function is increased in each move. These movements are repeated, until a (local) optimum is found. Fig. 6 illustrates the simplex algorithm for the following numerical example:

$$c = (2, 1, 1) \quad (11)$$

$$X = \{(2, 0, 1), (0, 4, 1)\}. \quad (12)$$

This means that we have 3 different types of components; on stock we have 2 components of type 1, 1 component of type 2, and 1 component of type 3; we have 2 orders waiting: order 1 needs 2 components of type 1 and 1 component of type 3, and order 2 needs 4 components of type 2 and 1 component of type 3. The corresponding optimization problem then reads:

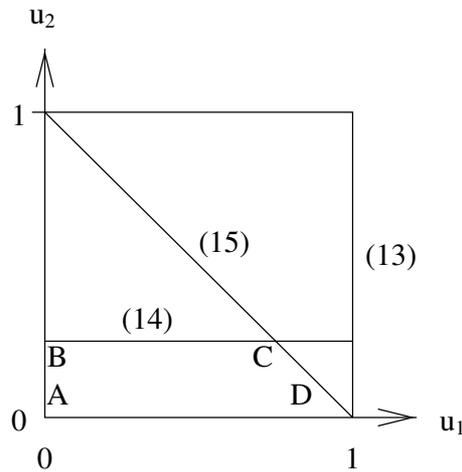


Figure 6: Example for the optimization of FCAP using the simplex algorithm.

Find $u \in [0, 1]^2$, where

$$2 \cdot u_1 \leq 2, \quad (13)$$

$$4 \cdot u_2 \leq 1, \quad (14)$$

$$u_1 + u_2 \leq 1 \quad (15)$$

so that

$$J(u) = 3 \cdot u_1 + 5 \cdot u_2 \quad (16)$$

is maximized.

The constraints $u \in [0, 1]^2$ and (13)...(15) form the quadrangle through the points $A \dots D$ shown in Fig. 6. Since J (16) is linear, the optimum of J under these constraints must be in one of the vertices $A \dots D$. The simplex algorithm iteratively finds the optimum vertex without necessarily checking all the vertices. However, in this simple example, we can just compute all the $J(A) \dots J(D)$,

$$J(A) = J((0, 0)) = 3 \cdot 0 + 5 \cdot 0 = 0, \quad (17)$$

$$J(B) = J((0, \frac{1}{4})) = 3 \cdot 0 + 5 \cdot \frac{1}{4} = \frac{5}{4}, \quad (18)$$

$$J(C) = J((\frac{3}{4}, \frac{1}{4})) = 3 \cdot \frac{3}{4} + 5 \cdot \frac{1}{4} = \frac{7}{2}, \quad (19)$$

$$J(D) = J((1, 0)) = 3 \cdot 1 + 5 \cdot 0 = 3, \quad (20)$$

and see that $J(C) = \frac{7}{2}$ is optimal. Thus, the optimal solution of FCAP is $u = (\frac{3}{4}, \frac{1}{4})$, i.e. order 1 should be delivered with membership $u_1 = \frac{3}{4}$, and order 2 should be delivered with membership $u_2 = \frac{1}{4}$. Notice that in *CAP* we have $u \in \{0, 1\}^n$, so in the corresponding *CAP* example only the vertices A and D are feasible, and thus $J(D) = 3$ with $u = (1, 0)$ is the optimal solution. So, in this example FCAP yields the optimum $J = \frac{7}{2}$, while *CAP* only yields the optimum $J = 3$. However, in our logistic problem we do not want to deliver order 1 by 75% and order 2 by 25%, but we only want to deliver orders completely, i.e. we are only interested in *crisp* solutions $u \in \{0, 1\}^n$. In order to obtain crisp solutions using FCAP, we can apply (subset) defuzzification [4] to the fuzzy partition u that is produced by FCAP. In this case, we use FCAP as an *approximation* to *CAP*.

This subset defuzzification can be done using the alpha cut u^α ,

$$u_k^\alpha = \begin{cases} 1 & \text{if } u_k \geq \alpha \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

$k = 1, \dots, n$, where the cut level α is chosen so that the constraints

$$\sum_{k=1}^m u_k^\alpha x_{ik} \leq c_i, \quad (22)$$

$i = 1, \dots, p$, are satisfied. This approach has the following drawback: If say order m , $m \in \{1, \dots, n-1\}$, has to be rejected ($u_m^\alpha = 0$) because it is too large to satisfy (22), then all the orders with $u_k < u_m$ will not be considered any more, although they might still fit into (22). This drawback is avoided in the following algorithm that is similar to the LOF scheme and this therefore called *largest membership first* (LMF):

1. initialize $s = \underbrace{(0, \dots, 0)}_p$ and $u^* = \underbrace{(0, \dots, 0)}_n$
2. sort orders, so that $u_1 \geq u_2 \geq \dots \geq u_n$
3. for $k = 1, \dots, n$
 - if $s + x_k \leq c$
 - $u_k^* = 1$
 - $s = s + x_k$
 - end if
- end for

Notice that LMF is very similar to the *equal size subset defuzzification* (ESSD) introduced in [5].

The third solutions presented in Figs. 3 and 4 have been obtained by solving the FCAP using the simplex algorithm and subsequently defuzzifying the results by LMF. The results for Figs. 3 and 4 can now be summarized as follows:

$$\text{exhaustive search:} \quad J(1, 0, 1, 0, 1, 0) = 59, \quad (23)$$

$$\text{LOF:} \quad J(1, 0, 0, 0, 0, 1) = 57, \quad (24)$$

$$\text{FCAP:} \quad J(1, 1, 1, 1, 0.15, 0) = 64, \quad (25)$$

$$\text{FCAP/LMF:} \quad J(1, 1, 1, 1, 0, 0) = 58. \quad (26)$$

The best results were obtained by exhaustive search and by FCAP. In this case, FCAP even yields the maximum possible score of 64. Even after LMF defuzzification FCAP yields a better result than LOF. Exhaustive search has the highest computational effort. In this example, $2^6 = 64$ possible solutions have to be checked. LOF checks $6 + 5 + 4 = 15$ solutions, and FCAP checks only 6 solutions. The examples from Figs. 3 and 4 were very simple, because they only involve one type of components ($p = 1$). In the following section we present some more experiments with more complex examples.

4 Experiments

In this section we consider examples with $p = 8$ types of components and $n = 6$ waiting orders. The components in stock and the component requirements in the orders are generated randomly. The component numbers in stock are random numbers in $\{0, 1, \dots, 9\}$, and the component numbers in the orders are random numbers in $\{0, 1, 2\}$. In all these examples, the exhaustive search checks all possible $2^6 = 64$ combinations. We do not present any experiments with more than 6 orders here, because of the combinatorial explosion in the run time of the exhaustive search. Remember that exhaustive search checks 2^n combinations, so it gets computationally infeasible for a higher number of orders. For example, for $n = 266$ orders we have $2^{266} \approx 10^{80}$ combinations, which is the estimate for the number of atoms in the universe.

Among the $10^8 \cdot 3^{(8 \cdot 6)} \approx 8 \cdot 10^{30}$ possible constellations for this experiment we picked five specific cases that illustrate the properties of the considered optimization algorithms:

1. $J_{\text{exhaustive}} = J_{\text{LOF}} = J_{\text{FCAP/LMF}}$
2. $J_{\text{exhaustive}} = J_{\text{FCAP/LMF}} > J_{\text{LOF}}$
3. $J_{\text{exhaustive}} = J_{\text{LOF}} > J_{\text{FCAP/LMF}}$
4. $J_{\text{exhaustive}} > J_{\text{FCAP/LMF}} > J_{\text{LOF}}$
5. $J_{\text{exhaustive}} > J_{\text{LOF}} > J_{\text{FCAP/LMF}}$

To find examples for these cases, we just randomly picked c and X , ran all four algorithms, and then checked if the results matched one of five cases defined above. For 1000 of these random picks we obtained the following statistics for the five cases:

case 1:	570 times = 57%
case 2:	101 times = 10.1%
case 3:	92 times = 9.2%
case 4:	13 times = 1.3%
case 5:	7 times = 0.7%

Notice that these percentages do not sum up to 100%, since the case $J_{\text{exhaustive}} > J_{\text{LOF}} = J_{\text{FCAP/LMF}}$ is not considered here. Following these statistics, for this specific CAP parametrization, LOF as well as FCAP/LMF very often finds the global optimum. In particular, LOF finds the global optimum 662 times (66.2%), and FCAP/LMF finds the global optimum 671 times (67.1%).

The left view of Fig. 7 shows the relative values of the objective function for FCAP/LMF (vertical) versus LOF (horizontal). In this view, a value of 1 represents the global optimum, so the ticks at the top and on the right hand side refer to the examples where FCAP/LMF and LOF find the global optima, respectively. There are also many ticks on the main diagonal referring to examples with equal but suboptimal solutions of both LOF and FCAP/LMF. Observe that the upper triangle contains more ticks than the lower triangle, i.e. FCAP/LMF more often outperforms LOF than vice versa. The average relative

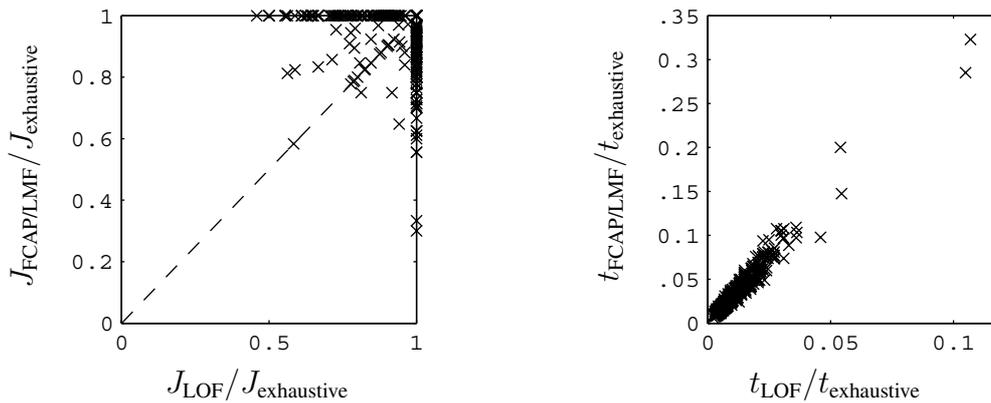


Figure 7: FCAP/LMF (vertical) versus LOF (horizontal). Left view: relative values of the objective function; a value of 1 corresponds with the global optimum. Right view: relative computation times; a value of 1 corresponds with the time for exhaustive search.

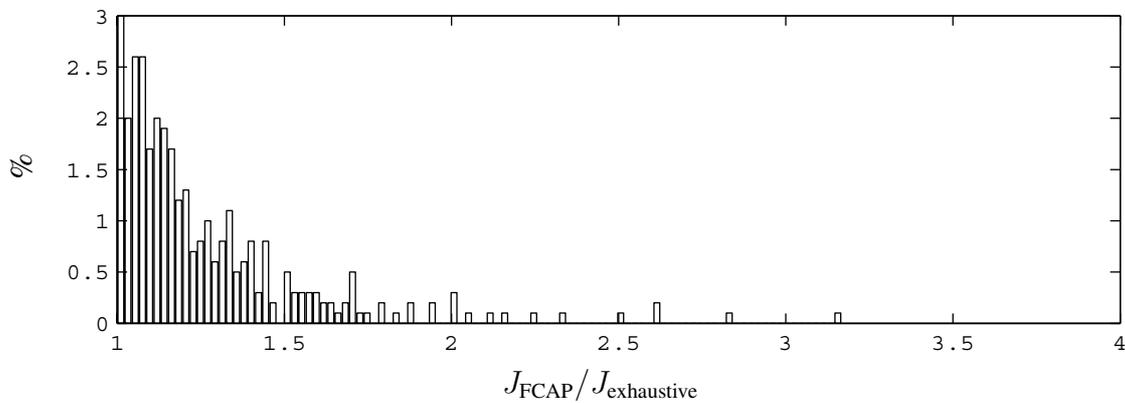


Figure 8: Histogram of the relative values of the objective function for FCAP. The first bar (at 1) is cut off. The actual height of this first bar is 495. A value of 1 corresponds with the global optimum, so all FCAP solutions are at least equal to the global optimum.

performance of LOF is 0.9633, and the average relative performance of FCAP/LMF is 0.9734.

The actual fuzzy FCAP yields a relative performance of 1 (i.e. the global optimum) in 495 cases (49.5%). In all other cases, FCAP performs even better! Fig. 8 shows the histogram of these other cases (for a clearer view, the first bar of height 495 is cut off at the value of 30). In our sample, the best result was $\frac{19}{6} \approx 3.1666$ times the global optimum.

We also compared the computation times of the different optimization methods. In these examples, the exhaustive search takes about 100 milliseconds on a 3 GHz Pentium 4 CPU. This might appear fast, but for $n = 20$, for example, exhaustive search takes about 2 minutes, and for $n = 25$, it takes about 2 hours, so there is a clear need for more efficient methods. The right view of Fig. 7 shows the scatter plot of the relative computation times of LOF and FCAP/LMF. A value of 1 corresponds here with the time for exhaustive search. So, LOF usually needs about 2% of the time for exhaustive search, and FCAP/LMF needs about 5% of this time. The right view of Fig. 7 shows a linear depen-

gency between both computation times. The slope is about 3, so FCAP/LMF is about 3 times slower than LOF, but about $\frac{1}{0.05} = 20$ times faster than exhaustive search.

In the following we will discuss one example for each of the five cases introduced above in more detail.

In the first example, we use the stock and order component vectors

$$\begin{aligned}
 c &= (2, 4, 4, 4, 3, 6, 7, 3) & (27) \\
 X &= \{(0, 1, 1, 2, 2, 1, 2, 0), \\
 &\quad (0, 0, 0, 2, 0, 2, 0, 0), \\
 &\quad (1, 2, 0, 0, 1, 1, 2, 0), \\
 &\quad (0, 0, 0, 2, 0, 0, 1, 0), \\
 &\quad (1, 2, 0, 2, 2, 2, 2, 2), \\
 &\quad (2, 0, 2, 1, 0, 1, 2, 1)\}. & (28)
 \end{aligned}$$

For this example, all four optimization methods find the optimum solution

$$\text{exhaustive search:} \quad J(0, 1, 1, 0, 1, 0) = 24, \quad (29)$$

$$\text{LOF:} \quad J(0, 1, 1, 0, 1, 0) = 24, \quad (30)$$

$$\text{FCAP:} \quad J(0, 1, 1, 0, 1, 0) = 24, \quad (31)$$

$$\text{FCAP/LMF:} \quad J(0, 1, 1, 0, 1, 0) = 24. \quad (32)$$

As mentioned above, such a case occurs very often. Notice that FCAP yields a crisp solution here, that leads to FCAP=FCAP/LMF. Hence, crisp FCAP solutions seem to be indicators for globally optimal solutions.

In the second example, we use

$$\begin{aligned}
 c &= (3, 8, 8, 5, 4, 8, 8, 6) & (33) \\
 X &= \{(2, 2, 1, 1, 2, 0, 1, 1), \\
 &\quad (1, 0, 1, 2, 2, 1, 1, 2), \\
 &\quad (1, 2, 2, 0, 0, 0, 1, 1), \\
 &\quad (0, 1, 1, 2, 0, 1, 1, 1), \\
 &\quad (1, 1, 2, 0, 2, 0, 1, 0), \\
 &\quad (1, 2, 2, 0, 0, 2, 0, 1)\}. & (34)
 \end{aligned}$$

Here, both FCAP and FCAP/LMF find the (global) optimum, but LOF fails:

$$\text{exhaustive search:} \quad J(0, 1, 0, 1, 1, 1) = 32, \quad (35)$$

$$\text{LOF:} \quad J(1, 1, 0, 1, 0, 0) = 27, \quad (36)$$

$$\text{FCAP:} \quad J(0, 1, 1, 1, 0, 1) = 32, \quad (37)$$

$$\text{FCAP/LMF:} \quad J(0, 1, 1, 1, 0, 1) = 32. \quad (38)$$

LOF is blocked here after picking order number 1. Notice that FCAP again produces a crisp, and again optimal result. Also notice that FCAP and FCAP/LMF yield a different solution than exhaustive search. So, obviously, this problem has multiple global optima.

In the third example, we use

$$c = (5, 4, 4, 6, 4, 4, 8, 1) \quad (39)$$

$$X = \{(2, 0, 0, 2, 2, 0, 1, 0), \\ (1, 2, 0, 0, 1, 2, 2, 1), \\ (1, 1, 1, 0, 0, 1, 0, 2), \\ (2, 0, 1, 1, 0, 2, 0, 2), \\ (0, 1, 0, 2, 1, 1, 2, 0), \\ (2, 2, 1, 0, 0, 1, 1, 0)\}. \quad (40)$$

Here, LOF finds the (global) optimum, but FCAP/LMF fails:

$$\text{exhaustive search:} \quad J(1, 1, 0, 0, 0, 1) = 23, \quad (41)$$

$$\text{LOF:} \quad J(1, 1, 0, 0, 1, 0) = 23, \quad (42)$$

$$\text{FCAP:} \quad J(1, \frac{1}{2}, 0, \frac{1}{4}, 1, 1) = 27\frac{1}{2}, \quad (43)$$

$$\text{FCAP/LMF:} \quad J(1, 0, 0, 0, 1, 1) = 21. \quad (44)$$

Notice that FCAP yields a fuzzy solution, so here a fuzzy FCAP solution coincides with a suboptimal FCAP/LMF solution. However, the actual fuzzy FCAP solution yields a higher J than the crisp global optimum of 23. Also notice that FCAP/LMF gets suboptimal, because it excludes the second order. However, the second order at least gets a considerable membership of $u_2 = \frac{1}{2}$ in FCAP. Also notice that LOF yields a different optimum than exhaustive search, so we have multiple optima here again.

In the fourth example, we use

$$c = (9, 9, 1, 9, 8, 2, 1, 8) \quad (45)$$

$$X = \{(0, 1, 0, 0, 0, 1, 1, 2), \\ (2, 1, 0, 0, 1, 0, 2, 0), \\ (1, 0, 0, 1, 1, 1, 1, 2), \\ (1, 1, 1, 0, 2, 1, 0, 2), \\ (0, 1, 1, 2, 0, 1, 1, 0), \\ (2, 2, 0, 0, 2, 2, 0, 1)\}. \quad (46)$$

Here FCAP/LMF performs better than LOF, but both are suboptimal:

$$\text{exhaustive search:} \quad J(0, 0, 1, 1, 0, 0) = 15, \quad (47)$$

$$\text{LOF:} \quad J(0, 0, 0, 0, 0, 1) = 9, \quad (48)$$

$$\text{FCAP:} \quad J(0, \frac{1}{2}, 0, 1, 0, \frac{1}{2}) = 15\frac{1}{2}, \quad (49)$$

$$\text{FCAP/LMF:} \quad J(1, 0, 0, 1, 0, 0) = 13. \quad (50)$$

LOF picks order number 6, the largest order, and is then blocked. FCAP yields a fuzzy solution again that corresponds to a suboptimal FCAP/LMF solution. However, the (fuzzy) FCAP solution still outperforms the crisp global optimum. Notice that both the second and the sixth orders have memberships $u_2 = u_6 = \frac{1}{2}$ in FCAP, but get zero memberships in FCAP/LMF. This is because after picking order number 4 the available components are

not sufficient any more to satisfy the second or the sixth order. Instead, order number 1 obtains membership one in FCAP/LMF, although $u_1 = 0$ in FCAP. Including order number 1 in FCAP/LMF improves the result J . This example shows that the LMF defuzzification is superior to a simple alpha cut (22) which would have excluded order number 1.

In the fifth and final example, we use

$$c = (5, 2, 5, 1, 2, 2, 7, 9) \quad (51)$$

$$X = \{(0, 2, 1, 1, 1, 2, 1, 2), \\ (1, 1, 2, 1, 0, 1, 2, 1), \\ (2, 2, 0, 1, 1, 1, 1, 0), \\ (2, 2, 1, 0, 0, 0, 1, 2), \\ (1, 0, 0, 0, 2, 2, 0, 2), \\ (0, 2, 0, 1, 1, 2, 0, 0)\}. \quad (52)$$

Here LOF performs better than FCAP/LMF, but both are suboptimal:

$$\text{exhaustive search:} \quad J(0, 0, 0, 1, 1, 0) = 15, \quad (53)$$

$$\text{LOF:} \quad J(1, 0, 0, 0, 0, 0) = 10, \quad (54)$$

$$\text{FCAP:} \quad J(0, 1, 0, \frac{1}{2}, \frac{1}{2}, 0) = 16\frac{1}{2}, \quad (55)$$

$$\text{FCAP/LMF:} \quad J(0, 1, 0, 0, 0, 0) = 9. \quad (56)$$

LOF picks order number 1, the largest order, and is then blocked. FCAP yields a fuzzy solution again that corresponds to a suboptimal FCAP/LMF solution. And again, the (fuzzy) FCAP solution outperforms the crisp global optimum. Orders number 4 and 5 form a (crisp) global optimum. These orders have the membership $u_4 = u_5 = \frac{1}{2}$ in FCAP, but are not included in FCAP/LMF, because order number 2 has membership 1 in FCAP and then blocks all the other orders.

5 Conclusions

To solve the component assignment problem (CAP), four different approaches were investigated: exhaustive search, largest order first (LOF), a simplex algorithm to solve the fuzzy component assignment problem (FCAP), and a largest membership first (LMF) defuzzification of the FCAP solutions.

In terms of the solutions found, FCAP outperforms all three other algorithms. The fuzzy solutions found by FCAP are at least as good as the crisp global optima, and the best result found in the experiments exceeded the global optimum by more than 200%. However, our logistic application example requires crisp solutions, so a defuzzification of the results is necessary. The defuzzified FCAP/LMF solutions outperform the LOF solutions. FCAP/LMF found the global optimum in 67.1% of the cases and yielded an average of 97.34% of the global optimum; LOF found the global optimum in only 66.2% of the cases and only yielded an average of 96.33% of the global optimum.

In terms of computational efficiency, the worst method is exhaustive search because of its exponential complexity. LOF and FCAP/LMF both have only quadratic complexity, where FCAP/LMF is about 3 times slower than LOF.

Even though FCAP/LMF is a little slower than LOF, it is still a computationally very efficient method to solve CAP. Its solutions statistically outperform the LOF solutions, and they are mostly very close to the global optima. Therefore, FCAP/LMF is a very attractive method for the component assignment problem.

References

- [1] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, 1963.
- [2] M. R. Garey, R. L. Graham, and J. D. Ullman. An analysis of some packing algorithms. In *Combinatorial Algorithms*, pages 39–47. Algorithmics Press, New York, 1973.
- [3] P. Hoffman. *The Man Who Loved Only Numbers: The Story of Paul Erdos and the Search for Mathematical Truth*. Hyperion, New York, 1998.
- [4] T. A. Runkler. Selection of appropriate defuzzification methods using application specific properties. *IEEE Transactions on Fuzzy Systems*, 5(1):72–79, 1997.
- [5] T. A. Runkler and J. C. Bezdek. RACE: Relational alternating cluster estimation and the wedding table problem. In W. Brauer, editor, *Fuzzy–Neuro–Systems '98, München*, volume 7 of *Proceedings in Artificial Intelligence*, pages 330–337, March 1998.
- [6] C. A. Silva, R. Palm, T. A. Runkler, and J. M. Sousa. Market-based negotiation of delivery dates in a logistic process. In *European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (eunite), Oulu, Finland*, pages 69–94, July 2003.
- [7] C. A. Silva, T. A. Runkler, and J. M. Sousa. Evolved ant colonies optimization in the scheduling of logistic processes. In *GMA/GI Workshop Fuzzy Systems, Dortmund*, pages 145–158, November 2002.
- [8] C. A. Silva, J. M. Sousa, T. A. Runkler, and J. M. Sá da Costa. Evolved genetic algorithms with fuzzy aggregation applied to priorities in logistic systems. In *IEEE International Conference on Emerging Technologies and Factory Automation, Lisbon, Portugal*, pages 775–781, September 2003.
- [9] J. M. Sousa, R. H. Palm, C. A. Silva, and T. A. Runkler. Optimizing logistic processes using a fuzzy decision making approach. *IEEE Transactions on Systems, Man, and Cybernetics A*, 33(2):245–256, 2003.

Ein genetischer Algorithmus zur Konturerkennung sternförmiger Gebiete

Jörg Krone¹⁾, Christian Douven¹⁾, Heinrich Brinck²⁾, Ulrich Lehmann¹⁾

- 1) Fachhochschule Südwestfalen, Frauenstuhlweg 31, 58644 Iserlohn, Tel.: (02371) 566-140, Fax: (02371) 566-274, E-Mail: krone@fh-swf.de
- 2) Fachhochschule Gelsenkirchen, August-Schmidt-R. 10, 45657 Recklinghausen, Tel.: (02361) 915-445, Fax: (02361) 915-499, E-Mail: brinck@fh-ge.de

1. Einleitung

Die Konturerkennung ist ein klassisches Problem der digitalen Bildverarbeitung. Falls die untersuchten Grauwertbilder stark verrauscht und kontrastarm sind, führen die elementaren Kantendetektoren, z. B. Sobel- oder Laplace-Operator, dazu, dass die Kontur eines Gebietes unvollständig oder falsch erkannt wird. Für dieses Problem wird hier ein genetischer Algorithmus vorgestellt. Er arbeitet von vornherein mit geschlossenen Konturen, die iterativ den abgebildeten Objekten angenähert werden.

Der vorgestellte Algorithmus dient der Konturerkennung zweidimensionaler, beschränkter Gebiete, die sternförmig sein müssen. Ein Gebiet G heißt sternförmig, wenn es einen Punkt $P \in G$ gibt, so dass für alle Punkte $Q \in G$ die Verbindungslinien $V(P,Q)$ ganz in G liegen. So sind konvexe Gebiete immer sternförmig und jeder Punkt P eines konvexen Gebietes erfüllt obige Bedingung.

Es wird eine BLOB-Analyse und eine klassische Bildsegmentierung zur Vorverarbeitung verwendet, um die ungefähre Lage der Gebiete zu ermitteln. Auf diesen Informationen baut dann der genetische Algorithmus auf. Als Individuen oder Chromosomen wurden B-Splines gewählt, weil sie schnell berechnet werden können und weil sie in der konvexen Hülle ihrer Kontrollpunkte liegen. Die Tatsache, dass B-Splines die Kontrollpunkte lediglich approximieren, ist für die Anwendung in der Konturerkennung unproblematisch, da die Lage der Kontrollpunkte aus der Vorverarbeitung ermittelt und nicht sicher bestimmt ist. Die Fitnessfunktion basiert auf der durchschnittlichen Grauwertänderung entlang der mittels B-Splines repräsentierten Kante und wird mit dem Sobel-Operator berechnet. Die übrigen genetischen Operatoren wurden zu Testzwecken variabel implementiert. So kann das Auswahlverfahren (Monte Carlo, Tournament, Fitness-Proportional, Linear Ranking) und das Ersetzungsschema (Strong Elitism, Weak Elitism, Complete Replacement) frei konfiguriert werden. Auch Populationsgröße, Mutationswahrscheinlichkeit und Selektionsdruck sind benutzerdefinierte Parameter.

Das Verfahren wurde auf eine große Anzahl künstlich erzeugter Bilddatensätze angewendet, wobei der Kontrast schrittweise verringert und das Verrauschen erhöht wurde. Probleme entstehen in erster Linie dann, wenn die Vorverarbeitung unzureichend ist oder wenn die Form der Gebiete nicht mehr oder fast nicht mehr sternförmig ist. Davon abgesehen sind die errechneten Konturen von guter Qualität. Deshalb wurde der Algorithmus auch auf reale, mikroskopische Bilddaten angewendet und die Ergebnisse sind überaus vielversprechend. Diese Untersuchung ordnet sich ein

in eine Reihe von Forschungsarbeiten zur Anwendung evolutionärer Algorithmen in der Bildsegmentierung [1], Gefäßerkennung [2] und anderer Bildverarbeitungsprobleme [3], [4].

2. Grundlagen und Problemstellung

2.1. Konturerkennung

Ein Problem der digitalen Bildverarbeitung ist das Erkennen von Konturen in Graustufenbildern. Die Problematik stellt sich sowohl in der Industrie als auch in der Medizin ein, wo aufgrund der Aufnahmeverfahren kontrastarme oder verrauschte Bilder entstehen. Genetische Algorithmen zeichnen sich in der Regel durch eine hohe Stabilität aus. Deshalb wird in dieser Arbeit ein Algorithmus entwickelt, welcher sich diese Eigenschaft zu Nutze machen soll.

Grauwertbilder werden in einer Matrix dargestellt, deren Elemente die Grauwertintensitäten der einzelnen Pixel angeben. Ein Pixel nimmt in der Regel Werte von 0 bis 255 an, wobei Null für schwarz und 255 für weiß steht. Ein Bild, welches nur die Werte 0 und 255 enthält, ist ein binäres Bild.

Als Ortsbereich eines Pixels wird die Umgebung eines Pixels bezeichnet. Dies können direkte Nachbarn sein oder aber auch deren Nachbarn. Ein Ortsbereich wird durch eine Filter- bzw. Faltungsmatrix definiert. Ein Beispiel für eine 3x3 Faltungsmatrix ist:

$$g(m,n) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \text{ mit } 1 \leq m \leq 3 \text{ und } 1 \leq n \leq 3$$

Als Kante wird eine abrupte Änderung der Grauwertintensität benachbarter Pixel bezeichnet [5]. Eine Änderung der Grauwerte wird durch die erste Ableitung indiziert. Das zweidimensionale Äquivalent zur ersten Ableitung ist der Gradient, welcher die Richtungsableitungen enthält.

$$\text{grad}(g)(x,y) = \begin{pmatrix} g_x(x,y) \\ g_y(x,y) \end{pmatrix}$$

Der Betrag des Gradienten gibt die Steigung des Grauwertgefälles in Richtung des steilsten Anstiegs an:

$$|\text{grad}(g)(x,y)| = \sqrt{(g_x(x,y))^2 + (g_y(x,y))^2}$$

Für die diskreten Werte eines Grauwertbildes kann der Gradient mit Hilfe einer Faltungsmatrix für beide Richtungen des Bildes approximiert werden. Zusätzlich kann noch eine approximierete Gaußverteilung zur Glättung in die Faltungsmatrizen eingebracht werden [5]. Diese Faltungsmatrizen bilden den Sobeloperator:

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

$$|\text{grad}(g)(x, y)| \approx \sqrt{(S_x(x, y))^2 + (S_y(x, y))^2}$$

2.2. B-Splines

Ein B-Spline ist eine aus kubischen Polynomen zusammengesetzte Funktion, deren erste und zweite Ableitung stetig sind. Ein B-Spline approximiert seine Kontrollpunkte. In der Bildverarbeitung bietet es sich an die Parameterdarstellung zu wählen:

$$G(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

Jedes Segment des Splines berechnet sich aus vier Basisfunktionen, welche für alle Segmente gleich sind, und vier Kontrollpunkten.

Wenn das Segment S_n das n -te von N Segmenten in Parameterdarstellung

$$S_n(t) = \begin{pmatrix} f_{xn}(t) \\ f_{yn}(t) \end{pmatrix}$$

ist, dann werden $N+3$ Kontrollpunkte P_k benötigt. $f_{xn}(t)$ und $f_{yn}(t)$ mit $t = [0, 1]$ berechnen sich dann wie folgt:

$$f_{xn}(t) = \sum_{i=1}^4 v_{xni} \quad f_{yn}(t) = \sum_{i=1}^4 v_{yni}$$

$$V = (v_1, v_2, v_3, v_4)^T$$

$$V_{xn} = C_{xn}T \quad V_{yn} = C_{yn}T$$

Dabei ist V_{xn} bzw. V_{yn} das Produkt einer Koeffizientenmatrix C_{xn} bzw. C_{yn} mit einem Vektor T , wobei $T = (t^3, t^2, t, 1)^T$ ist. Die Koeffizientenmatrizen C_{xn} und C_{yn} sind das Produkt einer Basismatrix M_B und einer Geometriematrix G_{xn} bzw. G_{yn} der entsprechenden Koordinaten der Kontrollpunkte:

$$C_{xn} = G_{xn} M_B$$

$$C_{yn} = G_{yn} M_B$$

$$M_B = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

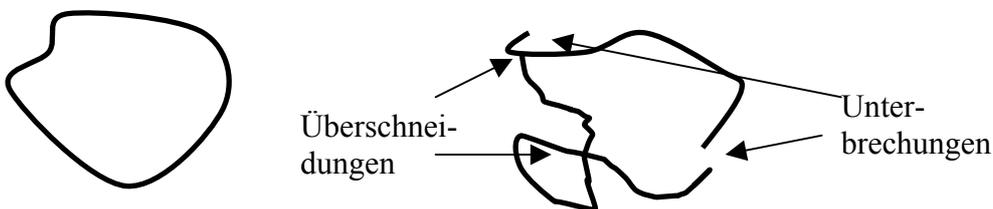
$$G_{xn} = \begin{pmatrix} P_{n,x} & 0 & 0 & 0 \\ 0 & P_{n+1,x} & 0 & 0 \\ 0 & 0 & P_{n+2,x} & 0 \\ 0 & 0 & 0 & P_{n+3,x} \end{pmatrix} \quad G_{yn} = \begin{pmatrix} P_{n,y} & 0 & 0 & 0 \\ 0 & P_{n+1,y} & 0 & 0 \\ 0 & 0 & P_{n+2,y} & 0 \\ 0 & 0 & 0 & P_{n+3,y} \end{pmatrix}$$

Der Spline ist damit für das Intervall $u=[0,N]$ definiert. $S(u) = S_n(t)$ mit $n = \text{trunc}(u)$ und $t = u-n$.

2.3. Problemstellung

Der in dieser Arbeit zu entwickelnde genetische Algorithmus soll Konturen, auch als Kanten oder Umrisse bezeichnet, von Objekten in Grauwertbildern erkennen. Dabei soll der Algorithmus möglichst wenig interaktiv sein. Da der GA jedoch die ungefähre Lage der zu erkennenden Objekte benötigt, wird zusätzlich zu dem zu untersuchenden Grauwertbild noch diejenige Information über die Lage der Objekte benötigt. Diese Information soll dem GA über ein Binärbild zukommen, welches durch eine grobe Vorsegmentierung des Grauwertbildes erzeugt wird.

Weitere Anforderung an den zu entwickelnden Algorithmus sind, dass die Konturen immer geschlossene und überschneidungsfreie Kurven sein sollen. Während die linke Abbildung diesen Anforderungen genügt, ist dies bei der rechts dargestellten Kurve nicht der Fall.

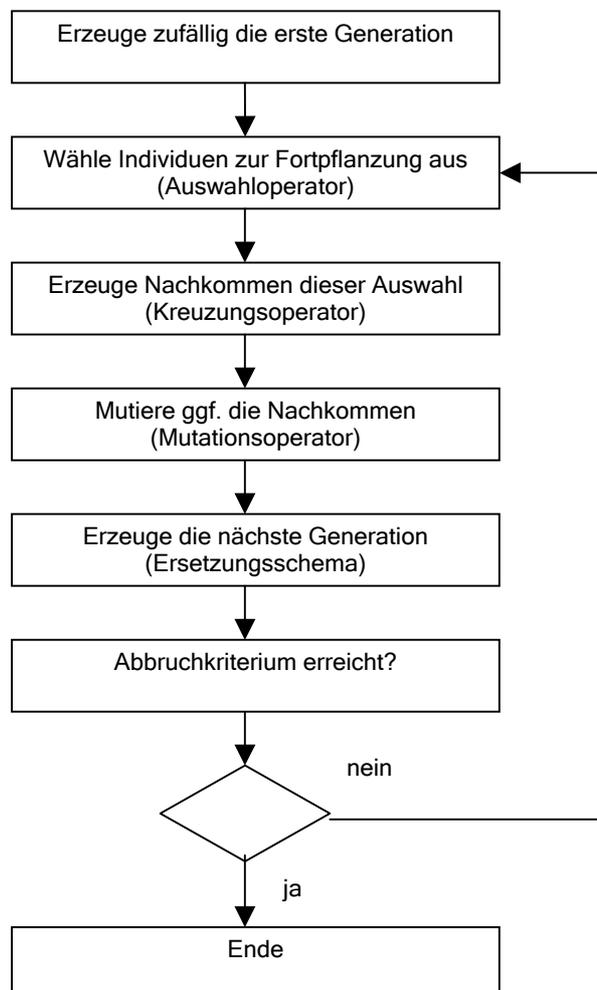


Wir beschränken unsere Untersuchung auf „sternförmige“ Objekte, der Grund für diese Restriktion wird in diesem Absatz erläutert. In der Mathematik wird eine Menge M sternförmig genannt, wenn es einen Punkt $P \in M$ gibt, so dass für alle Punkte $Q \in M$ die Verbindungslinien $V(P,Q)$ ganz in M liegen. So sind konvexe Mengen immer sternförmig und jeder Punkt P einer konvexen Menge erfüllt obige Bedingung. Im Folgenden sei M eine beschränkte, sternförmige Teilmenge der Zahlenebene und P ein Punkt aus M , der obige Bedingung erfüllt. Sei $F(\varphi)$ definiert als das Supremum (kleinste obere Schranke) der Menge $\{r > 0 : P + (r \cos(\varphi), r \sin(\varphi)) \in M\}$. F beschreibt in Polarkoordinaten eine Kurve um P , die im Rand von M verläuft. (Der Rand von M wird die Menge aller Punkte P der Zahlenebene bezeichnet, für die es Punktfolgen gibt Q_n und R_n gibt, die beide gegen P konvergieren, so dass die Punkte Q_n Element von M und die Punkte R_n nicht Element von M sind.) Allerdings kann im Allgemeinen die von F dargestellte Kurve eine echte Teilmenge des Randes von M sein. Z. B. sei die Funktion G definiert als 1 auf den rationalen und 0 auf den irrationalen Zahlen, so ist die Menge $M_G := \{(r \cos(\varphi), r \sin(\varphi)) : \varphi \in [0, 2\pi), r \in [0, F(\varphi)]\}$ sternförmig. Der Rand von M_G ist der Einheitskreis, so dass die von G dargestellte Kurve eine echte

Teilmenge von M_G und M_G eine echte Teilmenge ihres Randes ist. Falls F stetig ist (auch am Übergang $\varphi=0$ und $\varphi=2\pi$), so ist die von F dargestellte Kurve immer der komplette Rand von M . Also kann in diesem Fall der Rand als geschlossene Kurve mit Polarkoordinaten dargestellt werden. Diese Darstellung ist für unseren Algorithmus von grundlegender Bedeutung (vgl. Kapitel 3.1.)

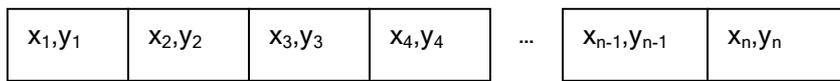
3. Darstellung des Algorithmus

Genetische Algorithmen (GA) sind randomisierte Suchstrategien, begrenzt auf einen Suchraum (siehe [6] für eine umfassende Information). Sie bedienen sich dabei der natürlichen Mechanismen der Selektion, Rekombination und der Mutation. Diese Mechanismen werden wie in der Natur dazu genutzt neue Generationen zu erzeugen. Ein GA wiederholt diesen Vorgang so oft, bis ein Abbruchkriterium erreicht ist. Die Selektion geschieht dabei über eine Fitnessfunktion, welche die Individuen auf dem Suchraum bewertet. Der GA geht dabei wie folgt vor:

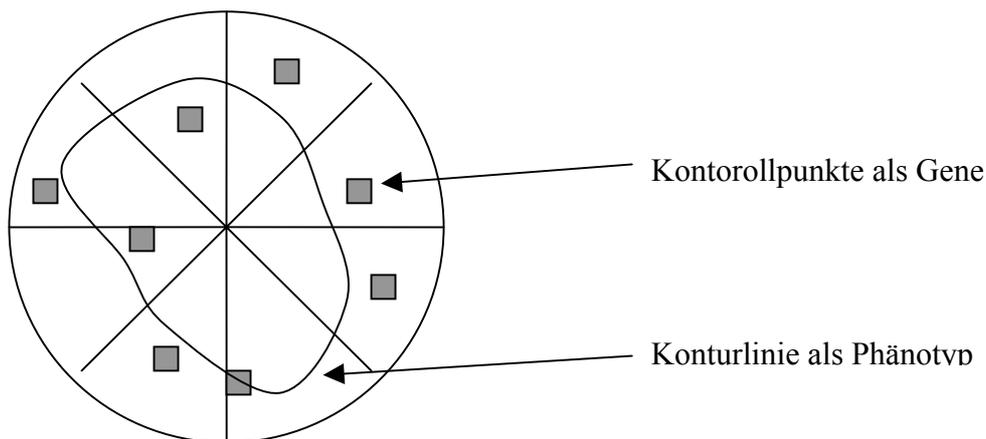


3.1 Darstellung der Individuen

Die Individuen des GA, auch Chromosom genannt, repräsentieren geschlossene Kurven, welche keine Schleifen enthalten. Die Lösung für die Forderung einer geschlossenen Kurve ist einfach gefunden: Ein geschlossener Spline bietet sich an. Obwohl ein B-Spline seine Kontrollpunkte nur approximiert, ist dieser besonders geeignet, weil B-Splines einfach berechnet werden können und gewährleistet ist, dass diese in der konvexen Hülle ihrer Kontrollpunkte liegen. Die Tatsache, dass B-Splines ihre Kontrollpunkte nicht unbedingt interpolieren ist vernachlässigbar, da für die Darstellung einer Kontur nicht die Kontrollpunkte ausschlaggebend sind, sondern nur die daraus resultierende Kurve. Da die Kontrollpunkte aber die Form der Kurve bestimmen, können die Kontrollpunkte als Gene für das Individuum bzw. Chromosom dienen. Die folgende Abbildung zeigt ein Chromosom mit seinen Genen, welche jeweils einen Kontrollpunkt eines B-Splines darstellen.



Zu verhindern, dass der dargestellte Spline eine Schlaufe durchläuft, kann dadurch realisiert werden, dass bei der Erzeugung und Mutation der Chromosomen Polarkoordinaten verwendet werden. Das Zentrum der Kontrollpunkte wird aus der BLOB-Analyse gewonnen. Hierzu ist die Voraussetzung, dass die untersuchte Menge sternförmig ist, notwendig. Jedem Kontrollpunkt wird ein gleichgroßer Winkelabschnitt zugeordnet. Bei der Erzeugung kann dann durch die Winkel gewährleistet werden, dass immer nur ein Kontrollpunkt in einem Winkelabschnitt liegt. Bei der Mutation kann verhindert werden, dass ein Kontrollpunkt seinen Winkelabschnitt verlässt. Es ist sichergestellt, dass ein Teil der Kurve durch jeden Winkelabschnitt (Segment) verlaufen wird. In der folgenden Schemadarstellung sind acht Sektoren und Kontrollpunkte als Quadrate eingezeichnet.

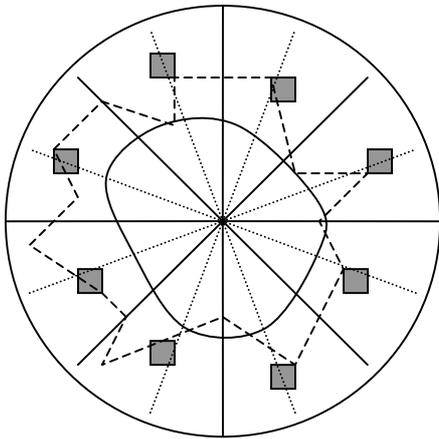


3.2 Erzeugung der Individuen

Die Individuen sollen auf zwei verschiedene Weisen erzeugt werden können. Einmal in der ungefähren Form, welche die Vorsegmentierung lieferte und einmal nach Vorlagen. Für beide Arten der Erzeugung ist die Anzahl der Gene der Chromosomen gleich.

3.2.1 Erzeugung nach BLOB

Die Position der Stützpunkte des Splines soll abhängig von der Form des vorsegmentierten Objektes sein. Dazu ist der Umriss des vorsegmentierten Objektes zu bilden und jeweils die Position desjenigen Pixels des Umrisses zu wählen, welcher der Winkelhalbierenden des entsprechenden Segmentes am nächsten liegt. Da die Kontrollpunkte aber dadurch auf dem Umriss liegen und der Spline diese nicht unbedingt durchläuft, sollen die Chromosomen in ihrer Größe skaliert werden können. Dies kann durch die Multiplikation der Radien der Kontrollpunkte in Polarkoordinaten mit einem konstanten Faktor erreicht werden. Da die Form des Umrisses mit hoher Wahrscheinlichkeit nicht genau dem Umriss des Objektes im Grauwertbild entspricht, soll zusätzlich noch jedes Gen mutiert werden.



3.2.2 Erzeugung nach Vorlage

Bei der Erzeugung nach Vorlage sollen die Chromosomen nach bestimmten vorgegebenen Formen erzeugt werden. Diese sollen sein:

- rein zufällig innerhalb der Begrenzung des Suchraumes
- kreisförmig
- nach einem Vorlagenbild, welches einen Umriss enthält.

Bei der rein zufälligen Erzeugung werden die Kontrollpunkte jeweils zufällig in ihrem entsprechenden Segment platziert.

Kreisförmig erzeugte Chromosomen sollen ihre Kontrollpunkte jeweils auf der Winkelhalbierenden der entsprechenden Segmente haben. Der gleiche Radius aller Kontrollpunkte soll innerhalb der Suchraumgrenzen zufällig gewählt werden. Jedes Gen soll aber anschließend noch einer Mutation unterzogen werden.

Wird ein Vorlagenbild verwendet, so geschieht die Erzeugung analog zu 3.2.1. Anstatt des Umrisses des Objektes wird jedoch der im Vorlagenbild gespeicherte Umriss benutzt. Da Größe und Form des Umrisses höchstwahrscheinlich nicht mit dem Umriss des vorsegmentierten Objektes übereinstimmen, muss der Umriss zuvor normiert werden. Alle Kontrollpunkte eines Umrisses aus einem Vorlagenbild werden so normiert, dass der Durchschnitt der Abstände vom Mittelpunkt 1 ist. Danach kann der Umriss (bzw. die Kontrollpunkte) neu skaliert werden, so dass Kurven unterschiedlicher Größen entstehen. Alle Gene sollen anschließend mutiert werden.

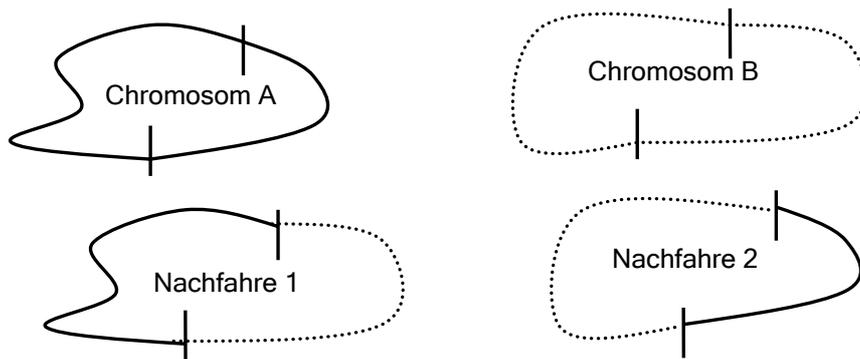
3.3 Auswahloperatoren

Durch den Auswahloperator wird bestimmt, welche Individuen sich miteinander reproduzieren dürfen. Je besser die Fitness, desto höher ist die Wahrscheinlichkeit der Auswahl (mit Ausnahme des Monte-Carlo-Operators). Es sind die folgenden Auswahloperatoren implementiert:

- Monte-Carlo (auch „Uniform-Prinzip“): Der Monte-Carlo-Operator wählt die Individuen zufällig, ohne Berücksichtigung der Fitness, aus.
- Fitnessproportionale Auswahl: Die Auswahlwahrscheinlichkeit eines Individuums ist proportional zu seiner Fitness. Dieses Verfahren hat zum Nachteil, dass besonders starke Individuen bei der Auswahl dominieren und dadurch die genetische Vielfalt verringern, was zu einer vorzeitigen Konvergenz des GA gegen ein lokales Extremum führen kann.
- Lineares Ranking-Prinzip: Beim Ranking-Prinzip ist nur die Position eines Individuums in einer nach der Fitness sortierten Liste ausschlaggebend. Dies verhindert die Dominanz der „Superindividuen“.
- Turnierprinzip: Die Individuen führen einen Wettkampf aus, dessen Sieger sich fortpflanzen dürfen: Es werden T Individuen zufällig ermittelt und das Beste darf sich fortpflanzen.

3.4 Kreuzungsoperator

Die Standardvarianten der Kreuzungsoperatoren müssen aufgrund der Problemstellung modifiziert werden. Da die Chromosomen eine geschlossene Kurve darstellen, ist ein Single-Point-Crossover wenig sinnvoll. Obwohl das Chromosom zweimal aufgetrennt werden muss (2-Point), wird aufgrund der Geschlossenheit nur ein Bereich getauscht.



3.5 Mutationsoperator

Der Mutationsoperator mutiert die Kontrollpunkte. Als Mutationsschrittweiten sind einmal eine maximale Winkeländerung und einmal eine maximale Radiusänderung anzugeben. Die Mutation geschieht also in Polarkoordinaten. Nachdem ein Gen mutiert wurde, muss noch geprüft werden, ob das Gen noch innerhalb des entsprechenden Segmentes liegt. Ist dies nicht der Fall, so soll die Mutation wiederholt werden, um Häufungen an den Rändern der Segmente zu vermeiden.

3.6 Das Ersetzungsschema

Das Ersetzungsschema ist eine Regel, nach welcher eine Folgegeneration zu erzeugen ist. Es sind die drei folgenden Ersetzungsschemata implementiert:

- General Replacement (auch „Generational Replacment): Die gesamte Folgegeneration wird durch Rekombination erzeugt.
- Starker Elitismus: Die n besten Chromosomen werden unverändert in die Nachfolgegeneration übernommen. Alle anderen werden durch Rekombination erzeugt. Dieses Schema neigt dazu, einzelne Chromosomen zur Dominanz zu verhelfen.
- Schwacher Elitismus: Die n besten Chromosomen werden mutiert in die nächste Generation übernommen. Dies soll der Entstehung von Superindividuen vorbeugen.

3.7 Die Fitnessfunktion

Da der GA Kanten erkennen soll, ist die Fitnessfunktion eine Funktion, welche die durchschnittliche Grauwertänderung aller Pixel des Splines berechnet. Dazu ist zunächst ein Fitnessoperator definiert, welcher auf dem Sobeloperator basiert. Der Fitnessoperator führt den Sobeloperator für ein Pixel auf dem ursprünglichen Graustufenbild aus. Der Rückgabewert des Fitnessoperators ist auf das Intervall $[0,1]$ begrenzt. Wobei 0 der höchstmöglichen Grauwertänderung und 1 keiner Grauwertänderung entspricht. Dies macht die Problemstellung zu einem Minimierungsproblem.

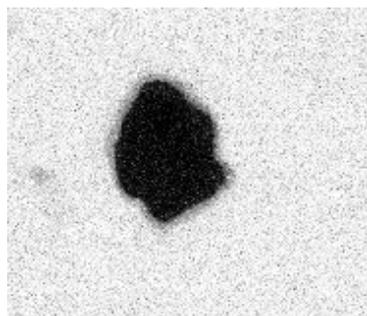
Die Berechnung der Fitness eines Chromosoms geschieht wie folgt:

1. Den Fitnesswert eines jeden Segments berechnen:
 - a. Die Länge eines Segmentes wird approximiert, indem das Splinesegment durch Geradenstücke interpoliert wird und deren Länge aufaddiert wird.
 - b. Das Splinesegment wird nochmals durch Geradenstücke interpoliert, deren Länge angegeben werden kann.
 - c. Für jedes Geradenstück wird der durchschnittliche Fitnesswert der korrespondierenden Pixel berechnet.
 - d. Der Fitnesswert eines Segmentes ergibt sich aus dem Durchschnitt der Geradenstücke.
2. Der Fitnesswert des Chromosoms ergibt sich aus dem Durchschnitt seiner Segmente.

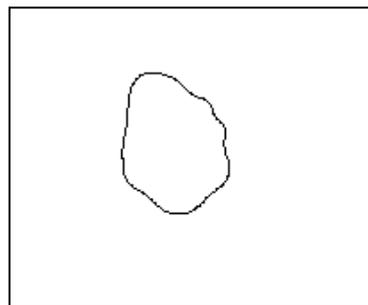
4. Erste Ergebnisse

Abschließend werden einige, erste Testergebnisse exemplarisch dargestellt. Detailliertere Tests sind in [7] enthalten. Zuerst wird das Ergebnis für ein synthetisches Bild mit geringem Rauschanteil dargestellt, welches mit den folgenden Parametern berechnet wurde:

- 50 Generationen
- Populationsgröße 50
- Chromosomen nur aus BLOB erzeugen
- Auswahloperator: Lineares Rangverfahren
- Mutationswahrscheinlichkeit: 5%
- Ersetzungsschema: Starker Elitismus mit einem Elitechromosom
- pro Generation 5 Chromosomen neu erzeugen



Originalbild

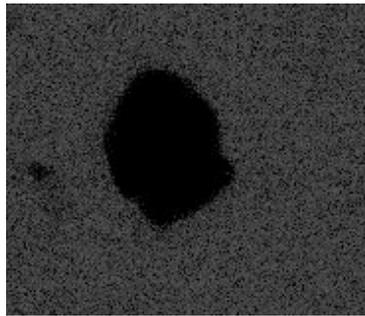


Ergebnisbild

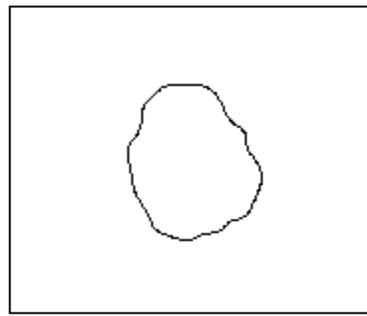
Die Kontur entspricht dem Objekt. Jedoch ist in diesem Fall das erzeugte Binärbild fast identisch mit dem Objekt, d.h. die Vorverarbeitung hat bereits wesentlich zur Erkennung der Kontur beigetragen und die Aufgabe des GA war somit recht einfach zu lösen.

Im zweiten hier dargestellten Beispiel ist das Auffinden der Kontur bereits wesentlich schwieriger, da der Kontrast wesentlich geringer als im ersten Beispiel ist. Es wurden die folgenden Parameter zur Berechnung verwendet:

- 150 Generationen
- Populationsgröße 100
- Chromosomen nur aus BLOB erzeugen
- Auswahloperator: Lineares Rangverfahren mit Selektionsdruck 80%
- Mutationswahrscheinlichkeit: 5%
- Ersetzungsschema: Starker Elitismus mit einem Elitechromosom



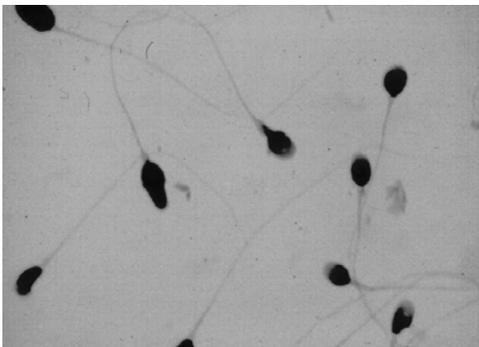
Originalbild



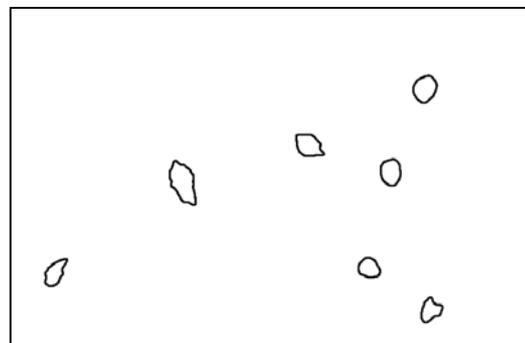
Ergebnisbild

Das nächste Bild enthält mehrere Objekte unterschiedlicher Größe und Form. Die Objekte im Binärbild stimmen in ihrer Größe nicht mit den Originalobjekten überein. Auch die Formen der Binärbildobjekte sind verzerrt. Auch dieses komplexe Problem löst der Algorithmus gut. Das Ergebnisbild wurde mit den folgenden Parametern berechnet:

- 50 Generationen
- Populationsgröße 50
- Chromosomen nur aus BLOB erzeugen
- Auswahloperator: Lineares Rangverfahren mit Selektionsdruck 80 %
- Mutationswahrscheinlichkeit: 5%
- Ersetzungsschema: Starker Elitismus mit einem Elitechromosom
- pro Generation 5 Chromosomen neu erzeugen



Originalbild



Ergebnisbild

5 Literatur

- [1] V. Metzler, R. Vandenhouten, J. Krone, R. Grebe, *Unsupervised Image Segmentation by Stochastic Reconstruction*. Proc. SPIE Conference on Medical Imaging, 1998, S. 575-586
- [2] M. Allelein, H. Brinck, C. Bucker, J. Krone, *Neue bildverarbeitende Verfahren zur Erkennung von Gefäßtopografien bei DSA-Aufnahmen*. Biomedizinische Technik, Vol. 41 Supp. 1, 1996, S. 182-183
- [3] H. Brinck, R. Kottenhoff, J. Krone, C. Strättgen, *Evolutionsalgorithmen in der medizinischen Bildverarbeitung*. in: Bildverarbeitung für die Medizin, Aachen; 1997, S. 152-157
- [4] H. Brinck, R. Kottenhoff, J. Krone, C. Strättgen, *Adaption von Hüftendoprothesen mit Methoden der Bildverarbeitung*. Biomedizinische Technik, Vol. 41 Supp. 1, 1996, S. 220-221
- [5] Encarnacao, J., Straßer, W., Klein, R., *Graphische Datenverarbeitung 2*. Oldenbourg-Verlag, München; 1997
- [6] Bäck, Thomas, Fogel, David B., Michalewicz, Zbigniew, Eds.: *Evolutionary Computing 1 und 2*. IoP Publishing, Bristol and Philadelphia; 2000
- [7] Douven, Christian, *Genetischer Algorithmus zur Erkennung von Konturen in Grauwertbildern*. Diplomarbeit, Iserlohn; 2004

Validierung hierarchischer Klassifikatoren für Mehrklassenprobleme

Markus Reischl, Ralf Mikut

Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik,
D-76021 Karlsruhe, Postfach 3640,
Telefon: (07247) 82-5749, Fax: (07247) 82-5786, E-Mail: Markus.Reischl@iai.fzk.de

1 Motivation

Es hat sich bewährt, Klassifikationsprobleme durch eine Kombination aus Merkmalsauswahl, linearer Merkmalsaggregation und statistischer Entscheidungsregel zu lösen. Meist wird versucht, alle Klassen in einem Durchlauf zu erkennen, um den Berechnungsaufwand in Grenzen zu halten (Einschrittverfahren). Sehr gut funktioniert dieses Verfahren im Zweiklassenfall, kann aber auch im Mehrklassenfall angewandt werden [1]. Probleme durch eine Vielzahl von Fehlern zwischen ähnlichen Klassen ergeben sich dann allerdings durch statistische Kriterien, die auf einem Kompromiss aus dem Durchschnitt der Abstände der Klassen zu dem Durchschnitt der Ausdehnung der Klassen aufbauen. Zwar lässt sich dieses Problem oftmals durch eine größere Anzahl von Merkmalen und eine höhere Dimension der Diskriminanzmerkmale beheben, jedoch bedingt die daraus folgende erhöhte Anzahl an Parametern auch eine große Anzahl an Beispielen, um ein Overfitting in der Lernphase zu vermeiden. Weiterhin führt eine erhöhte Anzahl an Merkmalen zu einer erhöhten Rechenzeit, welche gerade in zeitkritischen Echtzeit-Anwendungen (z. B. Mikrocontroller) nicht zur Verfügung steht. Alternative Entscheidungsregeln wie Fuzzy-Modelle und Künstliche Neuronale Netze kommen zwar ohne Merkmalsaggregation aus, erfordern aber tendenziell eine höhere Rechenzeit sowie größere Lerndatensätze wegen der höheren Parameteranzahl. Zudem sind sie bezüglich der Klassifikationsgüte nicht unbedingt überlegen.

Es bietet sich daher an, das Einschrittverfahren für das Mehrklassenproblem in mehrere (Zwei- oder Mehrklassen-)Einzelprobleme zu zerlegen (Mehrschrittverfahren) und so die Entscheidungsregeln für die einzelnen Klassen und Subklassen zu entkoppeln [2, 3] (Übersicht und Einteilung in Abschnitt 2). Verschiedene Umsetzungen wurden bereits in [4, 5] vorgestellt: Das in [4] dargestellte Verfahren versucht auf Basis von statistischen Klassifikatoren, eine Klasse nach der anderen hierarchisch vom Lerndatensatz abzuspalten (Abschnitt 3). [5] teilt ein k -Klassen-Problem in $\binom{k}{2}$ -Zweiklassenprobleme, führt die Klassifikationen durch und kumuliert anschließend alle Ergebnisse (Abschnitt 4).

Diese vorliegende Arbeit untersucht einen weiteren Ansatz, der auf Basis von statistischen Klassifikatoren gut zu erkennende Datentupel hierarchisch aussortiert und entsprechende Entscheidungsregeln automatisiert entwirft (Abschnitt 5). Der Artikel stellt auftretende Probleme sowie Lösungsvorschläge dar und diskutiert Vor- und Nachteile gegenüber den obengenannten Verfahren. Abschließend werden alle Algorithmen anhand von zehn biometrischen Datensätzen validiert, die zur Steuerung einer Mensch-Maschine-Schnittstelle bei Handprothesen eingesetzt werden (Abschnitt 6).

2 Klassifikationskonzepte

2.1 Übersicht

Der Ablauf der Klassifikation setzt sich aus den Teilschritten Merkmalsauswahl, Merkmalsaggregation und der Entscheidungsfindung mittels einer Entscheidungsregel zusammen. Um Merkmale zu erhalten, ist vorab eine Merkmalsextraktion durchzuführen.

Die *Merkmalsextraktion* dient der Ableitung bedeutungstragender Merkmale x aus Rohdaten u , die beispielsweise einer Zeitreihe entstammen können. Es ergibt sich eine Transformation

$$S_1 : (u_1, \dots, u_{s_u})^T \mapsto \mathbf{x} = (x_1, \dots, x_s)^T, \quad \mathbf{x} \in \mathbb{R}^s. \quad (1)$$

Die *Merkmalsauswahl* sucht aus den extrahierten s Merkmalen die s_m aussagekräftigsten gemäß eines Kriteriums heraus, eine Übersicht ist in [6, 7] gegeben. Mit der Bezeichnung \mathcal{I} für die Indexmenge der ausgewählten Merkmale ergibt sich die Abbildung

$$S_2 : \mathbf{x} = (x_1, \dots, x_s)^T \mapsto \mathbf{x}|_{\mathcal{I}} = (x_{i_1}, \dots, x_{i_{s_m}})^T, \quad s \geq s_m. \quad (2)$$

In dieser Arbeit kommt als Kriterium (sofern nicht anders kenntlich gemacht) die multivariate Analyse der Varianzen (MANOVA) zum Einsatz.

Das Ziel der *Merkmalsaggregation* besteht ganz allgemein darin, eine Transformation vom s_m -dimensionalen Merkmalsraum in einen s_d -dimensionalen Raum, $s_m \geq s_d$, mit einem geringen Verlust an Diskriminanzinformation zu finden

$$S_3 : \mathbf{x}|_{\mathcal{I}} = (x_{i_1}, \dots, x_{i_{s_m}})^T \mapsto \mathbf{z} = (z_1, \dots, z_{s_d})^T, \quad s_m \geq s_d, \quad \mathbf{z} \in \mathbb{R}^{s_d}. \quad (3)$$

Eine Übersicht ist [8] zu entnehmen. Zur Aggregation wird (sofern nicht anders kenntlich gemacht) die Diskriminanzanalyse betrachtet, die mittels der Transformationsmatrix \mathbf{A} linear Merkmale aggregiert [1]

$$\mathbf{Z} = \mathbf{X}|_{\mathcal{I}} \cdot \mathbf{A}, \quad \mathbf{X}|_{\mathcal{I}} = (\mathbf{x}|_{\mathcal{I},1}, \dots, \mathbf{x}|_{\mathcal{I},n})^T, \quad \mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)^T. \quad (4)$$

Die Entscheidungsregel weist schließlich dem so entstandenen Merkmalsatz eine Klasse y zu, dieser Schätzwert soll als \hat{y} bezeichnet werden

$$S_4 : \mathbf{z} = (z_1, \dots, z_{s_d})^T \mapsto \hat{y} \in \{1, \dots, k\}, \quad k \in \mathbb{N}^+. \quad (5)$$

Eine Übersicht geben [9, 10]. In dieser Arbeit wird ausschließlich die approximierete Maximum-Likelihood-Entscheidungsregel (AML) betrachtet

$$\hat{y}(\mathbf{z}) = \operatorname{argmax}_{1 \leq j \leq k} f_j(\mathbf{z}; \bar{\mathbf{z}}_j, \hat{\Sigma}_j) \quad (6)$$

$$= \operatorname{argmax}_{1 \leq j \leq k} \frac{1}{(2\pi)^{\frac{s}{2}} \sqrt{\det \hat{\Sigma}_j}} e^{-\frac{1}{2}(\mathbf{z} - \bar{\mathbf{z}}_j)^T \hat{\Sigma}_j^{-1} (\mathbf{z} - \bar{\mathbf{z}}_j)}. \quad (7)$$

Es stellen $\bar{\mathbf{z}}_j$ den Mittelwertvektor von Klasse j und $\hat{\Sigma}_j$ die geschätzte Kovarianzmatrix der Klasse j dar.

Die Anpassung des Klassifikationssystems (Anlernen) geschieht durch die Bestimmung der Parameter \mathcal{I} , \mathbf{A} , $\bar{\mathbf{z}}_j$ und $\hat{\Sigma}_j$.

2.2 Hierarchische Klassifikation

Während zur direkten Lösung des Mehrklassenproblems (Polychotomie) für jedes Datentupel \mathbf{x} die gleichen Abbildungen S_2 bis S_4 zum Einsatz kommen, betrachten hierarchische Klassifikatoren auf der Hierarchiestufe m individuelle Abbildungen $S_{2,m}$ bis $S_{4,m}$. Beruhen diese Abbildungen auf der Extraktion von Datensubgruppen, so kommen sogenannte Teile-und-Herrsche (engl. Separate-and-Conquer, Divide-and-Conquer)-Algorithmen zum Einsatz [3]. Eine Aufteilung des Klassifikationsproblems im Sinne des Erkennens bzw. Extrahierens von Klassensubgruppen führt zur Anwendung von Aufschlussverfahren (engl. decomposition methods) [11, 12]. Besondere Aufmerksamkeit hat hierbei die Aufteilung in Zweiklassenprobleme (Dichotomien) erlangt [2], und darunter die paarweise Kombination (engl. pairwise coupling) [13], die im Weiteren als Turnier-Klassifikation bezeichnet werden soll.

Die Hierarchie entsteht bei diesen Methoden durch

1. ein Abspalten *aller* Datentupel einer ausgewählten Klasse $j \in \{1, \dots, k\}$ (bezüglich des Mehrklassenproblems). Findet keine Abspaltung statt, folgt in der nächsten Hierarchiestufe der Versuch der Abspaltens einer anderen Klasse. Sobald ein unbekanntes Datentupel zu einer abgespaltenen Klasse als zugehörig befunden wird, folgt die finale Zuweisung des Tupels zu dieser Klasse.
2. eine Dekomposition des Lerndatensatzes in $\binom{k}{2}$ -Zweiklassenprobleme. Beim Turnier-Kriterium werden die Entscheidungen *aller* möglichen Zweiklassenprobleme aggregiert. Beim KO-Kriterium führen die Entscheidungen *einzelner* Zweiklassenprobleme zur Auswahl des nächsten zu betrachtenden Zweiklassenproblems, die unterlegene Klasse wird endgültig verworfen.
3. ein Abspalten von *ausgewählten* Datentupeln mit einer Mindestzugehörigkeit q_c zu einer Klasse j . Verbleibende Datentupel der Klasse j werden von dieser Entscheidung nicht beeinflusst. Ein Abspalten bedingt eine finale Zuweisung eines Datentupels zur Klasse j .

Methode 1 strebt danach, komplette Klassen zuverlässig zu erkennen (Hierarchische Klassifikation von Klassen, Abschnitt 3). Methode 2 versucht durch eine Hierarchie von Zwei-Klassen-Problemen eine Klassifikationsentscheidung herbeizuführen. Die Betrachtung von Klassensubgruppen führt zum Entwurf von Turnier-Klassifikatoren (Abschnitt 4). Methode 3 sucht nach einzelnen zuverlässig zu erkennenden Datentupeln (Hierarchische Klassifikation von Datentupeln, Abschnitt 5).

Alle angeführten Verfahren stützen sich auf die in Abschnitt 2.1 angeführten statistischen Methoden, da hierbei ein leicht verständlicher Parameterzugang und ein geringer Berechnungsaufwand gegeben sind. Allerdings eignen sie sich prinzipiell für beliebige Entscheidungsregeln.

2.3 Entwurf und Anwendung hierarchischer Klassifikatoren

Der in dieser Arbeit betrachtete hierarchische Entwurf gliedert sich gemäß folgendem Schema (Bild 1): Zu Beginn besteht optional die Möglichkeit einer Merkmalsvorauswahl (Schritt 1), um die Komplexität für den nachfolgenden Klassifikatorentwurf zu

reduzieren - alle folgenden Schritte beziehen sich dann auf einen Lerndatensatz mit diesen gewählten Merkmalen. Es folgt eine Endlosschleife (Schritt 2), wobei jeder Schleifendurchgang einer Erhöhung der Hierarchiestufe m entspricht. In den Hierarchiestufen selbst ist dann eine eigenständige Folge von Abbildungen $S_{2,m}$ bis $S_{4,m}$ zu entwerfen: Sofern keine Bedingungen für einen Abbruch gegeben sind, wird eine Klassifikationsstrategie inklusive des entsprechenden Lerndatensatzes ausgewählt (Schritt 2a). Darauf basierend findet eine Merkmalsauswahl statt (Schritt 2b). Eine Modifikation des Lerndatensatzes (Schritt 2c) kann nun für die folgende Merkmalsaggregation (Schritt 2d) notwendig sein, um Parameterschätzungen robuster zu machen. Es folgt der Entwurf einer Entscheidungsregel (Schritt 2e) und darauf basierend eine erneute Modifikation des Lerndatensatzes (Schritt 2f).

Die Klassifikation eines unbekanntes Datentupels (Anwendung des Klassifikators) beschränkt sich dann auf die Auswahl der in Schritt 1 berechneten Merkmalsindizes (Schritt 3), sowie innerhalb einer Schleife (Schritt 4) auf die Auswahl eines Klassifikators (Schritt 4a) und die Anwendung der zugehörigen Merkmalsauswahl, -aggregation und Entscheidungsregel (Schritt 4b). Die Schleife wird dann abgebrochen, wenn eine Klassenzuweisung geschehen ist. Diese Klassenzuweisung wird in der letzten Hierarchiestufe forciert.

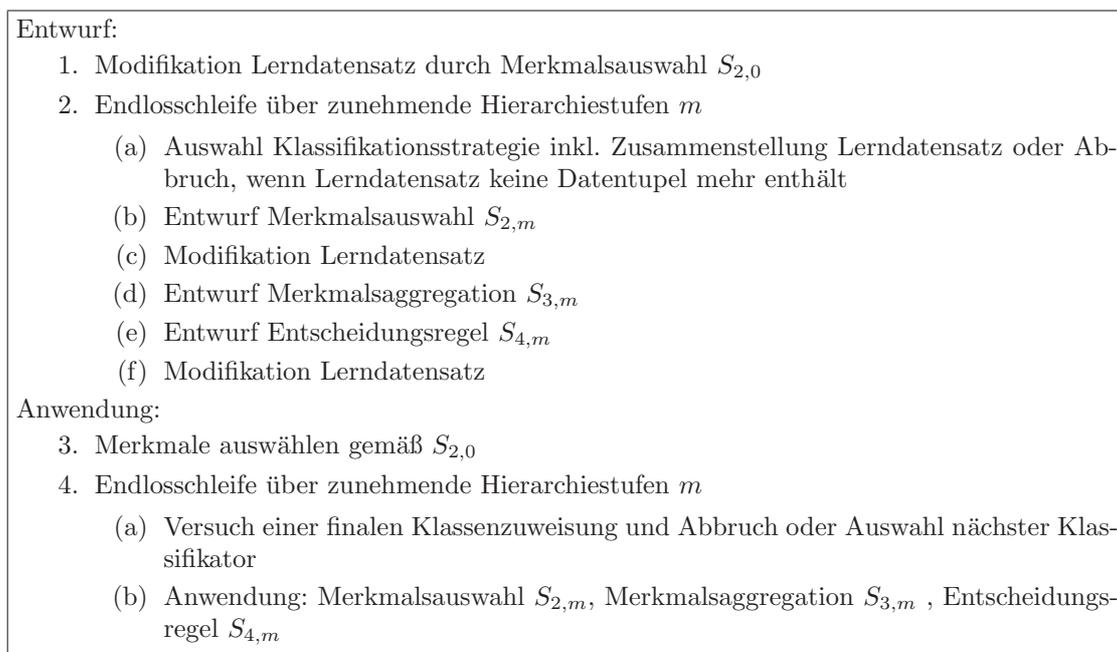


Bild 1: Algorithmus zum hierarchischen Klassifikatorentwurf

3 Hierarchische Extraktion von Klassen

Einen Algorithmus, der hierarchisch ganze Klassen aus dem Datenmaterial extrahiert, beschreibt [4]. Der Algorithmus (Bild 2) startet mit einer Merkmalsauswahl nach MANOVA (Schritt 1) zur Beschränkung des Lerndatensatzes auf s_m relevante Merkmale. Innerhalb der Endlosschleife (Schritt 2) wird dann zunächst geprüft,

ob bereits alle Lerndaten einer Klasse zugeordnet wurden oder ob die Anzahl auszuwählender Merkmale ein vorgegebenes Maximum erreicht hat (Schritt 2a). Im ersten Fall bricht der Algorithmus ab, im zweiten werden die verbliebenen Datensätze gemäß (2)-(5) direkt klassifiziert. Ansonsten findet in Schritt 2b die Merkmalsauswahl auf Basis der verbliebenen Datentupel statt. Die Anzahl zu betrachtender Merkmale wird sukzessive in jedem Schleifendurchlauf erhöht. Ziel der Merkmalsauswahl ist es, mindestens eine Klasse gut zu erkennen, um sie abtrennen zu können. Der folgende Klassifikatorentwurf verwendet nun den vorhandenen Lerndatensatz (Schritt 2c entfällt), um vorab mit der Merkmalsaggregation auf s_d Merkmale nochmals eine Klasse von den anderen abzusetzen (Schritt 2d). Auf den aggregierten Merkmalen baut eine AML-Entscheidungsregel auf (Schritt 2e). Wenn sich nun eine Klasse im Sinne eines Kriteriums deutlich von den anderen unterscheidet, wird ein Klassifikator entworfen, der die verbliebenen Datensätze der aktuellen Hierarchiestufe in die separierbare Klasse j und die Restklassen trennt. Die der Klasse j angehörenden n_j Datentupel werden danach aus dem Datenmaterial eliminiert (Schritt 2f).

Die Anwendung dieses Klassifikators findet wie in Abschnitt 2.3 beschrieben statt, in jeder Hierarchiestufe wird dabei der entsprechende Klassifikator ausgewählt. Eine Klassenzuweisung geschieht, wenn die in der Hierarchiestufe zu bestimmende Klasse tatsächlich zugewiesen wurde oder wenn die letzte Hierarchiestufe erreicht ist.

<p>Entwurf:</p> <ol style="list-style-type: none"> 1. MANOVA $S_{2,0}$ 2. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Wenn zuviele Merkmale, dann Datensätze direkt gemäß (2)-(5) klassifizieren (b) maximiere Abstand der Klassen (nach [4, 14]) (c) - (d) maximiere Abstand der Klassen (nach [4, 14]) (e) AML-Entscheidungsregel (f) wenn eine Klasse abgespalten wurde, entsprechende Tupel aus LDS entfernen <p>Anwendung:</p> <ol style="list-style-type: none"> 3. Merkmale auswählen gemäß $S_{2,0}$ 4. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Versuch einer finalen Klassenzuweisung (wenn Klassenzugehörigkeit = vorgegebene Klassenzugehörigkeit der Hierarchiestufe) und Abbruch oder Auswahl nächster Klassifikator (b) Anwendung: Merkmalsauswahl $S_{2,m}$, Merkmalsaggregation $S_{3,m}$, Entscheidungsregel $S_{4,m}$
--

Bild 2: Hierarchischer Klassifikatorentwurf gemäß Bild 1 zum Abtrennen von Klassen

4 Turnier- und KO-Klassifikation

Einen Algorithmus (Bild 3) zur Turnier- und KO-Klassifikation beschreibt [5]. Dieser beginnt mit einer Merkmalsauswahl nach MANOVA, mittels der der gesamte Lerndatensatz auf seine wesentlichen s_m Merkmale reduziert wird (Schritt 1). Ziel

ist es, Klassifikatoren für sämtliche Kombinationen von Klassen zu finden und somit die Polychotomie in eine Serie von Dichotomien zu überführen. Sofern also noch nicht alle paarweisen Klassifikatoren entwickelt wurden, werden nur die Datentupel zweier Klassen i, j als Lerndatensatz ausgewählt (Schritt 2a). Auf deren Basis wählt das MANOVA-Verfahren optional s_m^* wichtige Merkmale aus (Schritt 2b). Unter Verwendung desselben Lerndatensatzes (Schritt 2c) wird eine Merkmalsaggregation auf s_d Merkmale durchgeführt, mit dem Ziel, eine möglichst gute lineare Diskriminanzfunktion [15] zu erhalten (Schritt 2d). Anschließend wird die AML-Entscheidungsregel entworfen (Schritt 1). Die KO-Klassifikation fordert ebenfalls den Entwurf aller paarweisen Klassifikatoren, weil sich erst in der Anwendung entscheidet, welche Paare auszuwerten sind.

Beim Anwenden des Klassifikators werden aus einem unbekanntem Datentupel die Merkmale aus Schritt 1 ausgewählt. Anschließend wird der Klassifikator für die Klassenkombination i, j gewählt. Bei der Turnier-Klassifikation werden alle Klassenkombinationen durchgespielt. Beim KO-System treten in jedem Schleifendurchlauf zwei Klassen gegeneinander an, wobei die unterlegene Klasse final verworfen wird. Im ersten Fall wird eine Klassenzugehörigkeit des Datentupels auf Basis von kumulierten Klassenzugehörigkeiten nach Durchlauf aller Klassenkombinationen gefällt (Turnier-Kriterium), im zweiten Fall bestimmt sich die Klassenzugehörigkeit durch den Gewinner des KO-Systems (KO-Kriterium). Die einzelnen Klassenzugehörigkeiten berechnen sich aus der entsprechenden Merkmalsauswahl, -aggregation und dem Anwenden der Entscheidungsregel (Schritt 4b).

<p>Entwurf:</p> <ol style="list-style-type: none"> 1. MANOVA $S_{2,0}$ 2. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Wenn alle Klassenkombinationen i, j getestet, dann Abbruch. Auswahl von allen Datensätzen \mathbf{x} der Klassen $i \in \{1, \dots, k-1\}$ und $j \in \{i+1, \dots, k\}$ (b) MANOVA (c) - (d) finde beste lineare Diskriminanzfunktion (e) AML-Entscheidungsregel (f) - <p>Anwendung:</p> <ol style="list-style-type: none"> 3. Merkmale auswählen gemäß $S_{2,0}$ 4. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Versuch einer finalen Klassenzuweisung (Turnier: Zuweisung kumulierte Klassenzugehörigkeit, wenn alle Klassenkombinationen getestet wurden; KO: wenn alle Klassen außer einer final verworfen wurden) und Abbruch oder Auswahl nächster Klassifikator (Auswahl Klassifikator für die Klassen i, j, Turnier: Weiterzählen, KO: siegreiche Klasse vorherige Hierarchiestufe und noch nicht verworfene Klasse) (b) Anwendung: Merkmalsauswahl $S_{2,m}$, Merkmalsaggregation $S_{3,m}$, Entscheidungsregel $S_{4,m}$

Bild 3: Hierarchischer Klassifikatorentwurf gemäß Bild 1 zur Turnier- und KO-Klassifikation

5 Hierarchische Extraktion von Datentupeln

5.1 Übersicht

Die in den vorangegangenen Abschnitten beschriebenen Verfahren sollen nun mit der hierarchischen Extraktion einzelner Datentupel verglichen werden. Dieses Verfahren versucht schrittweise, gut zu erkennende Datentupel aus dem Lerndatensatz zu extrahieren und auf Basis der verbliebenen sowie zusätzlicher Merkmale einen neuen Klassifikator zum Abspalten weiterer Datentupel zu entwerfen. Sie verzichtet dabei auf eine Vorauswahl von Merkmalen gemäß Schritt 1. Der Entwurf neuer Hierarchien bricht ab, wenn die maximale Hierarchiestufe m_{\max} erreicht wurde oder wenn alle Datentupel klassifiziert wurden. Innerhalb einer Hierarchiestufe erweitert das MANOVA-Verfahren die bestehende Merkmalsauswahl um ein aussagekräftiges Merkmal auf Basis des verbliebenen Lerndatensatzes. Da die folgenden Schritte Entwurf der Merkmalsaggregation (Schritt 2d) und Entscheidungsregel (Schritt 2e) für robuste Schätzungen eine Mindestanzahl an Lerndaten benötigen, besteht in Schritt 2c die Möglichkeit, den Lerndatensatz zu modifizieren. Schließlich werden alle verbliebenen Datentupel aus dem Lerndatensatz entfernt, die eine vorgegebene minimale Klassenzugehörigkeit überschreiten (Schritt 2f).

Bei der Anwendung des Klassifikators findet in jeder Hierarchiestufe die Auswahl des entsprechenden Klassifikators statt. Eine Klassenzuweisung geschieht, wenn die Klassenzugehörigkeit des unbekanntes Datentupels eine minimale Klassenzugehörigkeit übersteigt oder wenn die letzte Hierarchiestufe erreicht ist.

- | |
|--|
| <ol style="list-style-type: none">1. -2. Endlosschleife über zunehmende Hierarchiestufen m<ol style="list-style-type: none">(a) Wenn maximale Hierarchiestufe erreicht, dann Datensätze direkt gemäß (2)-(5) klassifizieren(b) MANOVA mit $s_m = s_{m,\min} + m - 1$ Merkmalen(c) Erweiterung Lerndatensatz durch bereits klassifizierte Datentupel aus dem Lerndatensatz der Hierarchiestufe $m-1$ für robuste Schätzung, wenn zu wenige Datentupel für bestimmte Klassen vorhanden sind(d) Diskriminanzanalyse(e) AML-Entscheidungsregel(f) Entfernen erfolgreich klassifizierter Datentupel aus dem alten Lerndatensatz <p>Anwendung:</p> <ol style="list-style-type: none">3. -4. Endlosschleife über zunehmende Hierarchiestufen m<ol style="list-style-type: none">(a) finale Klassenzuweisung (wenn Klassenzugehörigkeit $>$ min. Klassenzugehörigkeit) und Abbruch oder Auswahl Klassifikator für Hierarchiestufe m(b) Anwendung: Merkmalsauswahl $S_{2,m}$, Merkmalsaggregation $S_{3,m}$, Entscheidungsregel $S_{4,m}$ |
|--|

Bild 4: Hierarchischer Klassifikatorentwurf gemäß Bild 1 zur Extraktion von Datentupeln

5.2 Algorithmus

Es soll nun detailliert auf die Funktionsweise des Algorithmus eingegangen werden. Die Indexmenge \mathcal{J} enthalte die Indizes aller noch nicht klassifizierten Datentupel der Hierarchiestufe m und ist somit zu Anfang

$$\mathcal{J}_m = \{1, \dots, n\}. \quad (8)$$

Weiterhin enthalte die Indexmenge $\mathcal{J}_{j,m}$ die noch nicht klassifizierten Datentupel der Klasse j

$$\mathcal{J}_{j,m} = \{i | y(i) = j\}, \quad i \in \mathcal{J}_m. \quad (9)$$

Auf Basis der bislang nicht klassifizierten Datentupel $\mathbf{X}|_{\mathcal{J}_m}$ wird nun die Indexmenge aussagekräftiger Merkmale nach MANOVA um ein Merkmal erweitert

$$\mathbf{X}|_{\mathcal{J}_m}, \mathbf{y}|_{\mathcal{J}_m}, \mathcal{I}_{m-1} \mapsto \mathcal{I}_m, \quad \text{card}(\mathcal{I}_m) = s_m. \quad (10)$$

Um anschließend Transformationsmatrix und Entscheidungsregel zu entwerfen, ist im Gegensatz zu (10) sicherzustellen, dass jede Klasse durch eine Mindestanzahl an Datentupeln n_c vertreten ist. Zwar passt sich der zu entwerfende Klassifikator nicht mehr individuell an die verbliebenen Tupel an, jedoch sind die Schätzungen seiner Parameter wesentlich robuster.

Die Indizes der Klassenvertreter der Klasse j im m -ten Durchlauf befinden sich in $\mathcal{J}_{j,m}$, die Klassenvertreter zum Anlernen seien als $\mathcal{J}_{j,m}^*$ bezeichnet, die Gesamtheit aller Datentupel zum Anlernen sei als \mathcal{J}_m^* bezeichnet. Eine Darstellung dieser Mengen zeigt Bild 5. Es gilt

$$\mathcal{J}_{j,m}^* = \begin{cases} \mathcal{J}_{j,m-1}^* & \text{für } 0 < \text{card}(\mathcal{J}_{j,m}) < n_c \\ \mathcal{J}_{j,m} & \text{sonst,} \end{cases} \quad (11)$$

$$\mathcal{J}_m^* = \bigcup_{j=1}^k \mathcal{J}_{j,m}^*. \quad (12)$$

Auf Basis der ausgewählten Merkmale \mathcal{I}_m und der zu verwendenden Datentupel \mathcal{J}_m^* wird nun eine Transformationsmatrix \mathbf{A}_m für die Merkmalsaggregation gesucht. Sofern die Anzahl ausgewählter Merkmale die Anzahl zu aggregierender Merkmale übersteigt, d. h. $s_m > s_d$ ist \mathbf{A}_m durch eine Diskriminanzanalyse zu bestimmen, ansonsten wird sie als Einheitsmatrix angenommen

$$\mathbf{X}|_{\mathcal{I}_m, \mathcal{J}_m^*}, \mathbf{y}|_{\mathcal{J}_m^*} \mapsto \mathbf{A}_m. \quad (13)$$

Nach Durchführung der Transformation (4) mit $\mathbf{A} = \mathbf{A}_m$ und $n = \text{card}(\mathcal{J}_m^*)$ können schließlich die Parameter der Entscheidungsregel $\bar{\mathbf{z}}_{j,m}$ und $\hat{\Sigma}_{j,m}$ geschätzt werden

$$\mathbf{Z}|_{\mathcal{J}_m^*}, \mathbf{y}|_{\mathcal{J}_m^*} \mapsto \bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m}. \quad (14)$$

Für alle in \mathcal{J}_m enthaltenen Datenindizes i wird nun der Wert des Entscheidungsmaßes $f_j(\mathbf{z}_i; \bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m})$ in (6) für die wahre Klassenzuordnung zur Summe der Entscheidungsmaße aller Klassenzuordnungen ins Verhältnis gesetzt. Ein ähnlicher Ansatz wurde bereits in [4] diskutiert

$$p_{j,m}(\mathbf{z}_i) = \frac{f_j(\mathbf{z}_i; \bar{\mathbf{z}}_{y_i,m}, \hat{\Sigma}_{y_i,m})}{\sum_{j=1}^k f_j(\mathbf{z}_i; \bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m})}, \quad i \in \mathcal{J}_m, \quad p_{j,m}(\mathbf{z}_i) \in [0, 1]. \quad (15)$$

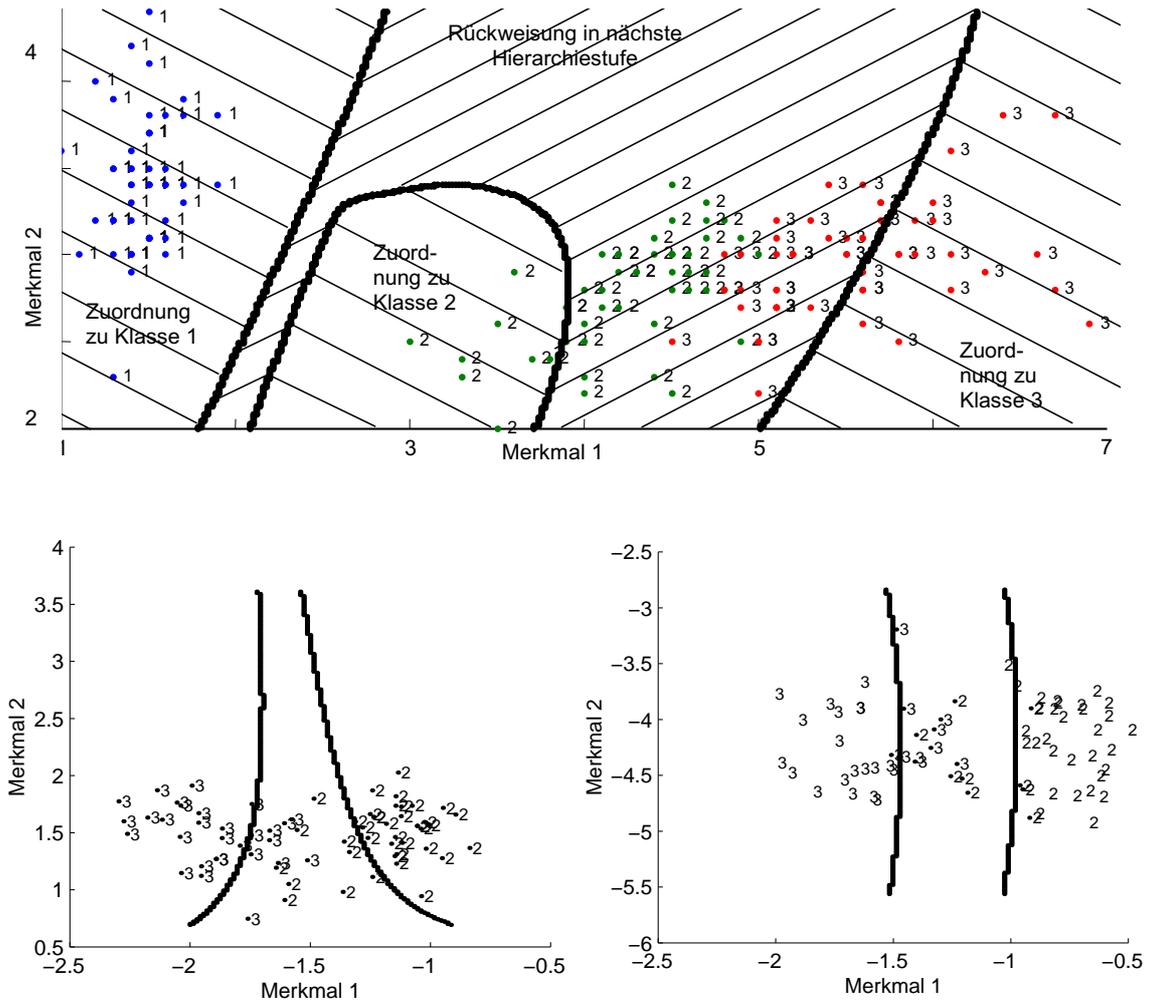


Bild 5: Hierarchische Klassifikation am Beispiel des IRIS-Datensatzes [16]. Oben: 1. Hierarchiestufe mit Auswahl von zwei Merkmalen, unten links: 2. Hierarchiestufe, Auswahl von drei Merkmalen, Reduktion auf zwei, unten rechts: 3. Hierarchiestufe, Auswahl von vier Merkmalen, Reduktion auf zwei. Alle Datentupel mit Nummer entsprechen $\mathcal{J}_{j,m}^*$ bzw. \mathcal{J}_m^* , alle Datentupel mit Punkt entsprechen $\mathcal{J}_{j,m}$ bzw. \mathcal{J}_m . Fehlklassifikationsrate bei 10-facher Crossvalidierung (1000 Wiederholungen) mit $s_{m,\min} = 3$, Aggregation auf $s_d = 3$ Merkmale, Anzahl Durchläufe $m_{\max} = 1$, Mindestkonfidenz $q_c = 0.999$, Mindesttupelzahl $n_c=20$: 2.34 ± 0.63 , Vergleich mit AML mit $s_m = 4$, $s_d = 3$: 2.60 ± 0.43 .

Übersteigt die maximale relative Klassenzugehörigkeit

$$q(\mathbf{z}_i) = \max_j p_{j,m}(\mathbf{z}_i) \quad (16)$$

eine vorgegebene Konfidenz $q_c \in [0.5; 1]$, so kann i aus der Indexmenge zu betrachtender Merkmale \mathcal{J}_m entfernt werden

$$\mathcal{J}_m = \begin{cases} \{\mathcal{J}_m \setminus i\} & \text{für } q(\mathbf{z}_i) > q_c \\ \mathcal{J}_m & \text{sonst} \end{cases}, \quad i \in \mathcal{J}_m \quad (17)$$

Ist die maximale Hierarchiestufe m_{\max} erreicht, so werden alle in \mathcal{J}_m verbliebenen Datentupel gemäß (2) - (5) klassifiziert. Ist dies nicht der Fall, so erhöht sich der Zähler m um eins und der Algorithmus wird von vorne gestartet.

<p>Gegeben:</p> <p>$m = 1$: Hierarchiestufe,</p> <p>m_{\max}: maximale Anzahl Durchläufe,</p> <p>$s_{m,\min}$: Mindestanzahl auszuwählender Merkmale,</p> <p>$s_m = s_{m,\min}$: Anzahl auszuwählender Merkmale,</p> <p>n_c: Minimal erlaubte Kardinalität einer Klasse j,</p> <p>q_c: Minimale Konfidenz zum Erkennen einer Klasse.</p> <ol style="list-style-type: none"> 1. Solange $\text{card}(\mathcal{J}_m) \neq 0$ 2. Finde verbliebene Datentupel $\mathcal{J}_{j,m}$ jeder Klasse j. 3. Finde neues relevantes Merkmal zur Klassifikation der verbliebenen Datentupel: \mathcal{I}_m. 4. Für jede Klasse j 5. Wenn Anzahl Datentupel zu klein: $0 < \text{card}(\mathcal{J}_{j,m}) < n_c$, dann 6. Lerndatensatz der Stufe $m-1$ verwenden: $\mathcal{J}_{j,m}^* = \mathcal{J}_{j,m-1}^*$ 7. sonst $\mathcal{J}_{j,m}^* = \mathcal{J}_{j,m}$. 8. Lerndatensätze aller Klassen verschmelzen: \mathcal{J}_m^*. 9. Transformationsmatrix mit $\mathbf{X}_{ \mathcal{I}_m, \mathcal{J}_m^*}$ für Merkmalsaggregation berechnen: \mathbf{A}_m. 10. Transformation für verbliebene Datentupel durchführen: $\mathbf{Z}_{ \mathcal{J}_m^*}$. 11. Parameter für Entscheidungsregel berechnen: $\bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m}$. 12. Für alle bislang nicht klassifizierten Datentupel $i \in \mathcal{J}_m$ 13. Gewissheit der Klassenzuordnung berechnen: $q(\mathbf{z}_i)$. 14. Wenn Klassenzuordnung gewiss: $q(\mathbf{z}_i) > q_c$, dann 15. Index des Datentupels i aus \mathcal{J}_m entfernen. 16. Wenn maximale Hierarchiestufe erreicht: $m = m_{\max}$, dann 17. Klassifikation aller verbliebenen Datentupel gemäß (2)-(5). Ende. 18. Hierarchiestufe erhöhen: $m = m + 1$ 19. Anzahl zu betrachtender Merkmale erhöhen: $s_m = s_m + 1$. 20. Ende.

Bild 6: Detaillierter Algorithmus zum hierarchischen Klassifikatorentwurf

Erweitern lässt sich der beschriebene Algorithmus, indem in aufeinander folgenden Hierarchiestufen keine Erweiterung der Merkmalsauswahl stattfindet. Auf diese Weise können die gleichen Merkmale verwendet werden, um mit modifizierten Diskriminanzfunktionen andere Datentupel zu extrahieren.

6 Ergebnisse

Die vorgestellten Algorithmen sollen nun anhand von 10 Patientendatensätzen zur Steuerung einer Funktionsprothese evaluiert werden. Für eine derartige Steuerung generiert ein Anwender myoelektrische Signale, d. h. Muskelkontraktionen der Stumpfmuskulatur, die seiner Bewegungsabsicht entsprechen [17, 18]. Da die Stumpfmuskulatur jedoch meist nur noch maximal zwei unabhängige Muskelgruppen enthält, ist eine Umsetzung von mehr als zwei bzw. drei Bewegungsmöglichkeiten schwierig. Deshalb wird in [19] ein Konzept vorgeschlagen, bei dem der Patient seine Bewegungsabsicht durch einen spezifischen Verlauf der Muskelkontraktion mitteilt. Das Auswerten dieser Bewegungsabsicht geschieht durch ein Klassifikationssystem,

	# Multiplikationen	# Parameter	# Multiplikationen	# Parameter
DI	$s_m s_d + s_d(s_d + 1)k$	$s_m s_d + \frac{k}{2} s_d(s_d + 3)$	96	78
TN	$\frac{1}{2}k(k-1)(s_m^* + 4)$	$\frac{1}{2}k(k-1)(s_m^* + 4)$	105	105
KO	$(k-1)(s_m^* + 4)$	$\frac{1}{2}k(k-1)(s_m^* + 4)$	35	105
HK	nicht fest	nicht fest	126	99
HKD	nicht fest	nicht fest	189	153
KNN	$(s_m + k)n_n$	$(s_m + k + 1)n_n + k$	168	186
Fuzzy	nicht fest	nicht fest	-	-

Tabelle 1: Anzahl Multiplikationen bei der Anwendung des Klassifikators und notwendige Parameteranzahl, rechts: Beispiel für $s_m = 8$, $s_d = 3$, $s_m^* = 3$, $k = 6$, $n_n = 12$, bei HK und HKD wurden Beispiele ausgewertet

da hierbei Parameter individuell und automatisch anhand von Beispieldaten an den Patienten angepasst werden können.

Da die Auswertung der Bewegungsabsicht auf einem Mikrocontroller stattfindet, ist nicht nur die Klassifikationsgüte der einzelnen Verfahren von Bedeutung, sondern auch die Anzahl der zu speichernden Parameter sowie die Komplexität der Berechnung (insbesondere die Anzahl der Multiplikationen).

Eine Übersicht gibt Tabelle 1. DI bezeichnet hierbei den herkömmlichen AML-Klassifikator mit vorgeschalteter Diskriminanzanalyse (DI) gemäß (2), (3) und (5). Mit HK wird Bezug auf die hierarchische Extraktion von Klassen (Abschnitt 3) genommen, TN und KO sind Turnier-Klassifikatoren gemäß Abschnitt 4 mit Turnier- bzw. KO-Kriterium. HKD bezeichne schließlich die hierarchische Klassifikation von Datentupeln. Vergleichend werden Multilayer Perceptrons (MLPs) mit einer verdeckten Schicht (Tansig-Übertragungsfunktion, $n_n = 12$ Neuronen, Bias) und einem Ausgangsneuron (lineare Übertragungsfunktion, Bias) pro Klasse sowie Fuzzy-Regelbasen nach [20] betrachtet.

Die Klassifikationsergebnisse über 10 zufällig gewählten Patientendatensätzen zeigt Tabelle 2 [21]. Es gelten folgende Parametrierungen:

- DI: Auswahl von $s_m = 8$, Aggregation auf $s_d = 3$ Merkmale,
- Fuzzy: Verwendung aller Merkmale,
- KNN: Auswahl von $s_m = 8$ Merkmalen, $n_n = 12$ Zwischenschichtneuronen,
- TN: Vorauswahl von $s_m = 8$, weitere Auswahl von $s_m^* = 3$ Merkmalen, Aggregation auf $s_d = 1$ Merkmal für alle Hierarchiestufen,
- KO: siehe TN,
- HK: Vorauswahl von $s_m = 8$ Merkmalen, wenn möglich Reduktion auf $s_d = 2$ Merkmale in jeder Hierarchiestufe und
- HKD: Startanzahl Merkmale $s_{m,\min} = 7$, Aggregation auf $s_d = 3$ Merkmale, Anzahl Durchläufe $m_{\max} = 1$, Mindestkonfidenz $q_c = 0.99$, Mindesttupelzahl $n_c = 10$.

Jeder Lerndatensatz wird 10-fach crossvalidiert (50 Wiederholungen).

Nr.	k	n_j	DI	Fuzzy	KNN	TN	KO	HK	HKD
A	6	15	4.4 ± 1.1	7.0 ± 2.8	3.6 ± 1.5	3.7 ± 1.1	3.5 ± 0.9	2.6 ± 1.1	4.7 ± 1.2
B	5	15	9.0 ± 2.4	5.7 ± 1.6	7.8 ± 2.7	5.9 ± 2.1	6.1 ± 1.8	7.3 ± 1.8	6.0 ± 0.0
C	8	10	13.8 ± 2.9	10.6 ± 2.6	9.7 ± 3.0	5.6 ± 1.7	6.0 ± 1.7	7.1 ± 2.2	9.7 ± 2.9
D	8	15	5.1 ± 1.7	10.6 ± 2.1	3.0 ± 1.2	2.8 ± 0.9	3.0 ± 1.0	2.1 ± 1.1	6.7 ± 1.9
E	7	15	13.3 ± 2.1	11.3 ± 2.7	11.3 ± 1.9	9.9 ± 1.5	9.7 ± 1.5	12.8 ± 2.7	13.6 ± 2.5
F	5	15	6.1 ± 1.4	8.5 ± 3.8	4.5 ± 1.8	4.7 ± 1.1	5.9 ± 0.9	6.1 ± 1.8	6.8 ± 1.3
G	5	10	8.9 ± 1.3	6.4 ± 2.3	5.7 ± 1.8	5.7 ± 1.0	5.9 ± 1.1	6.4 ± 1.5	8.9 ± 1.3
H	5	10	8.5 ± 1.6	15.6 ± 5.4	12.2 ± 2.8	7.8 ± 1.2	7.3 ± 1.2	4.7 ± 2.2	8.7 ± 2.3
I	5	10	0.2 ± 0.7	11.5 ± 5.1	0.7 ± 1.1	0.4 ± 0.8	0.5 ± 0.9	0.7 ± 1.3	0.2 ± 0.7
J	9	15	6.6 ± 1.7	7.1 ± 2.2	1.5 ± 1.2	1.3 ± 0.5	1.2 ± 0.6	3.3 ± 1.7	7.2 ± 1.8
Mittelwert			7.6	9.4	6.0	4.8	4.9	5.3	7.3

Tabelle 2: Klassifikationsfehler bei 10-facher Crossvalidierung, n_j : Anzahl Datentupel pro Klasse, Merkmalsanzahl: $s = 108$

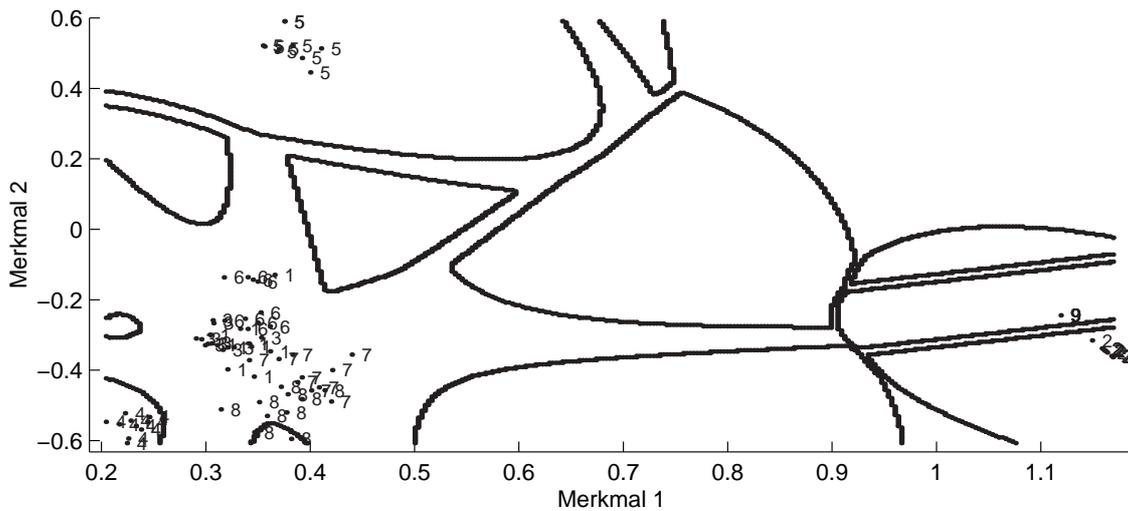


Bild 7: Erster Schritt der hierarchischen Extraktion von Datentupeln (Datensatz J)

Es ist zu erkennen, dass die Mehrschrittverfahren im Schnitt besser abschneiden als die Einschnittverfahren. Die besten Ergebnisse liefern hierbei die Turnier-Klassifikatoren mit Turnier- bzw. KO-Kriterium. Mit steigender Anzahl an Realisierungen pro Klasse und mit steigender Klassenanzahl bauen die Mehrschrittverfahren ihren Vorsprung aus.

Das vorgestellte hierarchische Verfahren zur Extraktion von Datentupeln liefert vergleichsweise schlechte Ergebnisse. Bild 7 zeigt das Verfahren anhand des Datensatzes J. Die schlechten Ergebnisse sind durch Abweichungen von der vorausgesetzten Wahrscheinlichkeitsdichte zu erklären, die durch das Abspalten von Datentupeln entsteht. Im Extremfall entstehen in höheren Hierarchieebenen ringförmige Wahrscheinlichkeitsdichten, wenn nur die Werte nahe dem Mittelwertvektor der Klasse sicher erkannt werden. Da die Diskriminanzanalyse und AML-Klassifikatoren Normalverteilungen voraussetzen, entstehen Fehlklassifikationen.

Ein zweiter Grund für das Scheitern ist die hohe Anzahl zu schätzender Parameter, robuste Schätzungen sind auf den vorhandenen Daten nicht möglich. Diese Fehler äußern sich besonders in den Gebieten im Merkmalsraum, die nicht durch Lerndaten beschrieben werden, in diesen Bereichen ergeben sich fragwürdige Klassenzuordnungen (siehe Bild 7). So werden zwar gute Ergebnisse über Lerndaten erhalten, über Testdaten jedoch ergeben sich mitunter hohe Klassifikationsfehler.

Sowohl eine große Anzahl an Hierarchiestufen m_{\max} als auch eine zu geringe Anzahl an Merkmalen in der ersten Hierarchiestufe $s_{m,\min}$ führen zu schlechten Approximationen und erhöhen somit den Klassifikationsfehler. Über den vorliegenden Daten konnten die besten Ergebnisse mit einer Beschränkung auf zwei Hierarchiestufen erreicht werden.

7 Zusammenfassung und Ausblick

Dieser Artikel stellt hierarchische Klassifikationsverfahren zum Lösen von Mehrklassenproblemen auf Basis von statistischen Klassifikatoren dar und evaluiert diese anhand von zehn Datensätzen. Es werden drei Verfahren vorgestellt, die hierarchische Extraktion von Klassen, die Dekomposition in Dichotomien durch Turnier- oder KO-Klassifikation und die hierarchische Extraktion von Datentupeln. Für letztere wird ein detaillierter Algorithmus vorgestellt. Zwar ist bekannt, dass sich ein Ranking verschiedener Klassifikatoren bei unterschiedlichen Problemstellungen stark unterscheidet (siehe z.B. STATLOG-Projekt [22]). Dennoch weisen alle vorgestellten Verfahren typische Besonderheiten auf, die je nach Problemstellung vorteilhaft oder nachteilig sind. Dieser Artikel leistet einen Beitrag zur Diskussion dieser Besonderheiten anhand typischer Effekte, die auch bei anderen Problemstellungen auftreten können.

Anhand von Patientendaten zur Steuerung einer Funktionsprothese zeigt sich, dass die hierarchischen Verfahren bessere Ergebnisse liefern als herkömmliche Einschrittverfahren. Unter den hierarchischen Verfahren schneidet die hierarchische Extraktion von Datentupeln am schlechtesten ab. Dies lässt sich darauf zurückführen, dass einerseits zu wenig Datensätze zur robusten Schätzung der Parameter vorhanden sind, andererseits zerstört die Extraktion von Datentupeln die Normalverteilung und verschlechtert damit die Klassifikationsergebnisse.

Verbessern lässt sich das Verfahren

- durch Einsatz von Entscheidungsregeln, deren Parameter robust geschätzt werden können (z. B. lineare Diskriminanzfunktionen, neuronale Netze, etc.),
- durch modifizierte Verfahren zur Merkmalsaggregation, die die Schätzung der Parameter verbessern, und
- durch eine Erweiterung der Lerndatensätze.

Danksagung

Die Autoren danken an dieser Stelle Christian Pylatiuk und Bernd Voelkel für ihre Hilfe in der Akquirierung von Patientendaten sowie Lutz Gröll für viele fruchtbare Diskussionen.

Literatur

- [1] Tatsuoka, M. M.: *Multivariate Analysis*. New York: Macmillan. 1988.
- [2] Mayoraz, E.; Moreira, M.: On the decomposition of polychotomies into dichotomies. In: *Proc. 14th International Conference on Machine Learning*, S. 291–226. Nashville, TN: Morgan-Kaufmann. 1997.
- [3] Fürnkranz, J.: Separate-and-Conquer rule learning. *Artificial Intelligence Review* 13(1) (1999), S. 1–46.

- [4] Reischl, M. and Gröll, L.; Mikut, R.: Optimierte Klassifikation für Mehrklassenprobleme am Beispiel der Bewegungssteuerung von Handprothesen. In: *Proc. 13. Workshop Fuzzy Systeme* (Mikut, R.; Reischl, M., Hg.), S. 124–143. Forschungszentrum Karlsruhe. 2003.
- [5] Reischl, M.; Gröll, L.; Mikut, R.: EMG-control of prostheses by switch signals: Extraction and classification of features. In: *IEEE International Conference on Systems, Man and Cybernetics*. The Hague. 2004.
- [6] Ahrens, H.; Läuter, J.: *Mehrdimensionale Varianzanalyse: Hypothesenprüfung, Dimensionserniedrigung, Diskrimination bei multivariaten Beobachtungen*. Berlin: Akademie-Verlag. 1974.
- [7] Mikut, R.; Peter, N.; Malberg, H.; Jäkel, J.; Gröll, L.; Bretthauer, G.; Abel, R.; Döderlein, L.; Rupp, R.; Schablowski, M.; Gerner, H.: Diagnoseunterstützung für die instrumentelle Ganganalyse (Projekt GANDI). *Bericht des Forschungszentrums Karlsruhe FZKA 6613* (2001).
- [8] Jain, A.; Duin, R.; Mao, J.: Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence* 22 (1) (2000), S. 4–37.
- [9] Schürmann, J.: *Pattern classification*. New York: John Wiley & Sons. 1996.
- [10] Vapnik, V.: *Statistical learning theory*. New York: John Wiley & Sons. 1998.
- [11] Masulli, F.; Valentini, G.: Comparing decomposition methods for classification. In: *Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies* (Howlett, R.; Jain, L., Hg.), S. 788–791. Piscataway, NJ. 2000.
- [12] Xu, L.; Krzyzak, A.; Suen, C.: Methods for combining multiple classifiers and their applications in handwritten character recognition. *IEEE Transactions on systems, man and cybernetics* 22 (1992), S. 418–435.
- [13] Hastie, T.; Tibshirani, R.: Classification by Pairwise Coupling. In: *Advances in Neural Information Processing Systems* (Jordan, M.; Kearns, M.; Solla, S., Hg.), Bd. 10. The MIT Press. 1998.
- [14] Kullback, S.; Leibler, R. A.: Information and sufficiency. *Ann. Math. Statist.* 22 (1951), S. 79–86.
- [15] N.N.: *Discriminant Analysis and Clustering*. Washington, D.C.: National Academy Press. 1988.
- [16] Anderson, E.: The irises of the gaspe peninsula. *Bulletin of the American Iris Society* 59 (1935), S. 2–5.
- [17] Hudgins, B.; Parker, P.; Scott, R.: A new strategy for multifunction myoelectric control. *IEEE Trans Biomed Eng.* 40 (1993), S. 82–94.
- [18] Nishikawa, D.: *Studies on electromyogram to motion classifier*. Dissertation, Graduate school of engineering, Hokkaido University, Sapporo, Japan. Dec, 2001.
- [19] Reischl, M.; Mikut, R.; Pylatiuk, C.; Schulz, S.; Beck, S.; Bretthauer, G.: Steuerungs- und Signalverarbeitungskonzepte für eine multifunktionale Handprothese. *Automatisierungstechnik (at)* 50 (6) (2002), S. 279–286.
- [20] Mikut, R.; Jäkel, J.; Gröll, L.: Interpretability Issues in Data-Based Learning of Fuzzy Systems. *accepted paper, Fuzzy Sets and Systems* (2004).
- [21] Voelkel, B.: *Auswertung von Patientendaten zur Evaluierung von EMG-Steuerungsstrategien*. Diplomarbeit, Forschungszentrum Karlsruhe. 2004.
- [22] Michie, D.; Spiegelhalter, D. J.; Taylor, C. C.: *Machine Learning, Neural and Statistical Classification*. London: Ellis Horwood. 1994.

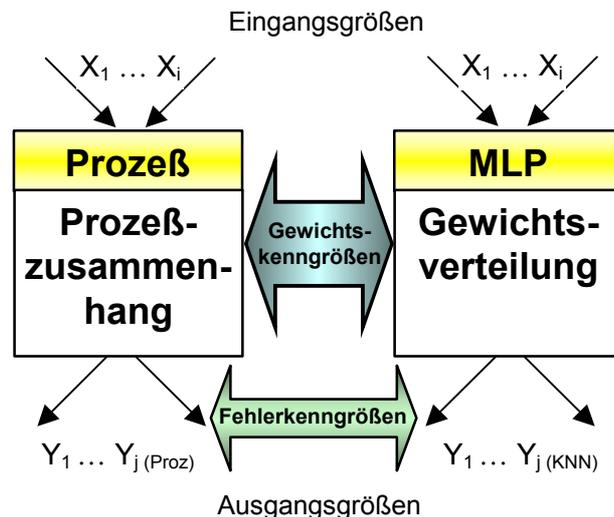
BEWERTUNG DER NETZGÜTE EINES MLP MIT HILFE DER GEWICHTSANALYSE – POTENTIAL UND GRENZEN

WOLFGANG KÄSTNER, TOM FÖRSTER, RAINER HAMPEL
Hochschule Zittau/Görlitz (FH)
Institut für Prozesstechnik, Prozessautomatisierung und Messtechnik (IPM)
02763 Zittau, Theodor-Körner-Allee 16

Abstrakt

Der Beitrag widmet sich dem Einsatz von Multilayer Perzeptren (MLP) als sogenannte Black-Box-Modelle zur exakten Modellierung nichtlinearer Prozeßzusammenhänge. Dargestellt werden ausgehend von [1, 2] die Ergebnisse der weiterführenden Untersuchungen zur nachfolgend beschriebenen Problematik.

Zur Bewertung der Netzgüte des MLP werden, über die bekannten Bewertungskriterien in Form von Fehlerkenngrößen hinaus, neue Kenngrößen definiert, die aus der sich einstellenden Gewichtsverteilung des MLP generiert werden.



Bewertung der Güte des MLP auf der Grundlage von Fehlerkenngrößen und Gewichtskenngrößen

Im Ergebnis dieser Gewichtsanalyse werden folgende Kenngrößen (Gewichtskenngrößen) definiert und interpretiert:

- *Gewichtsfaktor*
- *Korrelationsgewicht*
- *Gewichtsmaß*

Weiterhin wird ein Bezug zu korrespondierenden statistischen Kenngrößen wie:

- *Korrelationskoeffizient*
- *Bestimmtheitsmaß*

des nachzubildenden Prozeßzusammenhanges hergestellt (Datenanalyse).

Sowohl bei der Gewichtsanalyse als auch bei der Datenanalyse charakterisieren die genannten Kenngrößen die linearen Zusammenhänge zwischen den Eingangs- und Ausgangsgrößen.

Anhand des Vergleichs der Ergebnisse von Gewichtsanalyse und Datenanalyse wird geprüft, inwieweit sich der den Prozeßdaten innewohnende lineare Zusammenhang in den Gewichten des MLP abgebildet

hat. Damit wird ein zusätzliches Potential für die Bewertung der Netzgüte des MLP erschlossen. Die Methodik wird an geeigneten Beispielen demonstriert.

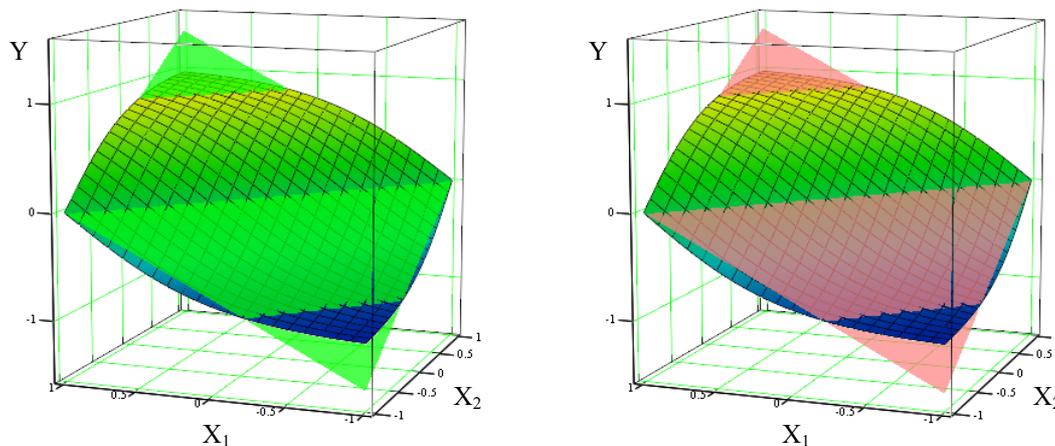
Die nachfolgenden Abbildungen veranschaulichen die Ergebnisse für ein einfaches Beispiel.

Das linke Bild stellt vergleichend gegenüber:

- nichtlineares Kennfeld für den Prozeßzusammenhang zwischen den Eingangsgrößen X_1 , X_2 und der Ausgangsgröße Y als Ausgangspunkt für das Training des MLP
- lineares Ersatzkennfeld, generiert mit Hilfe der statistischen Kenngröße *Korrelationskoeffizient* auf der Grundlage der Prozeßdaten (Ergebnis der Datenanalyse)

Das rechte Bild stellt vergleichend gegenüber:

- nichtlineares Kennfeld für den Prozeßzusammenhang zwischen den Eingangsgrößen X_1 , X_2 und der Ausgangsgröße Y reproduziert mit Hilfe des MLP
- lineares Ersatzkennfeld, generiert mit Hilfe der Gewichtskenngröße *Korrelationsgewicht* auf der Grundlage der Einzelgewichte des MLP (Ergebnis der Gewichtsanalyse)



Nichtlineares Kennfeld des Prozeßzusammenhanges und lineares Ersatzkennfeld

Darüber hinaus werden anhand von Beispielen die Auswirkungen unvollständiger Trainingsdaten auf die Bewertungsmöglichkeiten der Netzgüte mit Hilfe der Fehlerkenngrößen und Gewichtskenngrößen aufgezeigt und diskutiert.

Die Veranschaulichung der Problematik erfordert die Beschränkung auf Beispielprozesse, deren Prozeßzusammenhang im dreidimensionalen Zustandsraum (zwei Eingangsgrößen, eine Ausgangsgröße) darstellbar ist. Diskutiert werden jedoch auch Problemstellungen, die sich bei der Anwendung der Methodik auf höherdimensionale Prozeßzusammenhänge ergeben.

Die im Beitrag präsentierten Ergebnisse lassen die Schlußfolgerung zu, daß sich der den Prozeßdaten innewohnende Zusammenhang zwischen den Zustandsgrößen in den Gewichten des MLP abbildet. Die aus der Gewichtsanalyse abgeleiteten Kenngrößen werden als zusätzliche Bewertungskenngrößen für die Netzgüte eines MLP genutzt. Diese im Rahmen eines BMWA geförderten Forschungsprojektes (Förderkennzeichen 150 1248) erzielten Ergebnisse leisten einen Beitrag dafür, die als Black-Box-Modelle bezeichneten MLP transparenter zu gestalten. Dies gewinnt an Bedeutung für Anwendungen dieser Modellklasse zur Nachbildung sicherheitsrelevanter Prozeßzusammenhänge.

[1] Boger, Z: *Applications of ANN in Process Control Systems*. Vortrag im Rahmen des Seminars „Artificial Intelligence in Energetics“, Prag; 30.05.2000

[2] Kästner, W., u.a.: *Zur Problematik der Gewichtsverteilung in Künstlichen Neuronalen Netzen*. 12. GMA-GI-Workshop Fuzzy Systeme, Bommerholz, 14./15.11.2002