

Diplomarbeit im Fach Mathematik

Untersuchung von Klassifikationsverfahren im
Rahmen eines selbstadaptierenden regelbasierten
Systems

von

ANDREAS HEINEMANN

Prüfer:

Prof. Dr.rer.nat. D. Müller
Institut für Informatik

Prof. Dr.-Ing. F. Sieker
Institut für Wasserwirtschaft

Universität Hannover

JULI 1992

Diplomaufgabe

für cand. math. Andreas Heinemann Matr. Nr. 1183243

” Untersuchung von Klassifikationsverfahren im Rahmen eines selbstadaptierenden regelbasierten Systems”

On-line Überwachung und Steuerung eines Entwässerungssystems gewinnt in den letzten Jahren zunehmend an Bedeutung. Damit wird unter anderem während eines Niederschlags-Abfluß-Ereignisses versucht, Schwachstellen im Kanalnetz möglichst früh zu erkennen, entsprechende Gegenmaßnahmen zu bestimmen und anhand von Steuerungsorganen (z.B. Pumpen, Schieber, Wehre) umzusetzen.

Die Bestimmung der zweckmäßigen Steuerungs-Entscheidungen stellt einen wesentlichen Punkt dar. Optimierungsalgorithmen oder regelbasierte Systeme sind dabei anwendbar. Letztere sind deswegen einem Fachmann leichter verständlich, weil die sogenannte Wissensbasis eines Expertensystems in ihrer Struktur der menschlichen Denkweise besser entspricht; sie liegt in Form von Diagnose- bzw. Steuerungsregeln (logischen Assertionen) vor. Allerdings setzt die Herstellung jener Wissensbasis die Existenz eines Fachmanns (Experten) voraus, der eine optimierte Regelmenge vorschlägt. In den meisten Anwendungsfällen fehlen leider die Experten bzw. sie haben Schwierigkeiten, ihr Wissen vollständig und konsistent zu formulieren.

Der Bereich der Steuerung von Entwässerungssystemen ist in dieser Hinsicht keine Ausnahme. Es gibt nur wenige Menschen, die das nötige Steuerungs-Wissen gesammelt haben. Dementsprechend wird eine verbreitete Anwendung von regelbasierten Systemen erheblich erschwert. Einen gangbaren Ausweg bilden die jüngsten Entwicklungen von genauen (hydrodynamischen) Simulationsmodellen sowie Untersuchungen über automatisierte Lernalgorithmen. Im Institut für Wasserwirtschaft der Universität Hannover wurde ein Simulationsprogramm entwickelt, in dem eine hydrodynamische Simulation eines Entwässerungssystems mit einer Expertensystem-Shell gekoppelt wurde. Zudem wurde ein Lernalgorithmus implementiert, der aufgrund von Metaregeln eine Bewertung und Modifikation der vorhandenen Regelbasis durchführt.

Ein wesentlicher Bestandteil des Lernprozesses liegt in der Erfassung und Verarbeitung von Situationen, in denen die existierenden Steuerregeln unzulänglich waren, und der dafür verantwortlichen Zustandsfaktoren. Mathematisch entspricht dieses Problem einem Klassifikationsproblem.

Herr cand. math. Heinemann erhält die Aufgabe, Klassifikationsverfahren zu untersuchen, einen weiteren Lernalgorithmus im vorhandenen regelbasierten System zu implementieren und die Leistungsfähigkeit des neuen Verfahrens durch Simulationen zu prüfen.

Im einzelnen umfaßt die Aufgabe folgende Punkte:

- Theoretischer Überblick über Klassifikationsverfahren
- Implementierung eines geeigneten (oder mehrerer) Klassifikationsverfahrens in einem Lernprozeß des vorhandenen regelbasierten Systems
- Entwurf eines fiktiven Kanalnetzes und einer Menge von Meta-Regeln zur Optimierung einer Steuerung dieses Netzes
- Simulation mehrerer Niederschlagsereignisse und Durchführung des Lernens sowohl mit dem vorhandenen als auch mit dem neuen Lernverfahren
- Vergleich der Ergebnisse der Lernprozesse

Jedes Ergebnis soll nachvollziehbar sein. Dementsprechend sind die durchgeführten Arbeiten ausführlich darzustellen und nach Möglichkeit durch Abbildungen, Diagramme und Graphiken zu dokumentieren.

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig angefertigt und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Obershagen, Juli 1992

Inhaltsverzeichnis

I	Automatische Klassifikation	1
1	Ausgangsdaten	3
1.1	Datenrepräsentation	3
2	Beziehungen zwischen Objekten und Objektmengen	4
2.1	Ähnlichkeit und Distanz zwischen Objekten	4
2.2	Informationen über Objektmengen	12
3	Hierarchische Gruppierung	17
3.1	Definitionen	17
3.2	Agglomerative Verfahren	20
3.3	Divisive Verfahren	22
4	Nichtdisjunkte Gruppierung	23
4.1	Maximale Cliques	24
4.2	Unscharfe Klassifikation	25
5	Disjunkte Gruppierung	27
5.1	Gruppierung mittels Gütekriterien	27
5.2	Graphentheoretische Methoden	33
5.3	Analyse der Punktdichte	36
II	Das regelbasierte System	41
6	Kanalnetzsteuerung	41
7	Regelbasierte Systeme	42
8	Maschinelles Lernen	45

9	Das implementierte RBS	46
9.1	Die Wissensbasis	46
9.2	Auswahl und Anwendung der Regeln	48
9.3	Die Koppelung EXTRAN↔RBS	49
10	Verbesserung der Steuerung durch Lernen	51
10.1	Das bisherige Lernverfahren	52
10.2	Anmerkungen zum alten Lernverfahren	54
10.3	Das neue Lernverfahren	57
10.4	Anmerkungen zu den beiden Lernverfahren	62
11	Simulationen	71
12	Zusammenfassung	78

Teil I

Automatische Klassifikation

Häufig stellt sich das Problem, daß eine ungeordnete Menge von Objekten wegen ihrer Unübersichtlichkeit sehr schwer zu handhaben ist. Durch den Einsatz der elektronischen Datenverarbeitung entstehen außerdem in vielen Bereichen der Praxis Datenmengen, die allein durch ihre Größe sehr schwer zu analysieren sind. In einem solchen Fall wünscht man sich, diese Mengen in kleinere, zweckmäßige Teilmengen zu gliedern, also zu klassifizieren.

Die Ziele solcher Einteilungen können sehr unterschiedlich sein: Die Einteilung kann aus organisatorischen Gründen geschehen, auch wenn keine natürliche Struktur der Menge vorhanden ist, etwa bei Güteklassen im Handel oder Einkommensklassen. Sie kann der übersichtlichen Darstellung großer Datenmengen dienen, etwa Ordnung der Bücher einer Bücherei nach Sachgebieten, eingesetzt werden. Das Ziel einer Unterteilung der Objektmenge kann aber auch eine Reduktion der Informationen über die vorhandene Datenmenge sein, etwa um überflüssige Informationen zu beseitigen.

Unter den Begriff "automatische Klassifikation" (Cluster-Analyse oder numerische Taxonomie) werden Verfahren eingeordnet, die der Einteilung ungeordneter Mengen von Objekten in zweckmäßige oder der Struktur der Mengen entsprechende Klassen (Gruppen, Cluster) dienen.

Es gibt keine allgemeingültigen Aussagen darüber, was eigentlich eine Klasse ist. Häufig wählt man aber abhängig vom Ziel einer Klassifikation eine der folgenden Definitionen:

- Eine Klasse ist eine Menge von Objekten, die sich *ähnlich* sind. Objekte verschiedener Klassen sind sich nicht ähnlich.
- Eine Klasse ist eine Teilmenge einer Menge von Objekten, in der die Distanz zwischen zwei Objekten der Teilmenge geringer ist als die Distanz zwischen einem beliebigen Objekt der Teilmenge und einem Objekt der Objektmenge, welches nicht in der Teilmenge enthalten ist.
- Eine Klasse ist eine zusammenhängende Region in einem (mehrdimensionalen) Raum, die gegenüber ihrer Umgebung eine relativ hohe Punktdichte aufweist.

Diese Definitionen zeigen, daß die Klassifikationsverfahren eine gewisse Verwandtschaft zwischen den Objekten analysieren und anhand dieser die Ob-

jekte klassifizieren. Für eine Klassifikation einer Objektmenge wird vorausgesetzt, daß in der Objektmenge eine Verwandtschaftsstruktur vorhanden ist und sich diese messen läßt.

Die erzeugten Gruppierungen unterscheidet man nach folgenden Kriterien:

Eine Gruppierung heißt *disjunkt*, wenn jedes klassifizierte Objekt in höchstens einer Klasse enthalten ist, d.h. die Klassen sind paarweise disjunkt. Dementsprechend spricht man von einer *nichtdisjunkten* Gruppierung, wenn eine Überschneidung der Klassen zugelassen wird.

Eine Gruppierung heißt *exhaustiv*, wenn jedes Element der vorgegebenen Objektmenge in einer Klasse enthalten ist. Gibt es Objekte, die nicht klassifiziert wurden, dann spricht man von einer *nicht exhaustiven* Gruppierung. Eine exhaustive, disjunkte Gruppierung wird auch als *Partition* bezeichnet.

Ein *hierarchisches* Klassifikationsverfahren erzeugt eine Folge von Gruppierungen. Dabei entsteht eine Gruppierung aus ihrer vorhergehenden durch Spaltung oder Verschmelzung von Klassen. Die einzelnen Klassen sind daher stammbaumartig angeordnet. Wenn z.B. eine Menge von naturwissenschaftlichen Büchern klassifiziert werden soll, kann man die Bücher nach den Fachgebieten Chemie, Biologie, usw. sortieren. Eventuell ist diese Unterteilung zu grob und man unterteilt die einzelnen Klassen noch weiter. Die Klasse der Bücher aus dem Bereich Chemie wird dann vielleicht in die Klassen "Organische Chemie" und "Anorganische Chemie" unterteilt.

Nun stellt sich die Frage, wie man von den Ausgangsdaten zu einer fertigen Gruppierung kommt. Zuerst werden die Ausgangsdaten untersucht, um festzustellen, welche Informationen über die Objekte für ein Klassifikationsverfahren zur Verfügung stehen und welche Informationen davon sinnvoll sind. Bei einer Klassifikation von Büchern in einer Bücherei nach Sachgebieten ist z.B. die Farbe des Einbandes eines Buches nicht gerade eine nützliche Information. Aufgrund dieser Informationen wählt man ein Maß, mit dem die Verwandtschaft von Objekten oder Objektmengen gemessen werden kann. Dann wählt man ein Klassifikationsverfahren, welches zu den zur Verfügung stehenden Informationen und dem Ziel der Klassifikation paßt und führt die Klassifizierung der Objekte durch.

Diese Schritte können bei falscher Wahl der Mittel zu einer künstlichen, der tatsächlichen Struktur der Objektmenge nicht entsprechenden Gruppierung führen. Leider gibt es kein allgemein gültiges Rezept, wie diese Schritte bei einer gegebenen Problemstellung am besten durchzuführen sind. Daher ist eine sorgfältige Wahl der Methode und eine Untersuchung der Ergebnisse hinsichtlich ihres praktischen Nutzens von großer Wichtigkeit.

1 Ausgangsdaten

1.1 Datenrepräsentation

Die Ausgangsdatenmenge wird in den meisten Fällen auf eine der beiden folgenden Arten dargestellt: Durch die *Datenmatrix* oder durch eine *Ähnlichkeits-* oder *Distanzmatrix*.

1.1.1 Die Datenmatrix

Sei $S = \{O_1, \dots, O_N\}$ (oder $S = \{1, \dots, N\}$) die Menge der Objekte, die klassifiziert werden sollen. Jedes Objekt werde durch bestimmte Merkmale M_1, \dots, M_p charakterisiert. Bei Patienten in Krankenhäusern werden z.B. Alter, Gewicht, Blutdruck, usw. erfaßt. Im folgenden wird vorausgesetzt, daß sich das Verhalten eines jeden Objektes O_k ($k = 1, \dots, N$) bzgl. seiner p Merkmale M_j ($j = 1, \dots, p$) durch eine reelle Zahl x_{kj} ausdrücken läßt.

Diese Beobachtungswerte werden in einer $N \times p$ Matrix $X = (x_{kj})$ angeordnet, der *Datenmatrix*.

	M_1	M_2	\dots	M_p
O_1	x_{11}	x_{12}	\dots	x_{1p}
O_2	x_{21}	x_{22}	\dots	x_{2p}
\vdots	\vdots	\vdots		\vdots
O_N	x_{N1}	x_{N2}	\dots	x_{Np}

Abb. 1: Die Datenmatrix

Das k -te Objekt O_k wird dabei durch den k -ten Zeilenvektor $x_k = (x_{k1}, \dots, x_{kp})$ beschrieben. Deshalb wird S auch als $S = \{x_1, \dots, x_N\}$ geschrieben.

1.1.2 Ähnlichkeits- und Distanzmatrix

Für eine Klassifikation einer Menge von Objekten wird ein Maß für die Verwandtschaft zweier Objekte benötigt. Dieser Ausdruck kann mit den Ausgangsdaten vorliegen, er kann aber auch aus der Datenmatrix berechnet werden. Diese Verwandtschaft ist entweder eine Ähnlichkeit oder eine Distanz. Sie wird in einer symmetrischen $N \times N$ Matrix angegeben. Je nach Art der Verwandtschaft spricht man von einer *Ähnlichkeits-* oder *Distanzmatrix*.

1.1.3 Datentypen

Objekte können durch sehr verschiedenartige Merkmale (oder Variablen, Eigenschaften) beschrieben werden, welche jeweils eine bestimmte Menge von Zuständen (oder Kategorien) annehmen können. Je nach ihrem Wertebereich werden drei Arten von Merkmalen unterschieden:

Ein Merkmal M_i heißt *quantitativ* (oder metrisch), wenn die zugehörigen x_{ki} ($k = 1, \dots, N$) alle Werte eines endlichen oder unendlichen Intervalls annehmen können (Temperaturen, Blutdruck, Größe, Gewicht, usw.). Die entscheidende Eigenschaft solcher Merkmale ist, daß die Differenzen verschiedener Werte vergleichbar sind: Bei Gewichtsangaben z.B. haben 10 kg Unterschied überall auf der Skala die gleiche Bedeutung.

Ein Merkmal M_i heißt *qualitativ* (oder nominal), wenn für die Zustände x_{ki} nur endl. viele verschiedene Werte oder Alternativen zur Verfügung stehen. Z.B.: Güteklassen, Farben. Diese Variablen können durch die ersten positiven ganzen Zahlen gekennzeichnet werden (z.B. können den Zuständen blau, rot, gelb die Zahlen Eins bis Drei zugeordnet werden).

Ein qualitatives Merkmal M_i heißt *binär*, wenn die zugehörigen x_{ki} nur zwei Werte annehmen können (z.B. an - aus, ja - nein, männlich - weiblich). Diese Werte werden allgemein durch die Zahlen Null und Eins gekennzeichnet. Ein Merkmal mit mehr als zwei Alternativen wird auch als *mehrstufig* bezeichnet. Sind bei einer qualitativen Variable die möglichen Zustände geordnet, so daß die Reihenfolge der Zustände eine Bedeutung besitzt, dann nennt man sie eine *ordinale* Variable. Ein Beispiel sind bei den Zuständen riesig, groß, klein, winzig. Ein Abstand zweier Zustände läßt sich aber im allgemeinen nicht messen.

2 Beziehungen zwischen Objekten und Objektmengen

In diesem Abschnitt werden Definitionen für Ähnlichkeits-, Distanz- und Homogenitätsmaße angegeben, verschiedene Maße für unterschiedliche Ausgangsdaten vorgestellt und in diesem Zusammenhang auftretende Fragestellungen behandelt.

2.1 Ähnlichkeit und Distanz zwischen Objekten

Die Ähnlichkeit zweier Objekte O_j, O_k wird normalerweise durch eine Zahl s_{jk} angegeben, die alle Werte des reellen Intervalls $[0;1]$ annehmen kann. Je größer der Wert s_{jk} ist, desto ähnlicher sind sich die zwei Objekte. An die

Maßzahlen s_{jk} ($1 \leq j, k \leq N$) werden folgende Forderungen gestellt:

$$\begin{aligned} s_{jk} &= s_{kj} \\ s_{jk} &\geq 0 \\ s_{jk} &\leq s_{kk} \\ s_{kk} &= 1 \end{aligned}$$

Die Matrix (s_{jk}) heißt *Ähnlichkeitsmatrix* und gibt die Ähnlichkeitsstruktur der Objektmenge wieder. Man spricht auch von einem *Ähnlichkeitsmaß* s auf der Objektmenge S .

Die Unähnlichkeit zweier Objekte O_j, O_k wird durch einen Zahlenwert d_{jk} gemessen. Je größer der Wert d_{jk} ist, desto unähnlicher sind sich die Objekte. Dabei wird i.a. nicht gefordert, daß d_{jk} nur Werte aus dem Intervall $[0;1]$ annehmen darf, sondern meist sind Werte aus dem Intervall $[0;\infty[$ erlaubt. Es soll im folgenden für die d_{jk} ($1 \leq j, k \leq N$) gelten:

$$\begin{aligned} d_{jk} &= d_{kj} \\ d_{jk} &\geq 0 \\ d_{jk} &\geq d_{kk} \\ d_{kk} &= 0 \end{aligned}$$

Die Matrix (d_{jk}) heißt *Distanzmatrix* der Objekte. Sie legt ein *Distanzmaß* d auf S fest. Die Definition der Distanz unterscheidet sich etwas von der in der Mathematik gebräuchlichen: Gilt $d_{jk} = 0$ für zwei Objekte O_j, O_k , so bedeutet das nicht, daß es sich um dasselbe Objekt handelt, sondern es sagt nur aus, daß die Merkmalsausprägungen beider Objekte übereinstimmen (z.B. zwei Patienten mit gleichem Gewicht und Blutdruck).

Gilt außerdem

$$d_{ik} \leq d_{ij} + d_{jk} \quad 1 \leq i, j, k \leq N$$

(Dreiecksungleichung), so spricht man von einem *metrischen* Distanzmaß d oder einer *Metrik*.

Bei Distanz- und Ähnlichkeitsmaßen sind außerdem noch Invarianzeigenschaften interessant. Sei T eine Transformation des Merkmalsraumes. Ein Ähnlichkeitsmaß s_{jk} heißt *invariant* bezgl. T , wenn für x_1, \dots, x_N gilt:

$$s_{jk} = s(x_j, x_k) = s(Tx_j, Tx_k) \quad 1 \leq j, k \leq N$$

Die am häufigsten geforderte Invarianzeigenschaft ist die *Skaleninvarianz*.

Ein Ähnlichkeitsmaß s heißt *skaleninvariant*, wenn für

$$T : (x_{j1}, \dots, x_{jp}) \rightarrow (a_1 x_{j1}, \dots, a_p x_{jp}) \quad j = 1, \dots, N$$

mit $a_k \in \mathbb{R}_+$ ($k = 1, \dots, p$) gilt:

$$s_{jk} = s(x_j, x_k) = s(Tx_j, Tx_k) \quad 1 \leq j, k \leq N$$

Ein Maß heißt *translationsinvariant*, wenn es bzgl. einer Abb.

$$T : x \rightarrow x + a$$

mit $a \in \mathbb{R}^p$ fest und $x \in \mathbb{R}^p$ invariant ist.

Ein Maß heißt invariant bzgl. orthogonaler linearer Transformationen, wenn es bzgl. einer Abb.

$$T : x \rightarrow Cx$$

invariant ist. C ist dabei eine orthogonale $p \times p$ Matrix, $x \in \mathbb{R}^p$.

2.1.1 Distanzmaße für quantitative Merkmale

Es wird vorausgesetzt, daß alle p Merkmale der N Objekte quantitativ sind, d.h., alle Werte x_{jk} der Datenmatrix sind reelle Zahlen. Aus der L_r -Norm eines p -dimensionalen Vektors $y = (y_1, \dots, y_p)$

$$\|y\|_r = \left(\sum_{k=1}^p |y_k|^r \right)^{1/r} \quad r \geq 1$$

wird das gebräuchlichste Distanzmaß, die L_r -Distanz (oder auch *Minkowski-Metrik*) hergeleitet:

$$d_{jk}^r = d_r(x_j, x_k) = \|x_j - x_k\|_r := \left(\sum_{i=1}^p |x_{ji} - x_{ki}|^r \right)^{1/r} \quad r \geq 1$$

Bekannte Spezialfälle der L_r -Distanzen sind der *euklidische Abstand* ($r = 2$)

$$d_{jk}^2 = d_2(x_j, x_k) = \|x_j - x_k\|_2 := \left(\sum_{i=1}^p (x_{ji} - x_{ki})^2 \right)^{1/2}$$

und die *City-Block Distanz* oder *Manhattan Distanz* ($r = 1$).

$$d_{jk}^1 = d_1(x_j, x_k) = \|x_j - x_k\|_1 := \sum_{i=1}^p |x_{ji} - x_{ki}|$$

Für $r > 1$ haben die großen Differenzen der Komponenten ein stärkeres Gewicht als kleine. Die L_r -Distanzen sind nicht skaleninvariant. Sie sind

translationsinvariant und der euklidische Abstand ist zusätzlich noch invariant bzgl. linearer orthogonaler Transformationen.

Eine weiteres wichtiges Distanzmaß ist die *Mahalanobis-Distanz*. Sei $\hat{\Sigma}$ die $p \times p$ Kovarianzmatrix der p Merkmale.

$$\hat{\Sigma} := \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^T (x_i - \bar{x})$$

mit

$$\bar{x} := \frac{1}{N} \sum_{k=1}^N x_k$$

Die Mahalanobis-Distanz wird definiert durch

$$d_{jk} := \sqrt{(x_j - x_k)^T \hat{\Sigma}^{-1} (x_j - x_k)}$$

Im Gegensatz zu den L_r -Distanzen werden hier mögliche Korrelationen zwischen den Merkmalen berücksichtigt. Ist die Korrelation zwischen den Merkmalen 0, so ist die Mahalanobis-Distanz äquivalent zum euklidischen Abstand. Da die Mahalanobis-Distanz die Kovarianzmatrix benutzt, hängt bei diesem Distanzmaß die Distanz zwischen zwei Punkten von der gesamten Punktmenge $\{x_1, \dots, x_N\}$ ab. Das Maß ist sowohl skaleninvariant, translationsinvariant als auch invariant bzgl. orthogonaler linearer Transformationen.

2.1.2 Distanzmaße für qualitative Merkmale

Es wird vorausgesetzt, daß alle p Merkmale qualitativ sind. Zuerst werden Maße angegeben für binäre Merkmale (Die Datenmatrix enthält also nur die Ziffern "0" und "1").

Die Ähnlichkeit zweier Objekte O_j, O_k , die nur binäre Merkmale besitzen, wird häufig durch Auswertung einer Kontingenztafel für diese Objekte berechnet, in der die Übereinstimmungen und Nichtübereinstimmungen der Merkmalsausprägungen dargestellt werden:

		O_k	
		0	1
O_j	0	a_{jk}	b_{jk}
	1	c_{jk}	e_{jk}

a_{jk} gibt die Anzahl der Merkmale an, die bei O_j, O_k die Alternative "0" besitzen. b_{jk} gibt die Anzahl der Merkmale an, bei denen O_j die Alternative "0" und O_k die Alternative "1" besitzt. c_{jk} und e_{jk} werden analog gebildet.

Der *M-Koeffizient*, der relative Anteil der übereinstimmenden Komponenten von x_j und x_k

$$s_{jk} := \frac{a_{jk} + e_{jk}}{p}$$

, berücksichtigt in gleicher Weise die Übereinstimmungen bei dem Wert "1" und die bei dem Wert "0". Der *Jaccard* -Koeffizient

$$s_{jk} := \frac{a_{jk}}{a_{jk} + b_{jk} + c_{jk}}$$

berücksichtigt im Gegensatz dazu nicht die Übereinstimmungen bei der Alternative "0". Dies ist z.B. dann nützlich, wenn die "1" wichtiger als die "0" ist. In diesem Fall werden die Übereinstimmungen bei der "1" stärker gewichtet als die bei der "0".

Im folgenden seien alle Merkmale in der Datenmatrix qualitativ, die Anzahl der Alternativen des i -ten Merkmals sei $m_i \geq 2$. $Z_0^i, Z_1^i, \dots, Z_{m_i-1}^i$ seien die möglichen Alternativen des i -ten Merkmals. Die Alternativen von M_i werden in der Datenmatrix dann durch die Zeichen "0", "1", ..., " $m_i - 1$ " angegeben.

Auch bei mehrstufigen Merkmalen kann die Ähnlichkeit zweier Objekte aus der Anzahl der übereinstimmenden (und der Anzahl der nicht übereinstimmenden) Merkmalsausprägungen berechnet werden. Dabei ist evtl. eine Ordnung der Zustände zu beachten. Diese kann bedeuten, daß zwei Zustände umso unähnlicher sind, je weiter sie in der Anordnung auseinanderliegen. Dies ist z.B. der Fall bei Größenangaben (groß-mittel-klein).

Sei u_{jk} die Anzahl der übereinstimmenden Komponenten von x_j, x_k und p die Anzahl der Merkmale. Dann läßt sich die Ähnlichkeit der Objekte O_j, O_k durch das Ähnlichkeitsmaß

$$s_{jk} := \frac{u_{jk}}{p}$$

berechnen. Es ist eine Verallgemeinerung des M-Koeffizienten für binäre Merkmale.

Bei geordneten Merkmalen werden häufig Ähnlichkeits- und Distanzmaße verwendet, die auch bei quantitativen Merkmalen benutzt werden. Dazu werden die Alternativen "0", "1", ..., " $m_i - 1$ " als Zahlen $0, 1, \dots, m_i - 1$ interpretiert.

2.1.3 Gemischte Merkmale

In der Praxis werden die zu klassifizierenden Objekte häufig durch qualitative *und* quantitative Merkmale beschrieben. Für diesen Fall gibt es mehrere Ansätze, die Ähnlichkeit oder Distanz solcher Objekte zu messen:

- Sind die Alternativen der qualitativen Merkmale geordnet oder die Merkmale binär, dann kann ein quantitatives Distanzmaß verwendet werden. Es können auch die quantitativen Daten diskretisiert werden. Dann wird ein Ähnlichkeitsmaß für qualitative Merkmale mit geordneten Alternativen verwendet.
- Sowohl für die qualitativen als auch für die quantitativen Merkmale wird eine Ähnlichkeit oder Distanz berechnet. Aus diesen beiden Werten wird dann die Ähnlichkeit der Objekte berechnet, etwa durch einen gewichteten Mittelwert.

2.1.4 Normierung der Daten

Das Messen der Ähnlichkeit oder Distanz von Objekten ist immer dann schwierig, wenn die Merkmale der Objekte sehr unterschiedliche Eigenschaften beschreiben oder verschiedene Maßeinheiten besitzen. Wird z.B. ein Merkmal in Zentimetern gemessen, ein anderes Merkmal in Metern und ein Drittes in Kilogramm, dann ist eine Vergleichbarkeit der Merkmale nicht ohne weiteres gewährleistet. Meistens wird in solchen Fällen die Datenmatrix $X = (x_{jk})$ vor der Durchführung der Klassifikation normiert. Zwei der gebräuchlichsten Methoden werden hier vorgestellt:

Seien alle p Merkmale der Objekte quantitativ.

- Für jedes Merkmal M_i werden die entsprechenden Werte x_{1i}, \dots, x_{Ni} der i -ten Spalte von X linear in das Intervall $[0;1]$ transformiert. Seien

$$z_i := \min_{1 \leq k \leq N} x_{ki}$$

das Minimum,

$$u_i := \max_{1 \leq k \leq N} x_{ki}$$

das Maximum aller Werte der i -ten Spalte von X . Dann gibt $r_i := z_i - u_i$ die Spannweite der Werte der i -ten Spalte an. Die normierten Werte erhält man durch

$$\tilde{x}_{ki} := \frac{x_{ki} - z_i}{r_i} \quad i = 1, \dots, p \quad k = 1, \dots, N$$

$\tilde{X} = (\tilde{x}_{ki})$ ist dann die normierte Datenmatrix.

- Alle Merkmale werden auf Mittelwert 0 und Varianz 1 normiert. Mit den Mittelwerten

$$\bar{x}_i := \frac{1}{N} \sum_{k=1}^N x_{ki} \quad i = 1, \dots, p$$

und den empirischen Standardabweichungen

$$\sigma_i^2 := \frac{1}{N} \sum_{k=1}^N (x_{ki} - \bar{x}_i)^2 \quad i = 1, \dots, p$$

erhält man die normierten Werte durch

$$\tilde{x}_{ki} := \frac{x_{ki} - \bar{x}_i}{\sigma_i} \quad i = 1, \dots, p \quad k = 1, \dots, N$$

Die Normierung kann aber auch negative Folgen haben, da sie die Distanzen zwischen den Punkten x_1, \dots, x_N verändert. Wenn durch die nicht normierten Daten eine gute Separation von Klassen gegeben ist, kann eine Normierung der Daten diese Distanzen zwischen den Klassen verringern (JAIN/DUBES).

2.1.5 Fehlende Daten

Bei einer praktischen Anwendung von Klassifikationsverfahren kann es geschehen, daß für einige Objekte der zu klassifizierenden Objektmenge nicht alle Eigenschaften bekannt sind (z.B. durch den Ausfall von Meßgeräten), d.h. einige Einträge der Datenmatrix sind unbekannt. In diesem Fall gibt es einige einfache Verfahren, um mit fehlenden Daten umzugehen:

- Alle Objekte mit fehlenden Daten werden aus der Objektmenge entfernt und die Klassifikation wird dann mit Objekten durchgeführt, deren Eigenschaften vollständig bekannt sind.
- Es wird die Umgebung eines Objektes O_i mit fehlenden Daten untersucht. Mit den Daten der dort liegenden Objekte werden die fehlenden Daten von O_i ergänzt (etwa durch Bildung von Mittelwerten).
- Für die Berechnung der Ähnlichkeit oder Distanz zweier Objekte wird ein additives, bzgl. der Anzahl seiner Summanden invariantes Maß gewählt, in dessen Summe nur Merkmale berücksichtigt werden, deren Ausprägungen für beide Objekte bekannt sind.

Aus der City-Block-Distanz ergibt sich dann z.B. das Distanzmaß

$$\bar{d}_{jk} := \frac{1}{\bar{p}} \left(\sum_{i \in I} |x_{ji} - x_{ki}| \right)$$

wobei $I = \{ i \mid i \in \{1, \dots, p\} \text{ und } x_{ji}, x_{ki} \text{ sind bekannt} \}$ und $\bar{p} = |I| \leq p$ die Anzahl der Merkmale ist, die für beide Objekte bekannt sind.

2.1.6 Lineare Projektion

Seien die Objekte O_1, \dots, O_N mit p quantitativen Merkmalen gegeben und (x_{jk}) die zugehörige $N \times p$ Datenmatrix. Durch Anwendung von Projektionsverfahren soll die Dimension der p -dimensionalen Beobachtungsvektoren x_1, \dots, x_N verringert werden auf eine Dimension q mit $p > q$. Es gibt mehrere Gründe, warum eine solche Projektion im Zusammenhang mit Klassifikationsverfahren sinnvoll sein kann. Zum einen erlaubt eine Reduktion der Dimension auf $q = 2$ eine visuelle Erkennung von Punktgruppen, was zur Beurteilung einer von einem Klassifikationsverfahren erzeugten Gruppierung nützlich sein kann. Zum anderen kann eine Reduktion der Dimension überflüssige Informationen (etwa bei stark korrelierten Merkmalen) eliminieren. Die gebräuchlichste Methode ist das *Hauptkomponentenverfahren*.

Dabei werden die Beobachtungsvektoren linear auf eine q -dimensionale Hyperebene H des \mathbb{R}^p abgebildet. Dies geschieht durch die Abbildung

$$y_i = u + Qx_i \quad i = 1, \dots, N$$

x_i ist dabei ein Beobachtungsvektor, u ein Vektor des \mathbb{R}^p , Q eine $p \times p$ Matrix mit dem Rang s . und y_i der zu x_i gehörende Bildvektor.

Bei der Projektion der Punkte soll natürlich die Struktur der Objektmenge möglichst gut erhalten bleiben. Deshalb wird die Hyperebene so gewählt, daß der mittlere quadratische Abstand

$$\frac{1}{N} \sum_{i=1}^N |x_i - (u + Qx_i)|^2 = \text{Min} !$$

zwischen den Punkten x_i und ihren Bildpunkten y_i minimal ist.

Seien $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$ die Eigenwerte der empirischen Kovarianzmatrix $\hat{\Sigma}$ und v_1, \dots, v_p die zugehörigen (orthonormalen) Eigenvektoren.

Die gesuchte Hyperebene wird dann definiert durch

$$Q := \sum_{i=1}^q v_i v_i^T$$

und

$$u := (E - Q)\bar{x}$$

Durch die Abbildung

$$y(x) = u + Qx$$

werden die Punkte $x_1, \dots, x_N \in \mathbb{R}^p$ orthogonal auf die Hyperebene projiziert, die den Punkt \bar{x} enthält und von den Vektoren v_1, \dots, v_q aufgespannt wird.

Wählt man dann als neuen Ursprung im \mathbb{R}^p den Punkt \bar{x} und als Koordinatenrichtungen die v_1, \dots, v_p ein, so verschwinden für die y_i die letzten $(p - q)$ Komponenten. Die N Objekte können so als Punkte des \mathbb{R}^q dargestellt werden.

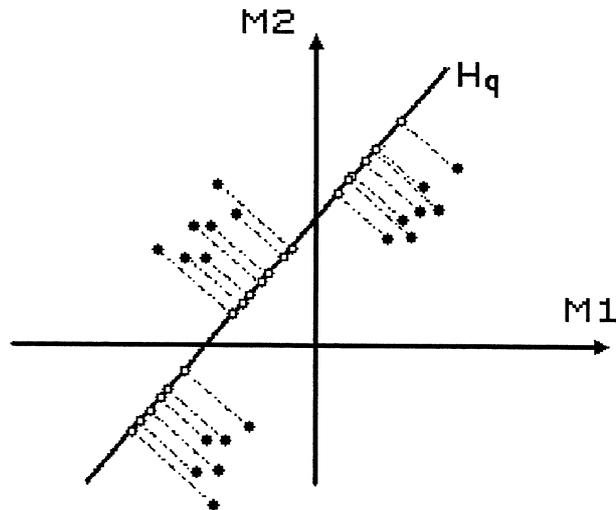


Abb. 2: Projektion auf Hyperebene mit $p = 2, q = 1$

2.2 Informationen über Objektmengen

Klassifikationsverfahren erzeugen in einer Objektmenge Klassen. An diese Klassen werden verschiedene Forderungen gestellt. Es kann Wert darauf gelegt werden, daß die Objekte einer Klasse sich besonders ähnlich sind (homogene Klassen) oder daß je zwei Klassen einer Gruppierung sich gut unterscheiden lassen. Bei der Konstruktion oder der Bewertung einer Klassifikation kann es daher notwendig sein, die Homogenität einer Objektmenge

oder die Ähnlichkeit/Distanz zweier Objektmengen messen zu können.

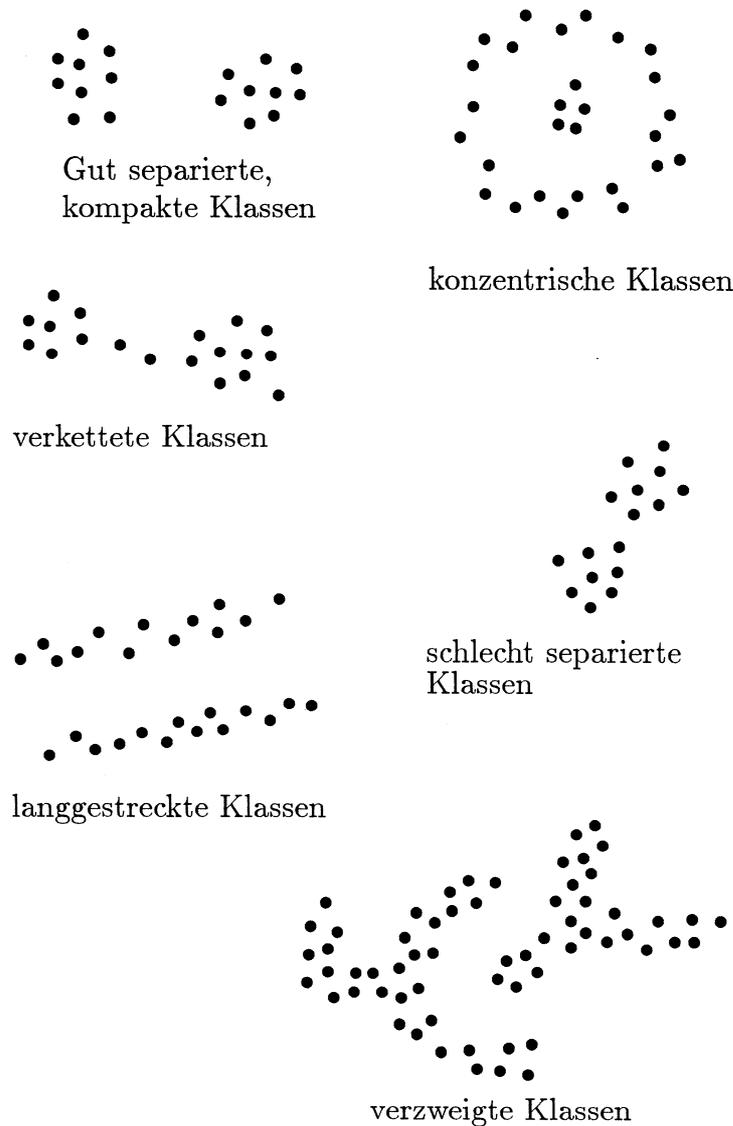


Abb. 3: Beispiele für Objektmengen mit Klassen unterschiedlicher Form

2.2.1 Repräsentative Objekte einer Menge

Bei einigen Homogenitätsmaßen (und Heterogenitätsmaße), aber auch bei einigen Gruppierungsverfahren (z.B. rekursiver Aufbau um Kerne) werden Klassen durch einzelne Objekte repräsentiert, die diese Klassen charakterisieren. Einige Möglichkeiten, solche typischen Objekte einer Menge zu bestimmen, werden jetzt vorgestellt:

Zentrale Punkte Für die Objektmenge $S=(x_1, \dots, x_N)$ mit quantitativen Merkmalen sei die Datenmatrix $X=(x_{kj})$ bekannt. Wenn eine Menge $A \subset S$ (interpretiert als Menge von Punkten des \mathbb{R}^p) im \mathbb{R}^p eine kugelförmige oder ovale Anordnung¹ aufweist, liegt die Idee nahe, die Menge durch ihren Mittelpunkt zu repräsentieren:

$$\bar{x}_A := \frac{1}{|A|} \sum_{k \in A} x_k$$

Bei dieser Methode ist offensichtlich, daß \bar{x}_A nicht Element von A sein muß.

Wenn für S eine Distanzmatrix $D = (d_{kj})$ bekannt ist, kann in der Menge A das Objekt $O_k \in A$ bestimmt und als zentrales Objekt bezeichnet werden, für das der Wert $\sum_{j \in A} d_{kj}$ minimal ist. Wenn eine Ähnlichkeitsmatrix (s_{kj}) bekannt ist, kann man analog das Objekt $O_k \in A$ wählen, für das der Wert $\sum_{j \in A} s_{kj}$ maximal ist.

Kernpunkte Besitzt die zu untersuchende Menge $A \subseteq S$ (interpretiert als Menge von Punkten des \mathbb{R}^p) keine kompakte Form, dann ist ein zentraler Punkt von A nicht unbedingt typisch für diese Klasse. Als typisches Objekt kann dann das Element von A angesehen werden, welches in dem Bereich mit der größten Punktdichte liegt. Es wird vorausgesetzt, daß für die Objektmenge S eine Distanzmatrix (d_{kj}) bekannt ist. Sei $d^* > 0$ eine Distanzschranke und sei $t(O_k)$ die Anzahl von Punkten $O_j \in A$, für die gilt: $d_{kj} \leq d^*$. Als Kernpunkt von A wird dann das i.a. nicht eindeutig bestimmte Objekt $O_k (= O_k(d^*))$ bezeichnet, für das $t(O_k)$ maximal ist.

2.2.2 Homogenität einer Objektmenge

Die Homogenität einer Objektmenge ist ein Maß für die mittlere Ähnlichkeit zwischen den Objekten dieser Menge. Sei $A \subseteq S$, dann wird diese mittlere Ähnlichkeit von A angegeben durch einen Wert $k(A) \geq 0$, der als *Homogenität* der Menge A bezeichnet wird. Je größer der Wert $k(A)$ ist, desto homogener ist die Menge A . Analog dazu gibt der Wert $g(A) \geq 0$ die *Heterogenität (Inhomogenität)* der Menge A an. Je größer $g(A)$ ist, desto unähnlicher sind sich die Objekte von A im Mittel.

Im folgenden werden verschiedene Homogenitäts- und Heterogenitätsmaße für unterschiedliche Ausgangsdaten vorgestellt. Dabei ist $n = n_A$ stets die Anzahl der Objekte in A .

Sei für die Objektmenge $S = \{O_1, \dots, O_N\}$ eine Distanzmatrix bekannt. Die Homogenität einer Menge $A \subseteq S$ läßt sich dann messen durch

¹Eine solche Klasse wird als *kompakt* bezeichnet. Der Begriff "kompakt" bezieht sich dabei auf die Form der Punktmenge, nicht aber auf die mathematische Definition der Kompaktheit einer Menge.

1.

$$g(A) := \frac{1}{n(n-1)} \sum_{k \in A} \sum_{j \in A} d_{kj}$$

2.

$$g(A) := \max_{k,j \in A} \{d_{kj}\}$$

Während das erste Maß die Distanzen der Objekte mittelt, ist das zweite Maß (der "Durchmesser" von A) nicht so tolerant gegenüber einem "Ausreißer" in A .

Bei Kenntnis einer Ähnlichkeitsmatrix (s_{kj}) für A erhält man analog die Maße

1.

$$k(A) := \frac{1}{n(n-1)} \sum_{k \in A} \sum_{j \in A} s_{kj}$$

2.

$$k(A) := \min_{k,j \in A} \{s_{kj}\}$$

Seien jetzt die p Merkmale der Objekte quantitativ und die Datenmatrix (x_{kj}) bekannt. Die einzelnen Objekte von A können dann wieder als Punkte des \mathbb{R}^p interpretiert werden. Sei \bar{x}_A der Mittelpunkt von A . Die Heterogenität kann gemessen werden durch

1.

$$g(A) := \sum_{k \in A} \|x_k - \bar{x}_A\|_2^2$$

2.

$$g(A) := \frac{1}{n-1} \sum_{k \in A} \|x_k - \bar{x}_A\|_2^2$$

Diese Maße basieren auf den Abweichungen vom Mittelpunkt der Menge. Das erste Maß berechnet die Summe der quadratischen Abweichungen der Objekte vom Mittelpunkt. Beim Zweiten wird die empirische Varianz von A berechnet.

2.2.3 Ähnlichkeit und Distanz von Objektmengen

Seien $A_i, A_j \subseteq S = \{O_1, \dots, O_N\}$ zwei disjunkte Objektmengen und n_i, n_j die Anzahlen der Elemente in A_i und A_j . Die Ähnlichkeit dieser Mengen wird gemessen durch eine reelle Zahl $S_{A_i A_j} \geq 0$. Je größer $S_{A_i A_j}$, desto

ähnlicher sind sich A_i und A_j . Analog dazu wird die Distanz (oder Unähnlichkeit) von A_i, A_j gemessen durch die reelle Zahl $D_{A_i A_j} \geq 0$. Je größer $D_{A_i A_j}$, desto unähnlicher (oder separierter) sind die Mengen.

Sei für S eine Ähnlichkeitsmatrix (s_{kj}) bekannt. Einige mögliche Ähnlichkeitsmaße sind

1.

$$S_{A_i A_j} := \max_{k \in A_i, l \in A_j} \{s_{kl}\}$$

2.

$$S_{A_i A_j} := \sum_{k \in A_i} \sum_{l \in A_j} s_{kl}$$

3.

$$S_{A_i A_j} := \frac{1}{n_i n_j} \sum_{k \in A_i} \sum_{l \in A_j} s_{kl}$$

4.

$$S_{A_i A_j} := \min_{k \in A_i, l \in A_j} \{s_{kl}\}$$

Die Maße unterscheiden sich in den Anforderungen an die Objektmenge: Während bei dem ersten Maß schon dann eine große Ähnlichkeit der Mengen besteht, wenn sich nur ein Objektpaar sehr ähnlich ist, müssen beim letzten Maß alle Objektpaare eine starke Ähnlichkeit aufweisen.

Wenn eine Distanzmatrix (d_{kj}) vorliegt, dann kann die Distanz zweier Mengen ganz analog zur Ähnlichkeit gemessen werden:

1.

$$D_{A_i A_j} := \min_{k \in A_i, l \in A_j} \{d_{kl}\}$$

2.

$$D_{A_i A_j} := \frac{1}{n_i n_j} \sum_{k \in A_i} \sum_{l \in A_j} d_{kl}$$

3.

$$D_{A_i A_j} := \sum_{k \in A_i} \sum_{l \in A_j} d_{kl}$$

4.

$$D_{A_i A_j} := \max_{k \in A_i, l \in A_j} \{d_{kl}\}$$

Benutzt werden solche Maße u.a. bei der hierarchischen Klassifikation, wenn bestimmt werden soll, welche Klassen einer Partition fusioniert werden sollen.

Sei für die Objektmenge S eine $N \times p$ Datenmatrix (x_{kj}) gegeben. Die Objekte seien ausschließlich durch quantitative Merkmale charakterisiert. Eine offensichtliche Methode, die Distanz zweier Objektmengen A_i, A_j zu berechnen, ist das Anwenden von Distanzmaßen für Objekte auf die Klassenmittelpunkte. Als Distanz könnte dann benutzt werden:

$$D_{A_i A_j} := d_2(\bar{x}_{A_i}, \bar{x}_{A_j})$$

Dies ist gerade der euklidische Abstand der Klassenmittelpunkte. Unter Anwendung der Mahalanobis-Distanz für Objekte ergibt sich das Distanzmaß

$$D_{A_i A_j} := (\bar{x}_{A_i} - \bar{x}_{A_j})^T \hat{\Sigma}^{-1} (\bar{x}_{A_i} - \bar{x}_{A_j})$$

3 Hierarchische Gruppierung

Hierarchische Klassifikationsverfahren erzeugen für eine gegebene Objektmenge $S = \{O_1, \dots, O_N\}$ eine Folge $\mathcal{A}^0, \mathcal{A}^1, \dots$ von Gruppierungen. Bei den *agglomerativen* Verfahren wird eine Gruppierung \mathcal{A}^ν aus der Gruppierung $\mathcal{A}^{\nu-1}$ durch Fusion einiger Gruppen erzeugt. Bei den *divisiven* Verfahren wird hingegen eine Gruppierung \mathcal{A}^ν aus der Gruppierung $\mathcal{A}^{\nu-1}$ durch Aufspaltung von Gruppen erzeugt.

3.1 Definitionen

Sei $S = \{O_1, \dots, O_N\}$ die Menge der zu klassifizierenden Objekte, \mathcal{H} ein System von nichtleeren Teilmengen aus S . \mathcal{H} heißt eine *Hierarchie* von S , wenn $A \neq B$ für je zwei Teilmengen $A, B \in \mathcal{H}$ gilt und genau eine der drei folgenden Möglichkeiten eintritt:

1. $A \cap B = \emptyset$
2. $A \subset B$
3. $B \subset A$

Die Elemente von \mathcal{H} heißen *Klassen der Hierarchie* \mathcal{H} .

Eine Hierarchie \mathcal{H} heißt *total*, wenn sie die ganze Objektmenge S und alle einelementigen Teilmengen von S enthält.

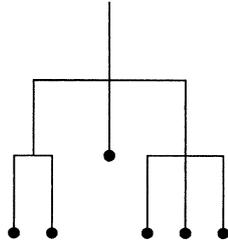


Abb. 4: Graphische Darstellung einer Hierarchie

Eine Hierarchie heißt *indizierbar*, wenn es eine nichtnegative Funktion $h : \mathcal{H} \rightarrow \mathbb{R}_+$ gibt, für die gilt:

$$A \subset B \Rightarrow h(A) < h(B) \quad \forall A, B \in \mathcal{H}$$

h heißt *Index* zur Hierarchie \mathcal{H} . Zusätzlich kann eine totale indizierte Hierarchie (\mathcal{H}, h) noch so normiert werden, daß gilt:

$$\begin{aligned} h(S) &= 1 & (i = 1, \dots, N) \\ h(\{O_i\}) &= 0 \end{aligned}$$

Eine indizierte Hierarchie wird auch als *Dendrogramm* bezeichnet.

Sei $A \in \mathcal{H}$. $h(A)$ heißt *Stufe (Niveau)* der Klasse A . Sei $h \in \mathbb{R}_+$ gegeben. Dann werden alle Klassen $A \in \mathcal{H}$ mit $h(A) \leq h$, für die es keine Klasse $B \in \mathcal{H}$ mit $h(B) \leq h$ und $A \subset B$ gibt, als *Klassen der Stufe h* bezeichnet.

Da eine Hierarchie die Ähnlichkeitsstruktur einer Objektmenge wiedergeben soll, wird für den Index $h(A)$ in der Praxis ein Maß für die Heterogenität einer Klasse benutzt. U.a. läßt sich jedes der drei folgenden Maße als Index verwenden:

$$h(A) := \sum_{j,k \in A} d_{jk} \quad \text{Distanzsumme in } A$$

$$h(A) := \max_{j,k \in A} \{d_{jk}\} \quad \text{„Durchmesser“ von } A$$

$$h(A) := \sum_{j \in A} \|x_j - \bar{x}_A\|_2^2 \quad \text{Varianz in } A$$

mit

$$\bar{x}_A := \frac{1}{|A|} \sum_{i \in A} x_i$$

Zu jeder Zahl $h \geq 0$ gibt es in einem Dendrogramm (\mathcal{H}, h) eine eindeutig bestimmte Partition $\mathcal{A}(h)$. Diese Partition enthält gerade die Klassen der Stufe h von \mathcal{H} . $\mathcal{A}(h)$ heißt *Partition der Stufe h* .

Bsp.: Sei $S = \{1, 2, 3, 4, 5\}$ und

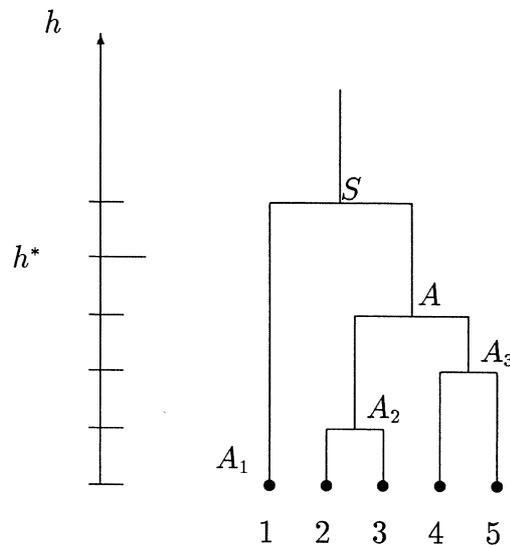


Abb. 5: Beispiel eines Dendrogramms

das zugehörige Dendrogramm. Dann ist $\mathcal{A}(h^*) = (\{1\}\{2, 3, 4, 5\})$ die Partition der Stufe h^* .

Eine Partition der Stufe h ist umso feiner, je kleiner h ist. Seien daher $h_1, h_2 \in \mathbb{R}$ mit $0 \leq h_1 < h_2$, dann stimmen die Partitionen $\mathcal{A}(h_1)$ und $\mathcal{A}(h_2)$ überein, oder $\mathcal{A}(h_1)$ ist eine Verfeinerung von $\mathcal{A}(h_2)$. Eine Verfeinerung findet nur bei bestimmten Verzweigungsstufen $h_0 = 0 < h_1 < h_2 < \dots < h_n$. Durch diese Zahlenfolge wird eine Folge von Partitionen \mathcal{A}^ν ($0 \leq \nu \leq n$) definiert, deren Klassenanzahlen m_ν streng monoton fallen.

Mit den folgenden Methoden soll für eine Objektmenge $S = \{O_1, \dots, O_N\}$ ein für das Ziel der Klassifikation geeignetes Dendrogramm (\mathcal{H}, h) konstruiert werden. Dies geschieht unter Auswertung der Ähnlichkeits- oder Distanzmatrix für die Objektmenge.

3.2 Agglomerative Verfahren

Bei agglomerativen Verfahren geht man von einer Partition der Stufe 0 $\mathcal{A}(0) = \{\{O_1\}, \{O_2\}, \dots, \{O_N\}\}$ aus. Durch Fusion von Klassen erhält man iterativ eine Folge von Partitionen $\mathcal{A}^\nu(h_\nu)$ mit monoton fallender Klassenanzahl m_ν . Die Iteration endet spätestens dann, wenn in einer Klasse alle Objekte aus S enthalten sind.

In der Praxis werden die hierarchischen Verfahren manchmal dazu benutzt, eine disjunkte Gruppierung der Objektmenge zu bestimmen, da diese Verfahren leicht verständlich und einfach durchführbar sind. Die Iteration wird dann nach bestimmten Kriterien abgebrochen, um eine Partition zu erhalten, die bestimmten Anforderungen genügt. Bei der Fusion von Klassen verringert sich die Homogenität innerhalb der Klassen. Um eine gewisse Ähnlichkeit der Objekte in einer Klasse zu garantieren, kann man daher eine Heterogenitätsgrenze wählen, die von keiner Klasse überschritten werden darf. Wird diese Grenze bei einer Partition \mathcal{A}^ν überschritten, dann ist $\mathcal{A}^{\nu-1}$ die gesuchte Partition. Als weiteres Abbruchkriterium kann eine Mindestanzahl von Klassen gefordert werden.

Sei $S = \{O_1, O_2, \dots, O_N\}$ gegeben und D_{AB} ein Maß, welches die Distanz zwischen zwei Objektmengen $A, B \subset S$ mißt. Die Bildung der Partitionenfolge geschieht dann in folgenden Schritten:

1. $\mathcal{A}^0 := (\{O_1\}, \dots, \{O_N\})$, $\nu = 0$
2. Die $(\nu - 1)$ -te Partition $\mathcal{A}^{\nu-1} = (A_1^{\nu-1}, \dots, A_{m_{\nu-1}}^{\nu-1})$ sei bereits konstruiert. Es werden die beiden Klassen $A_r^{\nu-1}, A_s^{\nu-1}$ (kurz: A_r, A_s) bestimmt, für die gilt:

$$D_{A_r, A_s} = \min_{i \neq j} \{D_{A_i, A_j}\} =: D_\nu$$

Die neue Partition \mathcal{A}^ν entsteht aus $\mathcal{A}^{\nu-1}$, indem aus den beiden Klassen A_r, A_s die Klasse $A = A_r \cup A_s$ gebildet wird. Die Partition \mathcal{A}^ν hat dann das Niveau D_ν . Wenn mehr als ein Klassenpaar den Abstand D_ν besitzt, gibt es zwei Möglichkeiten: Entweder wird zufällig ein Paar ausgewählt und nur dieses fusioniert, oder es werden solange je zwei Klassen mit dem Minimalabstand fusioniert, bis kein solches Klassenpaar mehr vorhanden ist.

3. Schritt 2 wird so lange wiederholt, bis alle Objekte aus S in einer Klasse enthalten sind².

Analog verfährt man mit einem Ähnlichkeitsmaß. Es werden dann die zwei Klassen fusioniert, welche die größte Ähnlichkeit besitzen.

²Dies ist bei \mathcal{A}^{N-1} der Fall, wenn bei einem Iterationsschritt die Klassenanzahl nur um 1 vermindert wird.

Die folgenden Verfahren unterscheiden sich im wesentlichen nur in der Wahl des Distanzmaßes. Diese unterschiedliche Wahl hat aber großen Einfluß auf die Eigenschaften der erzeugten Partitionen.

3.2.1 Single-Linkage-Methode

Sei eine Distanzmatrix (d_{jk}) für S gegeben. Das *Single-Linkage-Verfahren* (oder auch *nearest-neighbour-Verfahren*) benutzt das folgende Maß zur Messung der Distanz zwischen zwei Klassen A, B :

$$D_{A,B} := \min_{k \in A, j \in B} \{d_{jk}\}$$

Im ν -ten Iterationsschritt werden dann die beiden Klassen von $\mathcal{A}^{\nu-1}$ fusioniert, für die die Distanz der beiden Klassen minimal ist. Es werden also die zwei Klassen fusioniert, die zwei Punkte (einer aus jeder Klasse) mit minimaler Distanz enthalten. Dies kann aber zu langgestreckten Klassen oder zur sogenannten Verkettung (chaining)

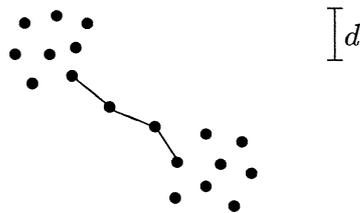


Abb. 6: Effekt der Verkettung

führen, da die Klassen einer Partition \mathcal{A}^ν gerade den Zusammenhangskomponenten der Stufe D_ν entsprechen, wenn man die Partition \mathcal{A}^ν graphentheoretisch interpretiert. Mit der Single-Linkage-Methode können also Klassen erzeugt werden, in denen sich zwei Objekte sehr unähnlich sind, da hier mehr Wert auf eine Separation der Klassen gelegt wird.

3.2.2 Complete-Linkage-Methode

Es sei wieder eine Distanzmatrix (d_{jk}) für S gegeben. Bei der *Complete-Linkage-Methode* wird die Distanz zweier Teilmengen A, B von S durch den maximalen Abstand zweier Punkte der Klassen gemessen:

$$D_{A,B} := \max_{j \in A, k \in B} \{d_{jk}\}$$

Es werden dann im ν -ten Iterationsschritt die Klassen der Partition $\mathcal{A}^{\nu-1}$ fusioniert, deren Abstand minimal ist. Die *Complete-Linkage-Methode* ist

das Gegenstück zur Single-Linkage-Methode: Durch Verwendung des maximalen Abstandes der Punkte zweier Klassen zur Berechnung der Distanz zwischen zwei Klassen müssen sich alle Punkte der Klassen hinreichend ähnlich sein. In dem mit Complete-Linkage erzeugten Dendrogramm (\mathcal{H}, h) ist bei Klassen der Stufe $h \geq 0$ der größte Abstand zweier Punkte einer Klasse gerade h . Graphentheoretisch³ betrachtet bilden diese Klassen Cliques der Stufe h . Es werden also homogene Klassen erzeugt, eine Verkettung wird vermieden. Haben bei diesem Verfahren mehrere Klassenpaare den minimalen Abstand, so wird zufällig ein Paar ausgewählt. Diese Klassen werden dann fusioniert.

3.2.3 Average-Linkage

Es sei eine Distanzmatrix (d_{jk}) für die Objektmenge S bekannt. Im ν -ten Iterationsschritt werden die Klassen $A, B \in \mathcal{A}^{\nu-1}$ fusioniert, für die der Abstand

$$D_{AB} := \frac{1}{|A| \cdot |B|} \sum_{j \in A} \sum_{k \in B} d_{jk}$$

minimal ist. Die Average-Linkage-Methode wurde als Gegenstück zu den beiden vorher beschriebenen extremen Verfahren entwickelt. Der Abstand D_{AB} zweier Klassen A und B ist dabei der mittlere Abstand der Punkte aus den Klassen. Eine etwas andere Variante des Average-Linkage-Verfahrens ist es, die Distanz zweier Klassen A, B durch den quadratischen Abstand der Klassenmittelpunkte zu bestimmen⁴:

$$D_{AB} := \|\bar{x}_A - \bar{x}_B\|_2^2$$

Die Eigenschaften dieser Verfahren, bei denen zwei Klassen fusioniert werden, für die die mittlere Distanz zwischen den Klassen minimal ist, lassen sich nicht so genau angeben wie bei Single-Linkage oder Complete-Linkage.

3.3 Divisive Verfahren

Bei den divisiven Verfahren wird eine Hierarchie \mathcal{H} dadurch erzeugt, daß ausgehend von einer Partition, die nur aus einer Klasse mit allen Objekte aus S besteht, die bisher erzeugten Klassen immer weiter in Untergruppen zerlegt werden. Das Verfahren endet, wenn eine Partition nur noch einelementige Klassen enthält. Die erste Partition \mathcal{A}^0 besteht also aus der Klasse $A_0 = S$. Sei nun eine Partition $\mathcal{A}^{\nu-1}$ konstruiert. \mathcal{A}^ν entsteht nun aus $\mathcal{A}^{\nu-1}$, indem alle Klassen $A_i \in \mathcal{A}^{\nu-1}$ unter Berücksichtigung der

³Siehe nichtdisjunkte Gruppierung

⁴Siehe Varianzkriterium

Ähnlichkeitsstruktur in zwei Untergruppen A_{i_1}, A_{i_2} zerlegt werden, sofern dies möglich ist. Die verschiedenen divisiven Verfahren unterscheiden sich darin, nach welchen Kriterien die zwei Untergruppen einer Klasse bestimmt werden⁵. Z.B. kann gefordert werden, daß der Abstand zwischen den beiden Untergruppen oder die Homogenität innerhalb der Untergruppen möglichst groß sein soll.

4 Nichtdisjunkte Gruppierung

Bei manchen Klassifikationsproblemen ist es nicht sinnvoll zu fordern, daß ein Objekt der Objektmenge $S = \{O_1, \dots, O_N\}$ nur Element *einer* Klasse A_i der gesuchten Gruppierung sein darf, daß also die gesuchten Teilmengen A_1, \dots, A_m von S paarweise disjunkt sein sollen. Sind z.B. die Übergänge zwischen den Klassen mehr graduell als abrupt (z.B. bei Eigenschaften wie riesig-groß-klein-winzig), dann sollte auch die Zugehörigkeit zu einer Klasse graduell sein, wenn die Struktur der Objektmenge durch eine Gruppierung möglichst gut erfaßt werden soll. Auch wenn die Übergänge der Klassenzugehörigkeiten abrupt sind, kann es sinnvoll sein, daß ein Objekt in mehreren Klassen enthalten ist: Bücher in Bibliotheken sind fast immer nach Sachgebieten geordnet. Manche Bücher gehören aber zu mehreren Sachgebieten⁶.

Das Problem der nichtdisjunkten Gruppierung besteht darin, für eine Objektmenge $S = \{O_1, \dots, O_N\}$ (oder kurz: $S = \{1, \dots, N\}$) ein System von Teilmengen von S zu finden, welches die Ähnlichkeitsstruktur von S möglichst gut wiedergibt, für das aber nicht gefordert wird, daß die gesuchten Teilmengen A_1, \dots, A_m paarweise disjunkt sind.

Die nichtdisjunkten Gruppierungsverfahren unterscheiden sich zum Teil erheblich. Bei einigen Verfahren definiert man den Begriff Gruppe (d.h. es wird festgelegt, wann eine Teilmenge von S als Gruppe gilt) und sucht anschließend alle in S vorhandenen Gruppen. Stellvertretend dafür wird die Gruppierung mit Hilfe von maximalen Cliques vorgestellt. Andere Verfahren gehen von einer disjunkten oder nichtdisjunkten Anfangsklassifikation aus und verbessern diese dann iterativ. Als Beispiel wird die unscharfe Klassifikation vorgestellt.

⁵Eine Beschreibung einiger divisiver Verfahren findet man u.a. bei BOCK

⁶Z.B. ein Buch, welches die Anwendung von Klassifikationsverfahren in der Wasserwirtschaft behandelt.

4.1 Maximale Cliques

Sei für $S = \{1, \dots, N\}$ eine Distanzmatrix (d_{jk}) bekannt, A eine Teilmenge von S und $d \geq 0$ eine vorgegebene Distanzschranke.

Die Menge A heißt genau dann *Clique* (der Stufe d), wenn gilt:

$$d_{jk} \leq d \quad \forall j, k \in A$$

Diese Definition enthält eine starke Homogenitätsforderung, da jedes Objekt aus A zu allen anderen Objekten aus A eine gewisse Ähnlichkeit besitzen muß.

Eine Clique A heißt *maximale* Clique (der Stufe d), wenn gilt:

$$\max_{j \in A} d_{jk} > d \quad \forall k \in S \setminus A$$

Es darf also kein Element aus $S \setminus A$ geben, das zu allen Objekten aus A eine Distanz kleiner oder gleich d besitzt.

Unter anderem sind die leere Menge und alle Mengen, die nur ein Objekt enthalten, maximale Cliques. Eine nichtleere maximale Clique wird in diesem Kapitel als *Gruppe (der Stufe d)* bezeichnet. Nach dieser Definition einer Gruppe müssen zwei Gruppen nicht disjunkt sein. Z.B. überschneiden sich in Abbildung 7 die Gruppen $\{1, 2, 5\}$ und $\{2, 3, 4\}$.

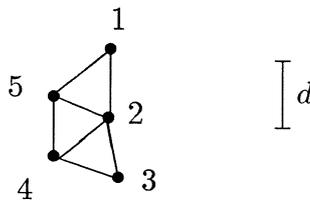


Abb. 7: Der Graph $G(d)$ für $S = \{1, 2, 3, 4, 5\}$

Sei die Objektmenge S vorgegeben. Eine maximale Clique A wird auf folgende Weise konstruiert⁷:

1. Es wird ein beliebiges Anfangsobjekt $O_j \in S$ gewählt und man setzt $A := \{j\}$.
2. Aus $S \setminus A$ wird ein Objekt O_k gesucht, für das gilt:

$$d_{jk} \leq d \quad \forall O_j \in A$$

Dieses Objekt wird dann A zugeordnet: $A = A \cup \{k\}$

⁷Ein Verfahren zur Konstruktion aller maximalen Cliques in einer gegebenen Menge findet man bei BOCK.

3. Schritt 2. wird so lange wiederholt, bis kein Objekt aus $S \setminus A$ zu A adjungiert werden kann.

je zwei Objekte einer Gruppe der Stufe d haben einen Abstand, der kleiner oder gleich d ist. Daher bildet eine solche Gruppe eine "Kugel" mit dem Durchmesser d . Für die Gruppen existiert nur eine Homogenitätsforderung. Durch die fehlende Forderung nach Separation der Gruppen können Objekte gleichzeitig zu vielen Gruppen gehören, die sich stark überschneiden, was zu einer sehr großen Anzahl von Gruppen führt. Diese Gruppen alle zu bestimmen, erfordert einen hohen Rechenaufwand. Auch mit einigen Modifikationen zur Reduzierung des Rechenaufwandes ist dieses Verfahren nur für kleinere Objektmengen geeignet.

4.2 Unscharfe Klassifikation

Diese Methode beruht auf den Grundlagen der unscharfen Mengenlehre⁸. Sei $X = (x_{jk})$ die $N \times p$ Datenmatrix, x_j der j -te Zeilenvektor. Die Zugehörigkeit eines Objektes zu einer Klasse A kann durch eine Zugehörigkeitsfunktion f_A angegeben werden. Bei der disjunkten Klassifikation nimmt diese Funktion nur zwei Werte an: $f_A(x_j) = 0$ bedeutet, daß x_j nicht Element von A ist. Ist $f_A(x_j) = 1$, so ist O_j Element der Klasse.

Bei der unscharfen Klassifikation (Fuzzy-Klassifikation) ist die Zugehörigkeit eines Objekt zu einer Klasse gradueller Natur. Dazu kann die Funktion f_A alle Werte des Intervalls $[0;1]$ annehmen. Der Wert $f_A(x_j)$ gibt dann den Grad der Zugehörigkeit des Objektes O_j zu einer Klasse A an⁹.

Das größte Problem bei der unscharfen Klassifikation besteht nun darin, eine Zugehörigkeitsfunktion zu definieren. Eine Möglichkeit ist die folgende [8]: Sei $\mathcal{A}^\nu = (A_1^\nu, \dots, A_{m_\nu}^\nu)$ eine Gruppierung der Objektmenge, n_i die Anzahl der Elemente in A_i^ν und $S_{A,j}$ ein Maß für die Nähe der Klasse A und des Objektes O_j . Je kleiner also $S_{A,j}$, desto weiter ist O_j von A entfernt. Dieses Maß kann z.B. die Ähnlichkeit zwischen Klassenmittelpunkt und dem Objekt sein.

Als Zugehörigkeitsfunktion für eine Klasse $A_i = A_i^\nu \in \mathcal{A}^\nu$ kann die folgende gewählt werden:

$$f_{A_i}(x_j) := \frac{n_i}{N} \frac{S_{A_i,j}}{\sum_{k=1}^{m_\nu} \frac{n_k}{N} S_{A_k,j}}$$

Es gilt:

⁸Diese werden z.B. bei DEIMER erklärt.

⁹Nicht zu verwechseln mit der Wahrscheinlichkeit, daß ein Objekt zu einer Klasse gehört.

$$f_{A_i}(x_j) \geq 0 \quad \text{und} \quad \sum_{k=1}^{m_\nu} f_{A_k}(x_j) = 1 \quad \forall x_j \in S$$

Alle Zugehörigkeitswerte sind also nichtnegativ und für jedes Objekt ist die Summe aller Zugehörigkeiten 1. Sei

$$I(A_i, A_j) := \frac{1}{N} \sum_{x_k \in S} \min[f_{A_i}(x_k), f_{A_j}(x_k)]$$

und

$$\Phi_f := 1 - \frac{2}{m_\nu - 1} \sum_{i=1}^{m_\nu-1} \sum_{j=i+1}^{m_\nu} I(A_i, A_j)$$

Dann ist Φ_f ein Gradmesser für die Unschärfe einer Gruppierung bzgl. der Zugehörigkeitsfunktion f . Bei $\Phi_f = 0$ besitzt die Gruppierung maximale Unschärfe. $\Phi_f = 1$ bedeutet, daß die Klassen der Gruppierung paarweise disjunkt sind.

Das Ziel der Klassifikation ist es nun, die Unschärfe zu minimieren, d.h. mit einem Iterationsverfahren wird eine Folge von Gruppierungen $\mathcal{A}^0, \mathcal{A}^1, \dots$ konstruiert, so daß $\Phi_f^\nu \leq \Phi_f^{\nu+1}$ gilt:

1. Wähle eine Anfangsgruppierung $\mathcal{A}^0 = (A_1^0, \dots, A_{m_0}^0)$
2. Sei die Gruppierung $\mathcal{A}^\nu = (A_1^\nu, \dots, A_{m_\nu}^\nu)$ konstruiert. Berechne die Klassenzugehörigkeiten $f_{A_i^\nu}(x_j)$ für alle Klassen der aktuellen Gruppierung \mathcal{A}^ν und alle Objekte aus S .
3. Berechne das Gütekriterium $\Phi_f(\mathcal{A}^\nu)$
4. Verändere die Klassen, um Φ_f zu maximieren.
5. Wiederhole die Schritte 2. – 4., bis ein Abbruchkriterium erfüllt ist.

Man erhält durch dieses Verfahren nicht nur eine Gruppierung der Objektmenge, sondern auch noch für jedes Objekt Werte über die Zugehörigkeiten zu den einzelnen Gruppen.

5 Disjunkte Gruppierung

Das Problem der disjunkten Gruppierung läßt sich wie folgt beschreiben:

Sei $S = \{O_1, \dots, O_N\}$ und $\mathcal{A} = (A_1, \dots, A_m)$ ein System von nichtleeren Teilmengen aus S . Gilt

$$A_j \cap A_k = \emptyset \quad \forall j, k \in \{1, \dots, m\}, j \neq k$$

dann heißt \mathcal{A} eine *disjunkte Gruppierung* von S . Die Mengen A_1, \dots, A_m heißen *Gruppen* oder *Klassen*. Gilt außerdem $A_1 \cup A_2 \cup \dots \cup A_m = S$, so nennt man \mathcal{A} eine *Partition* oder *exhaustive Gruppierung* von S . In einer Partition von S ist also jedes Element von S in genau einer der Gruppen A_1, \dots, A_m enthalten.

Aus der (evtl. sehr großen) Menge aller Partitionen von S wird nun die Partition gesucht, welche die Ähnlichkeitsstruktur der Objektmenge möglichst gut beschreibt.

Die meisten Methoden, um eine geeignete Partition zu finden, lassen sich in eine der drei folgenden Kategorien einordnen:

- Gruppierung mittels Gütekriterien
- Graphentheoretische Methoden
- Analyse der Punktdichte

Diese Methoden werden im folgenden genauer untersucht:

5.1 Gruppierung mittels Gütekriterien

Sei $\mathcal{A} = \{A_1, \dots, A_m\}$ eine disjunkte Gruppierung der Objektmenge S . Sei $k(\mathcal{A})$ ein Kriterium, welches angibt, wie gut \mathcal{A} die Ähnlichkeitsstruktur von S wiedergibt. Je größer der Wert der Funktion $k(\mathcal{A})$ ist, desto "besser" ist die Gruppierung \mathcal{A} . Zur Beurteilung einer Gruppierung können u.a. die Homogenität innerhalb der Klassen und die Separation der Klassen benutzt werden.

5.1.1 Varianzkriterium

Es wird vorausgesetzt, daß die Objekte O_1, \dots, O_N durch p quantitative Merkmale beschrieben werden. Es sei die Datenmatrix $X = (x_{jk})$ bekannt.

Das Varianzkriterium basiert auf der Überlegung, daß eine Klasse ähnlicher Objekte eine geringe Varianz innerhalb der Klasse besitzt. Sei

$\mathcal{A} = \{A_1, \dots, A_m\}$ eine Partition der Objektmenge S und \bar{x}_{A_i} der Mittelpunkt der Klasse A_i . Das zu minimierende *Varianzkriterium*¹⁰ wird dann definiert durch:

$$g(\mathcal{A}) := \sum_{i=1}^m \sum_{k \in A_i} \|x_k - \bar{x}_{A_i}\|_2^2$$

Bei der Anwendung des Varianzkriteriums sollten folgende Punkte beachtet werden:

- Da der Abstand eines Punktes zu seinem Klassenmittelpunkt durch den euklidischen Abstand gemessen wird, sollte die Datenmatrix normiert werden (fehlende Skaleninvarianz!).
- Die p Merkmale sollten unkorreliert bzw. stochastisch unabhängig sein, da Korrelation zu einer größeren Gewichtung einzelner Merkmale führt.
- Eine bzgl. des Varianzkriteriums günstige Partition enthält als Klassen "Hyperkugeln" des \mathbb{R}^p mit annähernd gleichem Radius und eine annähernd gleiche Anzahl von Objekten in jeder Klasse. Daher können langgestreckte oder verzweigte Klassen nicht erkannt werden.

Werden die genannten Punkte nicht beachtet, so kann die Anwendung des Varianzkriteriums zu einer verzerrten Wiedergabe der natürlichen Struktur der Objektmenge führen.

5.1.2 Kriterium für qualitative, ungeordnete Merkmale

Die p Merkmale der Objekte O_1, \dots, O_N seien qualitativ und ungeordnet. $\mathcal{X}_t = \{0, 1, \dots, m_t - 1\}$ sei die Menge der Alternativen für das Merkmal M_t ($t = 1, \dots, p$) und $X = (x_{jk})$ die zugehörige Datenmatrix.

Für jedes Merkmal M_t wird in der Objektmenge eine Partition $\mathcal{A}^t = (A_1^t, \dots, A_{m_t}^t)$ so konstruiert, daß zwei Objekte O_j, O_k genau dann in derselben Klasse von \mathcal{A}^t enthalten sind, wenn die beiden Objekte im Merkmal M_t übereinstimmen ($x_{jt} = x_{kt}$).

Eine Partition \mathcal{A} ist umso besser, je genauer sie die Struktur der Objektmenge wiedergibt. Um diese Genauigkeit zu bestimmen, wird \mathcal{A} mit den Partitionen $\mathcal{A}^1, \dots, \mathcal{A}^p$ verglichen. Je größer die Übereinstimmungen zwischen \mathcal{A} und den \mathcal{A}^t sind, desto größer die Genauigkeit. Der Unterschied zwischen den Partitionen \mathcal{A} und \mathcal{A}^t soll durch die Anzahl der Objektpaare, die bei der einen Partition in derselben Klasse enthalten sind und bei der

¹⁰Strenggenommen müßte das Kriterium das Abstandsquadratsummenkriterium heißen, da die sonst in der Mathematik übliche Definition der Varianz etwas von der hier benutzten abweicht. Es hat sich aber der Begriff *Varianzkriterium* eingebürgert.

anderen in verschiedenen Klassen liegen, gemessen werden. Sei

$$g_{jk} = \begin{cases} 1 & : O_j, O_k \text{ liegen bei } \mathcal{A} \text{ in derselben Klasse} \\ 0 & : O_j, O_k \text{ liegen bei } \mathcal{A} \text{ in verschiedenen Klassen} \end{cases}$$

($1 \leq j, k, \leq N$). Analog seien die Zahlen g_{jk}^t berechnet. Der Unterschied zwischen \mathcal{A} und \mathcal{A}^t wird ausgedrückt durch

$$\vartheta(\mathcal{A}, \mathcal{A}^t) := \sum_{j=1}^N \sum_{k=1}^N (g_{jk} - g_{jk}^t)^2$$

Für die Güte $g(\mathcal{A})$ der Partition \mathcal{A} wird das arithmetische Mittel der $\vartheta(\mathcal{A}, \mathcal{A}^1), \dots, \vartheta(\mathcal{A}, \mathcal{A}^p)$ benutzt:

$$g(\mathcal{A}) := \frac{1}{p} \sum_{t=1}^p \vartheta(\mathcal{A}, \mathcal{A}^t)$$

Es stellt sich nun die Aufgabe, in der Menge aller zulässigen Partitionen diejenige zu finden, für die $g(\mathcal{A})$ minimal wird.

5.1.3 Kriterien für beliebige Ähnlichkeits-/Distanzmaße

Sei \mathcal{B} die Menge aller möglichen Partitionen der Objektmenge S . Es sei ein Maß $g(A)$ für die Inhomogenität einer Teilmenge $A \subseteq S$ gegeben. Als Güte einer Partition $\mathcal{A} = (A_1, \dots, A_m)$ kann man dann die Summe der Inhomogenitäten der einzelnen Klassen von \mathcal{A} verwenden:

$$g(\mathcal{A}) := \sum_{i=1}^m g(A_i) = \text{Min!}, \mathcal{A} \in \mathcal{B}$$

Je kleiner also $g(\mathcal{A})$, desto homogener sind die Klassen der Partition \mathcal{A} .

Wenn man Wert auf eine gute Separation der Klassen einer Partition $\mathcal{A} = (A_1, \dots, A_m)$ legt und ein Maß D_{A_i, A_j} für die Distanz zweier Klassen A_i, A_j bekannt ist, kann man folgendes Gütekriterium verwenden:

$$k(\mathcal{A}) := \sum_{i=1}^m \sum_{j=1}^m D_{A_i, A_j} = \text{Max!}, \mathcal{A} \in \mathcal{B}$$

Wenn Maße für die Homogenität einer Klasse oder die Ähnlichkeit zweier Klassen bekannt sind, kann man die vorstehenden Kriterien leicht umformen, indem das Inhomogenitätsmaß (Distanzmaß) durch das Homogenitätsmaß (Ähnlichkeitsmaß) ersetzt wird. Die Extremalforderungen werden dann vertauscht (Min \leftrightarrow Max).

5.1.4 Numerische Verfahren

In den letzten Abschnitten wurden Kriterien für die Güte einer Partition vorgestellt. Nun soll mit diesen Kriterien aus der Menge aller disjunkten Gruppierungen diejenige Gruppierung \mathcal{A} gefunden werden, für die $k(\mathcal{A})$ maximal wird (oder analog ein Kriterium $g(\mathcal{A})$ minimal wird).

Eine Möglichkeit, das Problem zu lösen, ist das Berechnen der Werte $k(\mathcal{A})$ für alle zulässigen Partitionen $\mathcal{A} = \{A_1, \dots, A_m\}$. Für eine Objektmenge $S = \{O_1, \dots, O_N\}$ berechnet sich die Anzahl $S(N, m)$ der möglichen disjunkten Partitionen mit m nichtleeren Teilmengen wie folgt¹¹:

$$S(N, m) = \frac{1}{m!} \sum_{r=0}^m (-1)^r \binom{m}{r} (m-r)^N$$

Bei einer größeren Anzahl von Objekten ist diese Methode daher nicht praktikabel, weil der Rechenaufwand zu groß wird. Verschiedene andere Methoden, die zwar nur näherungsweise die gesuchte Partition bestimmen, dafür aber mit vertretbarem Rechenaufwand, werden hier vorgestellt.

Da die Suche nach der (bzgl. des Gütekriteriums) optimalen Partition durch Prüfen aller Möglichkeiten, meist einen zu großen Rechenaufwand erfordert, konstruiert man eine "gute" Partition der Objektmenge. Es werden drei Methoden vorgestellt¹².

Das Austauschverfahren: Sei $\mathcal{A}^0 = (A_1^0, \dots, A_m^0)$ eine vorgegebene Gruppierung der Objektmenge S . Insbesondere sei also die Anzahl m der Klassen vorgegeben. Sei $g(\mathcal{A})$ ein Kriterium für die Güte einer Partition. Beim Austauschverfahren wird versucht, eine Folge $\mathcal{A}^0, \mathcal{A}^1, \dots, \mathcal{A}^\nu, \dots$ immer besserer Partitionen von S zu erzeugen, indem einzelne Objekte aus ihren Klassen entfernt und anderen Klassen zugeordnet werden. Sei die Partition $\mathcal{A}^\nu = (A_1^\nu, \dots, A_m^\nu)$ konstruiert. Die Partition $\mathcal{A}^{\nu+1}$ wird nach folgenden Schritten erzeugt:

1. Es wird für jedes Objekt $O_k \in S$ geprüft, ob und wie stark durch den Transfer dieses Objektes aus seiner Klasse A_i^ν in eine andere Klasse A_j^ν die Güte der Partition verbessert wird.
2. Wenn es Objekte gibt, bei denen der Austausch zu einer Verbesserung führt, wird das Objekt ausgewählt, welches bei einem Transfer die größte Verbesserung der Partition bewirkt. Dann wird dieses Objekt seiner neuen Klasse zugeordnet. Die so veränderte Partition ist die gesuchte Partition $\mathcal{A}^{\nu+1}$.

¹¹Vgl. BOCK Seite 109-112

¹²Weitere Methoden, die meist Kombinationen und Modifikationen der hier aufgeführten Methoden sind, findet man bei BOCK

Es werden so lange neue Partitionen erzeugt, bis es kein Objekt aus S mehr gibt, welches durch einen Wechsel in eine andere Klasse zu einer Verbesserung führt. Abweichend könnte man auch im zweiten Schritt das erste Objekt, welches eine Verbesserung bewirken kann, transferieren.

Bei diesem Verfahren ist die Klassenanzahl vorgegeben. Da man aber a priori nicht immer die beste Klassenanzahl kennt, kann man auch so verfahren: Es wird anfangs mit einer kleinen Klassenanzahl iteriert. Wenn der Iterationsprozeß beendet ist, wird versucht, die Partition durch Aufspaltung einer Klasse in zwei neue Klassen zu verbessern. Führt die Aufspaltung zu einer Verbesserung, dann wird der Iterationsprozeß mit dieser neuen Partition und der vergrößerten Klassenanzahl erneut gestartet.

Minimal-Distanz-Verfahren: Auch bei diesem Verfahren wird versucht, iterativ eine Anfangspartition zu verbessern. Sei eine Anfangspartition $\mathcal{A}^0 = (A_1^0, \dots, A_m^0)$ von S gegeben und $D_{A,k}$ ein Maß für die Distanz zwischen der Klasse A und dem Objekt O_k . Ist in der zu berechnenden Partitionenfolge die Partition $\mathcal{A}^\nu = (A_1^\nu, \dots, A_m^\nu)$ bekannt, dann wird die Partition $\mathcal{A}^{\nu+1}$ nach folgender Vorschrift gebildet:

$$A_i^{\nu+1} := \{ O_j \mid O_j \in S \text{ und } D_{A_i^\nu, j} = \min_{1 \leq k \leq m} \{ D_{A_k^\nu, j} \} \}$$

Die Iteration bricht ab, wenn $\mathcal{A}^{\nu+1} = \mathcal{A}^\nu$. $\mathcal{A}^{\nu+1}$ heißt die zu \mathcal{A}^ν gehörende *Minimal-Distanz-Partition*. Bei diesem Verfahren wechseln im Gegensatz zum Austauschverfahren in einem Iterationsschritt häufig mehrere Objekte ihre Klasse. Dadurch erreicht dieses Verfahren oft schneller einen stabilen Zustand. Nicht bei allen Gütekriterien führt das Minimal-Distanz-Verfahren zu einer Verbesserung der Ausgangsklassifikation. Für das Varianzkriterium jedoch (für dieses ist das Minimal-Distanz-Verfahren entworfen), kann mit dem Distanzmaß $D_{A,k} := \|x_k - \bar{x}_A\|_2$ bei jedem Iterationsschritt eine Verbesserung erreicht werden [2].

Aufbau um Kerne: Sei $S = \{O_1, \dots, O_N\}$ die Objektmenge. Die Klassenanzahl m der gesuchten disjunkten Gruppierung von S ist bei dieser Methode nicht von vornherein anzugeben. Es kann sowohl eine Datenmatrix, als auch eine Ähnlichkeits- oder Distanzmatrix benutzt werden. Im Gegensatz zu anderen Methode, bei denen zuerst Klassen gebildet und dann deren Kerne (oder zentrale Punkte) mit Hilfe der Objekte in den jeweiligen Klassen berechnet werden, geht man beim rekursiven Aufbau von Klassen um Kerne den entgegengesetzten Weg: Zuerst wird ein Kernpunkt einer neuen Klasse bestimmt und dann alle Objekte gesucht, die in dieser Klasse enthalten sein sollen. Wurden alle geeigneten Objekte dieser Klasse zugeordnet, dann wird in der restlichen Objektmenge der nächste Kernpunkt bestimmt. Die Konstruktion der Gruppierung geschieht allgemein nach folgenden Schritten:

1. In $S = \{O_1, \dots, O_N\}$ wird das Objekt O_k gesucht, welches als Kern der ersten Klasse A_1 am besten geeignet ist, und man wählt $A_1 = \{O_k\}$.
2. Aus der Menge der restlichen Objekte $A \setminus A_1$ wird das Objekt gesucht, welches die geringste Distanz oder die größte Ähnlichkeit zu A_1 besitzt. Dieses wird der Klasse A_1 zugeordnet.
3. Schritt 2. wird so lange wiederholt, bis das Zuordnen eines weiteren Objektes zu A_1 die Heterogenität innerhalb von A_1 zu groß werden lassen würde oder ein anderes Abbruchkriterium erfüllt ist. Ist kein weiteres geeignetes Objekt vorhanden, dann ist der Aufbau der Klasse A_1 beendet. Die Objekte der Klasse A_1 werden aus S entfernt.
4. Mit der reduzierten Objektmenge werden weitere Klassen A_2, A_3, \dots durch Ausführung der Schritte 1.-3. gebildet. Das Verfahren endet, wenn keine Objekte mehr zu klassifizieren sind oder eine Klassifikation der restlichen Objekte ungünstig ist (z.B., wenn die folgenden Klassen zu klein wären).

Sei jetzt im Verfahren die ν -te Klasse zu bilden und $U_\nu \subseteq S$ die Menge der bisher unklassifizierten Objekte.

Bestimmung des Kernpunktes: Als Kern der Klasse A_ν kann das Objekt $O_k \in U_\nu$ gewählt werden, welches die größte Punktdichte aufweist, d.h. das Objekt, für das die Anzahl der Objekte $O_j \in U_\nu$ mit $d_{jk} \leq d^*$ maximal ist. d^* ist dabei eine vorgegebene Distanzschranke und d_{jk} wie bisher ein Maß für die Distanz zweier Objekte. Eine zweite Möglichkeit ist, als Kern von A_ν mehrere Objekte aus U_ν zu wählen, z.B. das Objektpaar, welches von allen möglichen Objektpaare in U_ν die größte Ähnlichkeit besitzt.

Wahl eines zu adjungierenden Objektes: Sei jetzt $A_\nu \neq \emptyset$. Die Wahl des nächsten, zu A_ν zu adjungierenden Objektes O_k , ist abhängig von dem Maß für die Distanz $D_{A_\nu, k}$ (oder für die Ähnlichkeit $S_{A_\nu, k}$) zwischen A_ν und O_k . Die Wahl eines Maßes unterliegt keinen besonderen Einschränkungen, beeinflusst aber evtl. das Endergebnis.

Kriterien zum Abbruch der Adjunktion: Wenn das Objekt $O_k \in U_\nu$ bestimmt wurde, welches zu A_ν die geringste Distanz (größte Ähnlichkeit) besitzt, muß entschieden werden, ob O_k zu A_ν adjungiert werden soll oder ob eine neue Klasse gebildet werden soll. Folgende Kriterien können zur Entscheidung verwendet werden:

1. Ist die Distanz $D_{A_\nu, k}$ größer als eine vorgegebene Schranke $d > 0$, dann wird das Objekt nicht mehr der Klasse A_ν zugeordnet.

2. Sei $g(\cdot)$ ein Maß für die Heterogenität einer Objektmenge und γ eine vorgegebene Schranke. O_k wird genau dann der Klasse A_ν zugeordnet, wenn

$$g(A_\nu \cup \{O_k\}) \leq \gamma$$

gilt.

3. Der Abbruch kann auch abhängig vom Zuwachs der Heterogenität bei Hinzunahme von O_k zu A_ν erfolgen: Sei $\delta > 0$ eine vorgegebene Schranke. Die Adjunktion stoppt, wenn gilt:

$$g(A_\nu \cup \{O_k\}) - g(A_\nu) > \delta$$

Das Verfahren zum rekursiven Aufbau einer Gruppierung um Kernpunkte wird häufig dazu benutzt, eine Anfangsgruppierung für andere Klassifikationsverfahren zu erzeugen.

5.2 Graphentheoretische Methoden

Zum Verständnis der graphentheoretischen Interpretation der folgenden Methoden werden zuerst einige Grundbegriffe der Graphentheorie eingeführt:

Sei V eine Menge und sei $E \subseteq \{\{u, v\} \mid u, v \in V\}$. Ein Graph $G = (V, E)$ ist das geordnete Paar der *Ecken* V und *Kanten* E .

Ein Graph G heißt *endlich*, wenn $V(G)$ eine endliche Menge ist.

Betrachtet man als Kante statt $\{u, v\}$ das geordnete Paar (u, v) , dann spricht man von einem *gerichteten* Graphen, ansonsten von einem *ungerichteten*.

Eine Kante $\{u, u\}$ bezeichnet man als *Schlinge*. Ist $u \neq v$ für alle $\{u, v\} \in E(G)$, dann ist der Graph *schlingenfrei*.

Ist I eine endliche Indexmenge, und ist $E(G) \subseteq \{\{u, v\} \mid u, v \in V(G)\} \times I$, dann ist $G = (V(G), E(G))$ ein *Multigraph* mit *Mehrfachkanten*.

In der vorliegenden Arbeit werden nur endliche, ungerichtete, schlingenfreie Graphen ohne Mehrfachkanten betrachtet.

Ein Graph heißt *vollständig*, wenn je zwei Ecken des Graphen durch eine Kante verbunden sind.

Ein *Teilgraph* entsteht aus einem Graphen G , indem in G einige Kanten entfernt werden.

Ein *Untergraph* entsteht aus einem Graphen G , indem einige Ecken und alle von diesen Ecken ausgehenden Kanten aus G entfernt werden.

Ein Graph heißt *gewichtet*, wenn jeder Kante des Graphen eine Zahl (*Gewicht*) zugeordnet ist.

Sei $G = (V, E)$ ein Graph und $i, n \in \mathbb{N}$. Eine endliche Folge von Ecken (v_1, v_2, \dots, v_n) heißt *Kantenzug*, wenn für $1 \leq i < n$ gilt: $(v_i, v_{i+1}) \in E$. Ist $v_1 = v_n$, dann ist der Kantenzug *geschlossen*, ansonsten *offen*. Die *Länge* dieses Kantenzuges ist $n - 1$.

Ein *offener (geschlossener) Weg* ist ein offener (geschlossener) Kantenzug mit paarweise verschiedenen Ecken, d.h. ein Weg ist *kreuzungsfrei*.

Ein Graph heißt *zusammenhängend*, wenn zwischen je zwei Ecken ein Kantenzug existiert. Ein nichtzusammenhängender Graph zerfällt in maximale zusammenhängende (*Zusammenhangs-*) *Komponenten*.

Ein *aufspannender Baum* ist ein zusammenhängender Graph, der keine geschlossenen Wege enthält. Daher hat ein aufspannender Baum mit N Ecken genau $N - 1$ Kanten.

Sei G ein gewichteter zusammenhängender Graph. Der *minimale aufspannende Baum (Minimalbaum)* ist aus der Menge aller möglichen aufspannenden Bäume von G derjenige, bei dem die Summe der Gewichte seiner Kanten minimal ist.

5.2.1 Definition einer Gruppe

Sei für $S = \{O_1, \dots, O_N\}$ eine Distanzmatrix (d_{jk}) bekannt, A eine Teilmenge von S und $d > 0$ eine vorgegebene Distanzschranke.

Wählt man ein Objekt $O_k \in A$, dann bezeichnet man die Menge aller Objekte $O_j \in A$, für die $d_{jk} \leq d$ gilt, als *d-Umgebung* von O_k .

Eine Teilmenge $A \subseteq S$ heißt *Gruppe der Stufe d*, wenn gilt:

1. $A \neq \emptyset$
2. $O_j \in A \Rightarrow$ Alle Objekte der d -Umgebung von O_j sind Elemente von A .
3. Es gibt keine Menge $B \subset A$, welche die ersten beiden Bedingungen erfüllt.

Wenn man festlegt, daß sich zwei Objekte $O_j, O_k \in A$ genau dann ähnlich sind, wenn $d_{jk} \leq d$ gilt, dann wird also nicht gefordert, daß alle Objekte einer Gruppe A einander ähnlich sind. Erzeugt man in der Menge S alle Gruppen der Stufe d , so erhält man eine Partition $\mathcal{A} = (A_1, \dots, A_m)$ der

Menge S . Die Anzahl $m = m(d)$ der disjunkten Gruppen ist für ein festes d eindeutig bestimmt. In einer so konstruierten Gruppierung wird mehr Wert auf Separation der Gruppen gelegt als auf die Homogenität innerhalb der Gruppen. Eine so konstruierte Gruppierung heißt *Gruppierung durch Zusammenhangskomponenten* oder auch *Single-Linkage-Methode*.

Graphentheoretische Interpretation: Sei $S = \{O_1, \dots, O_N\}$ die Menge der zu gruppierenden Objekte und (d_{jk}) eine Distanzmatrix für S . G sei ein vollständiger, gewichteter Graph mit N Ecken. Die Ecken sollen die Objekte O_1, \dots, O_N repräsentieren. Das Gewicht einer Kante zwischen zwei Objekten O_j, O_k soll der Wert d_{jk} sein.

Wenn man aus G alle Kanten entfernt, deren Länge größer als d ist, erhält man einen Teilgraph $G(d)$ des Graphen G . Der Graph $G(d)$ enthält dann nur noch Kanten einer Länge kleiner als d . Die Zusammenhangskomponenten von $G(d)$ sind dann identisch mit den Gruppen der Stufe d .

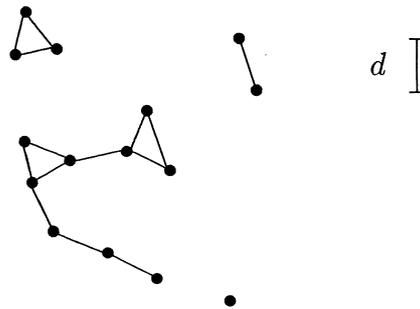


Abb. 8: Eine Gruppierung der Stufe d . Die Distanz sei der euklidische Abstand.

5.2.2 Konstruktion einer Gruppierung durch Zusammenhangskomponenten

Bis jetzt wurde festgelegt, wie eine Gruppe beschaffen sein soll. Nun wird beschrieben, wie man eine entsprechende Gruppierung durch Zusammenhangskomponenten konstruiert.

Eine Gruppe A der Stufe d in der Objektmenge $S = \{O_1, \dots, O_N\}$ kann so konstruiert werden:

1. Wähle ein beliebiges Objekt $O_k \in S$ und setze $A := \{O_k\}$
2. Füge zu A alle Objekte $O_j \in S$ hinzu, für die $d_{jk} \leq d$ und $O_j \notin A$ gilt. Markiere O_k .

3. Wähle ein Objekt $O_k \in A$, welches bisher noch nicht markiert wurde. Ist kein unmarkiertes Objekt mehr vorhanden, dann ist A eine Gruppe der Stufe d , das Verfahren endet. Ansonsten weiter bei Schritt 2.

Soll eine Gruppierung der Stufe d konstruiert werden, wird der Algorithmus auf die verkleinerte Objektmenge $S \setminus A$ angewendet. Dies geschieht solange, bis jedes Objekt klassifiziert wurde.

5.2.3 Verwendung des Minimalbaumes

Sei G ein vollständiger, gewichteter Graph, dessen Ecken die Objekte aus S repräsentieren. Den Minimalbaum von G kann man mit folgendem Algorithmus erzeugen:

1. Wähle die Kante mit dem kleinsten Gewicht. Sie ist Teil des Minimalbaumes.
2. Wähle aus der Menge der übrigen Kanten die Kante mit dem kleinsten Gewicht aus, die keinen geschlossenen Weg erzeugt.
3. Schritt 2 wird so lange wiederholt, bis es keine Kante mehr gibt, die keinen geschlossenen Weg erzeugt (bei N Ecken sind es gerade $N - 1$ Kanten).

Entfernt man im Minimalbaum von G alle Kanten, deren Gewicht größer als d ist, dann sind die so entstandenen Zusammenhangskomponenten die Gruppen der Stufe d , die Gruppierung ist fertig.

Die Verwendung des Minimalbaumes hat einen Vorteil: Hat man mit Hilfe des Minimalbaumes eine Gruppierung der Stufe d erzeugt und wählt ein $\bar{d} < d$, dann erhält man die Gruppen der Stufe \bar{d} , indem man in den Zusammenhangskomponenten alle Kanten mit einem Gewicht größer als \bar{d} entfernt. Mit einer Folge monoton fallender d_i ($i \in \mathbb{N}$) kann man auf diese Weise mit dem Minimalbaum von G auch eine hierarchische Gruppierung konstruieren.

5.3 Analyse der Punktdichte

Wenn die zu klassifizierenden Elemente einer Objektmenge S als Punkte des \mathbb{R}^2 dargestellt werden können, und die Aufgabe gestellt wird, diese Punkte anhand der 2-dimensionalen Darstellung zu klassifizieren, dann werden sehr wahrscheinlich die Bereiche der Ebene als Klassen angegeben, in denen die

Punktdichte (im Gegensatz zu ihrer Umgebung) relativ hoch ist. Solche Klassen werden auch als "natürliche" Klassen bezeichnet.

Sei $S = \{O_1, \dots, O_N\}$ die Menge der zu klassifizierenden Objekte. Es wird angenommen, daß die Objekte p quantitative Merkmale besitzen, und ihre zugehörige Datenmatrix $X = (x_{jk})$ soll bekannt sein. Die Objekte werden als Punkte x_1, \dots, x_N des \mathbb{R}^p interpretiert. Das Klassifikationsproblem besteht darin, eine disjunkte, nicht notwendigerweise exhaustive Gruppierung der Objekte zu bestimmen. Die Anzahl der Klassen sei dabei unbekannt.

Zur Analyse der Punktdichte wird eine p -dimensionale Zufallsvariable X betrachtet. X sei stetig verteilt mit der Dichte f .

Besitzt die Dichtefunktion $f : \mathbb{R}^p \rightarrow \mathbb{R}$ nur ein lokales Maximum, dann wird f als *unimodal* bezeichnet. Besitzt f mehrere lokale Maxima, dann heißt f *multimodal*. Die Maxima werden als *Modalwerte* von f bezeichnet.

Die Modalwerte haben für die Klassifikation folgende Bedeutung: Man betrachtet die Punkte $x_1, \dots, x_N \in S$ als zufällige Beobachtungen von X , also als zufällige Stichprobe einer größeren Objektmenge (Population) Π . Es wird angenommen, daß Π tatsächlich eine Struktur aus "natürlichen" Klassen besitzt. Π zerfalle in die Einzelpopulationen Π_1, \dots, Π_m . Besitzt jede Population Π_i eine eigene Verteilungsdichte $f_i(x)$ mit $x \in \mathbb{R}^p$, dann häufen sich die $x_k \in S$ in der Nähe der Modalwerte der $f_i(x)$.

Eine Klassifikation wird dadurch konstruiert, daß die Modalwerte der Dichtefunktion von X näherungsweise bestimmt werden und dann jedes Objekt $x_k \in S$ einem Modalwert zugeordnet wird. Alle Objekte, die dem gleichen Modalwert zugeordnet werden, bilden eine Klasse.

5.3.1 Verwendung der relativen Punktdichte

Das Verfahren von SCHNELL: Sei $F(x) = f(x; x_1, \dots, x_N)$ ein Maß für die Punktdichte der Punkte x_1, \dots, x_N an der Stelle $x \in \mathbb{R}^p$. Ziel des Verfahrens ist es, die zu F gehörenden Modalwerte näherungsweise zu bestimmen. Dabei werden die Beobachtungspunkte x_i mit einem Gradientenverfahren in Richtung der gesuchten Modalwerte verschoben. Eine Partitionierung der Objektmenge erhält man, indem man alle Objekte, die zu dem gleichen Modalwert gehören, derselben Klasse zuordnet. Die Anzahl der Klassen ergibt sich aus der Anzahl der Modalwerte.

Für das Maß f des Einflusses eines Punktes $x_k \in S$ auf einen Punkt $x \in \mathbb{R}^p$ wird die Normalverteilung benutzt:

$$f(x; x_k; \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^p} \cdot \exp\left(-\frac{1}{2\sigma^2}\|x - x_k\|^2\right)$$

x_k ist also der Erwartungswert einer p -dimensionalen, normalverteilten Zufallsvariable. Der Gesamteinfluß der Objekte auf einen Punkt x wird durch die Summe

$$F(x; x_1, \dots, x_N; \sigma) = \sum_{k=1}^N f(x; x_k; \sigma)$$

angegeben. F wächst mit der Anzahl der Punkte in der Umgebung von x . Somit ist F ein Maß für die Konzentration der x_k in der Umgebung von x .

Die Objekte x_k werden nun iterativ in Gebiete mit höherer Punktdichte in Richtung der gesuchten Modalwerte verschoben. Alle Punkte, die zum gleichen lokalen Maximum verschoben werden, bilden eine Klasse. Die Klassenanzahl wird bei dem Iterationsverfahren nicht vorgegeben, sie ergibt sich aus der Anzahl der Modalwerte, welche von σ abhängt. Der Parameter σ beeinflusst die Separation der Klassen: Wird σ hinreichend klein gewählt, dann erhält man, wenn die x_1, \dots, x_N paarweise verschieden sind, als Gruppierung die Klassen $\{1\}, \dots, \{N\}$. Je größer σ gewählt wird, desto geringer ist die Anzahl der Klassen¹³.

Sei $\delta > 0$ eine gewählte Schrittweite. Die Punkte werden in Richtung des steilsten Anstieges verschoben: Diese Richtung wird angegeben durch den Gradientenvektor

$$\begin{aligned} \nabla F(x) &:= \text{grad } F(x) = (F_1(x), \dots, F_p(x)) \\ &= \frac{-1}{(2\pi)^{p/2} \cdot \sigma^{p+2}} \sum_{k=1}^N (x - x_k) \cdot \exp\left(-\frac{1}{2\sigma^2} \|x - x_k\|_2^2\right) \end{aligned}$$

mit

$$F_i(x) = \frac{\partial}{\partial x_i} F(x)$$

Für jedes Objekt x_k ($k \in (1, \dots, N)$) wird eine Folge von Punkten $x_k^{(0)}, x_k^{(1)}, x_k^{(2)}, \dots$ nach folgender Vorschrift konstruiert:

$$\begin{aligned} x_k^{(0)} &:= x_k \\ x_k^{(i)} &= x_k^{(i-1)} + \delta \cdot \frac{\nabla F(x_k^{(i-1)})}{\|\nabla F(x_k^{(i-1)})\|_2} \quad k = 1, \dots, N \end{aligned}$$

Dabei gibt δ die Schrittweite für die Verschiebung an. Das Iterationsverfahren bricht ab, wenn die Punkte nicht mehr in ein Gebiet höherer

¹³Wegen dieser Eigenschaft kann das Verfahren von Schnell auch zur Konstruktion einer hierarchischen Gruppierung benutzt werden. Man benutzt dazu eine geeignete streng monoton wachsende Folge $0 < \sigma_0 < \sigma_1 < \dots$ von σ -Werten. Ist die Differenz zweier aufeinanderfolgender Werte $\sigma_{i+1} - \sigma_i$ hinreichend klein, dann entsteht die Partition $\mathcal{A}(\sigma_{i+1})$ aus $\mathcal{A}(\sigma_i)$ durch Vereinigung einiger Klassen von $\mathcal{A}(\sigma_i)$.

Punktdichte verschoben werden können. Dann gilt

$$F(x_k^{(n)}) - F(x_k^{(n-1)}) < 0$$

5.3.2 Verwendung der absoluten Punktdichte

Manchmal sind bei einem Klassifikationsvorgang kleine Gruppen mit geringer Punktdichte unerwünscht, etwa weil sie für das Ziel der Klassifikation keine Bedeutung haben ("Ausreißer"). Ein Beispiel wären die Hintergrundgeräusche bei der Klassifizierung von akustischen Signalen. Es interessieren dann nur Bereiche, die eine vorgegebene Mindestdichte $d > 0$ aufweisen und durch Bereiche, die die Mindestdichte nicht besitzen, voneinander getrennt sind. Diese Bereiche hoher Punktdichte werden auch als Zusammenhangskomponenten des \mathbb{R}^p bezeichnet. Objekte in Bereichen mit geringer Punktdichte werden von der Klassifikation ausgeschlossen, d.h. im allgemeinen sind die so erzeugten Gruppierungen nicht exhaustiv. Eine Erhöhung der Mindestdichte d bewirkt, daß bestehende Gruppen aufgespalten werden und die Anzahl der nicht klassifizierten Objekte wächst.

Auch das Verfahren von Schnell kann zu einer Klassifikation anhand der absoluten Punktdichte verwendet werden, wenn man nach Anwendung des Verfahrens diejenigen Punkte aus der konstruierten Gruppierung entfernt, die in Bereichen liegen, deren Punktdichte geringer ist als die vorgegebene Mindestdichte. Man kann aber auch wie folgt vorgehen: Die Objekte der Objektmenge S seien durch p quantitative Merkmale charakterisiert. Auf den p Koordinatenachsen werden M Intervalle so eingerichtet, daß jeder Punkt $x_k \in S$ in einem der auf diese Weise konstruierten M^p Mehrfachintervalle liegt. Für jedes dieser Intervalle wird berechnet, ob die Punktdichte den vorgegebenen Mindestwert erreicht. Erreichen zwei benachbarte Mehrfachintervalle die Mindestdichte, dann gehören sie zu derselben Zusammenhangskomponente. Ein solches Abzählverfahren erfordert jedoch einen hohen Rechen- und Speicherplatzaufwand.

Das Verfahren von WISHART Das Verfahren von Wishart benutzt sowohl eine Analyse der Punktdichte als auch graphentheoretische Methoden. Die Dichte an einem Punkt x_k wird gemessen durch die Anzahl von Punkten, die in einer d -Umgebung von x_k liegen.

Als Voraussetzung für diese Methode muß eine (metrische) Distanzmatrix (d_{jk}) für die Menge $S = \{O_1, \dots, O_N\}$ der zu klassifizierenden Objekte vorliegen.

1. Wahl einer Distanzschranke $d > 0$ und eines Schwellwertes $c \geq 0$ für die Punktdichte.

2. Berechnung der Punktdichte c_j für jeden Punkt $x_j \in S$, definiert als die Anzahl der Punkte $x_k \in S \quad k \neq j$ mit $d_{jk} \leq d$
3. Alle Punkte $x_k \in S$, für die $c_k < c$ gilt, werden aus S entfernt:
 $S^* := S \setminus \{x_k \mid c_k < c\}$
4. Für S^* wird der Graph $G(d)$ berechnet. Die Zusammenhangskomponenten des Graphen bilden die nichtexhaustive, disjunkte Gruppierung $\mathcal{A}^* = (A_1^*, \dots, A_m^*)$ von S^* .
5. Die Punkte aus $S \setminus S^*$ werden nach der nearest neighbour Methode den Klassen A_1^*, \dots, A_m^* zugeordnet. Es wird dadurch eine Partition $\mathcal{A} = (A_1, \dots, A_m)$ von S konstruiert.

Wenn man Punkte in Gebieten mit geringer Punktdichte von der Klassifikation ausschließen will, stoppt man das Verfahren nach der Konstruktion von \mathcal{A}^* , da \mathcal{A}^* dann die gewünschte Gruppierung ist.

Durch Veränderung der Parameter c und d kann man den Klassifikationsprozeß beeinflussen:

Gilt $c > 0$, dann werden isolierte Punkte "Ausreißer" vorläufig aus der Klassifikation ausgeschlossen. Je größer der Wert c (bei festem d), desto stärker wird eine mögliche Verkettung berücksichtigt, da zwei Bereiche mit hoher Punktdichte, die durch einen Bereich mit geringer Punktdichte getrennt sind, dann nicht aufgrund einiger weniger Zwischenpunkte zu einer Klasse vereinigt werden können.

Je größer der Wert d (bei festem c) gewählt wird, desto weniger Zusammenhangskomponenten werden gebildet (schlecht separierte Klassen werden zu einer Klasse fusioniert) und desto weniger Punkte werden von der Klassifikation vorläufig ausgeschlossen.

Dieses Verfahren setzt nur Kenntnis einer Distanzmatrix voraus. Da eine Datenmatrix $X = (x_{jk})$ nicht benötigt wird, ist dieses Verfahren nicht auf quantitative Merkmale beschränkt und auch für Objekte mit qualitativen oder gemischten Merkmalen geeignet.

Teil II

Das regelbasierte System

6 Kanalnetzsteuerung

Kanalnetze haben die Aufgabe, anfallende Wassermengen (Abwasser und/oder Regenwasser) ohne Belastung der Umwelt zu transportieren. Vermeidung von Umweltbelastungen bedeutet dabei, daß

- ein Überstauen der Schächte verhindert wird.
- eine Entlastung von Schmutzwasser (um Überstau zu vermeiden) auf ein notwendiges Minimum reduziert wird.
- die Kläranlagen nicht überlastet werden, um die Reinigungsleistung zu erhalten.

Sind die drei Forderungen erfüllt, dann tritt die Forderung nach einer Minimierung der Betriebskosten (z.B. Energiekosten) in den Vordergrund.

Im Zuge eines wachsenden Umweltbewußtseins und einer ständig wachsenden Bebauung steigen auch die Anforderungen an die Kanalnetze. Um die Leistung der bestehenden Kanalnetze zu vergrößern und den gesteigerten Anforderungen anzupassen, gibt es mehrere Möglichkeiten: Durch bauliche Veränderungen kann die Kapazität oder Transportleistung eines Netzes vergrößert werden. Solche Maßnahmen sind aber nicht immer durchführbar (z.B. wenn der Platz für Rückhaltebecken fehlt). Zudem sind diese Änderungen kostenintensiv. Durch Verbesserungen im betrieblichen Bereich kann versucht werden, die vorhandenen Kapazitäten besser zu nutzen. Dies setzt voraus, daß ungenutzte Kapazitäten vorhanden sind und der Abwassertransport durch eine Steuerung hinreichend stark beeinflußt werden kann.

Steuerung eines Abwassertransportes bedeutet dabei, daß durch die aktiven Bauelemente (Pumpen, Schieber, verstellbare Wehre usw.) der Abwassertransport so gelenkt wird, daß vorhandene Kapazitäten optimal genutzt werden. Jedes Steuerelement wird dabei anhand der aktuellen Situation im Kanalnetz gesteuert. Wird ein Bauwerk nur nach lokalen Gesichtspunkten gesteuert¹⁴ und ohne Berücksichtigung des Zustandes der übrigen Bereiche des Netzes, dann spricht man von einer *lokalen* Steuerung.

¹⁴Speziell für eine Pumpe bedeutet lokale Steuerung, daß die Entscheidungen nur vom Zustand des Pumpensumpfes abhängen.

Um eine optimale Bewirtschaftung des Netzes zu erreichen, muß für die Steuerentscheidungen über die Pumpen, Schieber, usw. der Zustand des gesamten Netzes berücksichtigt werden (Verbundsteuerung). Durch eine Steuerung des Schmutzwassertransportes nach lokalen Gesichtspunkten wird häufig nicht die gesamte Kapazität des Netzes genutzt. Zwar können auch mit einer Verbundsteuerung nicht immer die vorhandenen Kapazitäten aufgrund von baulichen Gegebenheiten¹⁵ oder zu geringer Möglichkeiten der Beeinflussung des Abwassertransportes zu 100% genutzt werden, in vielen Fällen übertrifft jedoch die Leistung einer Verbundsteuerung die einer lokalen Steuerung.

Die Steuerung eines komplexen Kanalnetzes (unter Berücksichtigung der genannten Anforderungen) ist eine komplizierte Angelegenheit. Erschwert wird sie durch den zeitlichen Aspekt: In kritischen Situationen müssen die Steuerentscheidungen innerhalb kurzer Zeit getroffen werden. Im Institut für Wasserwirtschaft der Universität Hannover wird seit einigen Jahren daran gearbeitet, die Kanalnetzsteuerung mit Mitteln der künstlichen Intelligenz zu erleichtern. Dazu wurde ein regelbasiertes System¹⁶ entwickelt, welches die Steuerung eines Kanalnetzes übernehmen oder Empfehlungen für den Betreiber liefern kann. Dieses System wird in Bremen dazu benutzt, im On-Line-Betrieb Empfehlungen für die Steuerung des Kanalnetzes Bremen-links der Weser zu geben. Um eine Steuerung mit diesem regelbasierten System off-line beobachten zu können, wurde das regelbasierte System in einer Version mit dem hydrodynamischen Abwassertransportmodell EXTRAN gekoppelt. Das Transportmodell EXTRAN simuliert für einen vorgegebenen Oberflächenabfluß den Abwassertransport innerhalb eines Kanalnetzes. Diese Koppelung ermöglicht es, anhand von Simulationen die Leistung von Steuerstrategien zu beobachten und zu bewerten.

7 Regelbasierte Systeme

Ein regelbasiertes System (RBS) ist ein Programm, welches Wissen über einen speziellen Problemkreis in Form von Regeln speichert, bei Vorgabe konkreter Probleme dieses Gebietes aus seinem Wissen Schlußfolgerungen zieht und Problemlösungen anbietet. Regelbasierte Systeme zeichnen sich durch eine strikte Trennung verschiedener Wissensebenen aus. Es werden drei Wissensebenen unterschieden:

- Das allgemeine Wissen über die Durchführung des Problemlösens

¹⁵In dem später vorgestellten Kanalnetz kann z.B. nicht durch Steuerung erreicht werden, daß die Speicherschächte 6 und 11 (EXTRAN-Darstellung des Netzes) gleichmäßig gefüllt werden, da der Zufluß von Schacht 6 zu Schacht 11 nicht gesteuert werden kann

¹⁶Die erste Generation der Expertensysteme bestand überwiegend aus regelbasierten Systemen

- Das Fachwissen über einen Problemkreis
- Das Anwendungswissen über ein konkretes Problem des Problemkreises

Bei herkömmlichen Programmen sind die ersten beiden Wissens Ebenen Teil des Programmcodes. Im Gegensatz dazu ist bei einem RBS nur das Wissen über die Durchführung des Problemlösens ein fester Bestandteil des Programmcodes. Die Regelmenge, welche das Fachwissen enthält, ist ein Bestandteil der Daten. Diese Trennung des Fachwissens vom Programmcode hat verschiedene Vorteile:

Jemand, der einem regelbasierten System das Wissen über einen speziellen Anwendungsbereich in Form von Regeln vermittelt, braucht keine Kenntnisse über das Programmieren zu besitzen. Andererseits braucht jemand, der ein RBS programmiert, sich nicht um den späteren Anwendungsbereich zu kümmern.

Da die Regelmenge ein Teil der Daten ist, kann sie geändert werden, ohne das Programm zu verändern. Dies ist z.B. nützlich, wenn das Einsatzgebiet des RBS geändert oder erweitert wird.

Die Regeln enthalten also das anwendungsbezogene Wissen. Um konkrete Probleme (evtl. im Dialog mit dem Benutzer) lösen zu können, muß dem System das Problem in Form von Fakten mitgeteilt werden. Diese Fakten müssen das Problem hinreichend genau beschreiben, damit das RBS alle Informationen über das spezielle Problem erhält, die zur Lösung des Problems notwendig sind. Das Faktenwissen wird im "Kurzzeitgedächtnis" (short time memory = STM) gespeichert. Das Regelwissen steht in der Regelbasis (production memory). Die Wissensbasis (working memory) umfaßt das Regel- und Faktenwissen.

Regelbasierte Systeme haben folgende Struktur:

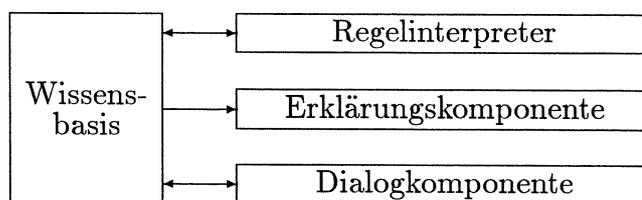


Abb. 9: Struktur eines regelbasierten Systems

Die Dialogkomponente dient der Kommunikation des Benutzers mit dem System. Über die Dialogkomponente wird dem RBS das Problem mitge-

teilt. Dazu werden die Fakten, die das Problem beschreiben, in die Wissensbasis eingetragen. Der Regelinterpreter (oder Problemlösungskomponente, Inferenzmaschine) versucht, mit Hilfe der in der Wissensbasis gespeicherten Fakten und Regeln, das Problem zu lösen. Wenn zum Lösen des Problems Fakten fehlen oder vorhandene Fakten zu ungenau sind, können diese bei manchen Systemen durch Interaktion mit dem Benutzer über die Dialogkomponente ergänzt werden. Das Faktenwissen wird entweder dadurch geändert, daß dem System über die Dialogkomponente neue Fakten zugeführt werden, oder es werden in einem Deduktionsprozeß durch Anwendung von Regeln durch den Regelinterpreter neue Fakten hergeleitet (daher "Deduktion"). Dabei können evtl. auch Zwischenresultate in der Wissensbasis gespeichert werden, wenn sich das gewünschte Ergebnis nicht unmittelbar durch die Regeln erzeugen läßt. Eine Regel hat die allgemeine Form

```

WENN
    Bedingung 1    und
    Bedingung 2    und
        :
    Bedingung n
DANN
    Aktion 1      und
    Aktion 2      und
        :
    Aktion m

```

Eine Regel kann ausgeführt (angewendet) werden, wenn ihre Bedingungen erfüllt sind. Anwendung einer Regel heißt, daß die Aktionen der Regel auf das Faktenwissen angewendet werden. Im allgemeinen bedeutet es, daß im STM neue Fakten eingetragen werden und evtl. veraltete Fakten aus dem STM entfernt werden.

Ein Deduktionsprozeß kann entweder datenorientiert oder zielorientiert durchgeführt werden: Ein Regelinterpreter arbeitet datenorientiert, wenn er ausgehend von den vorhandenen Fakten durch Anwendung von Regeln neue Fakten erzeugt. In diesem Fall wird ein Regelinterpreter als vorwärtsverkettend oder vorwärtsschließend bezeichnet (Entsprechend nennt man das regelbasierte System dann ein vorwärtsverkettendes R3S), Bei einem zielorientierten (rückwärtsschließenden) Regelinterpreter werden zuerst ein oder mehrere Ziele ausgewählt. Dies kann z.B. eine zu prüfende Hypothese sein. Dann wird versucht, eine Verbindung zwischen den Zielen (der Hypothese) und den bekannten Fakten durch die Regeln herzustellen. Jede angewendete Regel kann dabei zu neuen Zielen führen. Die Ziele werden solange untersucht, bis sie aus der Faktenmenge abgeleitet werden können

(bis sich die Hypothese als 'wahr' oder 'falsch' erweist).

Die Regelmenge wird als *vollständig* bezeichnet, wenn zu jedem Problem des Problemkreises vom RBS eine Lösung erzeugt wird. Die Regelmenge heißt *konsistent*, wenn sie widerspruchsfrei ist, d.h. wenn zu keiner Zeit eine Aussage gleichzeitig wahr und falsch ist.

Durch die Erklärungskomponente kann das regelbasierte System dem Benutzer darlegen, wie es aus den gegebenen Fakten über den Deduktionsprozeß eine Problemlösung erarbeitet hat. Dies umfaßt im wesentlichen die Angaben, welche der Regeln angewendet wurden und in welchem Zustand das STM sich bei der Anwendung befand.

8 Maschinelles Lernen

Unter maschinellem Lernen eines Systems versteht man die selbständige Adaption von Wissen, um die Leistung des Problemlösens zu erhöhen. Bei einem regelbasierten System ist das Ziel des Lernens, die Regelmenge qualitativ oder quantitativ zu verbessern. Dies kann durch Erweiterung der Regelmenge geschehen, durch das Löschen von Regeln oder durch Modifikation der Kriterien zur Auswahl von Regeln im Inferenzprozeß.

Die Steigerung der Leistung kann darin bestehen, daß

- die Qualität der Problemlösungen bzgl. eines Gütekriteriums verbessert wird.
- die Menge der vom RBS lösbaren Probleme vergrößert wird (quantitative Verbesserung).
- die Effektivität des Systems verbessert wird. Die Effektivität kann gemessen werden an dem vom System benötigten Speicherplatz oder an dem zur Lösung der Probleme benötigten Rechenaufwand. Die quantitative und qualitative Problemlöseleistung bleibt bei einer Verbesserung der Effektivität unverändert.

Es werden mehrere Arten von Lernen unterschieden:

- Lernen aus Beispielen
- Lernen aus Beobachtungen/Erfahrungen
- Lernen durch Experimente

Diese Verfahren sind in der Reihenfolge der Ansprüche, die an das lernende System gestellt werden, geordnet.

Beim *Lernen aus Beispielen* werden dem System einige (gute und schlechte) Beispiele für die Änderung des Wissens vorgegeben. Das System verallgemeinert diese Beispiele, um die gesamte Änderung des Wissens zu erzeugen. Danach wird wie beim Lernen mit Lehrer diese Änderung in Regeln umgesetzt. Lernen aus Beispielen erfordert Wissen über die Erstellung von Regeln, über Konsistenz und Vollständigkeit der Regelmenge sowie über das Umfeld, in dem das System arbeitet, damit die Verallgemeinerung sinnvoll geschieht.

Beim *Lernen aus Beobachtungen* leitet das System die Änderung des Wissens von Beobachtungen seines Umfeldes ab. Diese Beobachtungen werden dann verallgemeinert und in Regeln umgesetzt. Um aus diesen Beobachtungen lernen zu können, muß das System wissen, wie diese Beobachtungen zu bewerten sind und wie diese Bewertung in die Änderung des Wissens einfließen soll.

Beim *Lernen durch Experimente* führt das System selbständig Versuche durch, deren Beobachtung und Bewertung zur Änderung des Wissens führt. Von dem System wird dabei zusätzlich gefordert, daß es eigenständig Experimente planen und durchführen kann.

9 Das implementierte RBS

Der Regelinterpretier des implementierten RBS ist eine vorwärtsschließende Inferenzmaschine. Die Dialogkomponente des RBS besteht aus einer Schnittstelle zum Abwassertransportmodell EXTRAN. Über diese Schnittstelle erhält das RBS die Fakten über das zu lösende Problem. Außerdem wird EXTRAN die Lösung des Problems über die Schnittstelle mitgeteilt. Die Erklärungskomponente besteht aus einer Ausgabedatei, die Informationen über die durchgeführten Deduktionsprozesse enthält.

9.1 Die Wissensbasis

In das STM werden Werte eingetragen, die den aktuellen Zustand des simulierten Kanalnetzes beschreiben. Durch die Anwendung der Steuerregeln werden diese Werte ergänzt und verändert. Das Kurzzeitgedächtnis (short time memory, STM) des RBS ist eine Liste von Tripeln der Form (V,O,W). V ist hierbei eine Variable (Parameter), O ein Operator und W ein Wert. Bei den Werten wird zwischen Zahlenwerten (quantitativ) und Prädikaten (qualitativ) unterschieden. Ob einer Variablen ein quantitativer Wert

oder ein qualitativer Wert zugeordnet ist, entscheidet der Operator. Eine Variable kann gleichzeitig ein qualitativer Wert und ein Prädikat. Aber keine zwei (oder mehr) Werte der gleichen Art. Die Mengen der erlaubten Variablen, Operatoren und Prädikate werden beim Start des Programms bestimmt.

Ein Eintrag im STM bedeutet, daß eine bestimmte Aussage wahr ist: Steht z.B. im STM der Eintrag (XYZ, ' =R', 200), dann bedeutet dies, daß die Aussage "Die Variable XYZ besitzt den numerischen Wert 200" wahr ist.

Die Regelmenge wird beim Start des Programms in den Regelspeicher (production memory) eingelesen. Die allgemeine Form einer Regel ist

$$B_1 B_2 \dots B_n \rightarrow A_1 A_2 \dots A_m$$

wobei die Literale B_i, A_i wie im STM aus Variablenname, Operator und Wert bestehen. Die Literale B_i werden als Bedingungen, die Literale A_i als Aktionen bezeichnet. Folgende Operatoren sind für die Bildung einer Bedingung erlaubt:

Direkter Vergleich von Wert einer Variablen und einem Wert (z.B. "XYZ =R 100"):

Für quantitative Werte : <R <=R =R >=R >R

Für qualitative Werte : < <= = >= >

Vergleich der Werte zweier Variablen (z.B. "XYZ >=C ZYX"):

Für quantitative Werte : <V <=V =V >=V >V

Für qualitative Werte : <C <=C =C >=C >C

Diese Menge der Operatoren ist fest und kann nicht ohne eine Programmänderung vergrößert werden, da das Wissen über die Auswertung einer Bedingung Teil des Programmes ist. Zusätzlich können noch weitere Operatoren benutzt werden, um Fakten zu beschreiben (z.B. die Aktionen von Metaregeln im Lernprozeß). Diese Fakten dürfen aber nicht benutzt werden, um zu prüfen, ob eine Bedingung einer Regel erfüllt ist.

Beispiel einer Regel:

(REGEN = TRUE) (SCH1 >=R 200) -> (P1 = 2)

Für jede Regel stehen noch die folgenden Informationen zur Verfügung:

- Alter einer Regel: Dieser Wert gibt an, wie lange eine Regel schon in der Regelmenge enthalten ist. Bei einer Koppelung mit EXTRAN bezieht sich der Wert auf die Simulationszeit.
- Anwendungshäufigkeit: Jedesmal, wenn in einem Deduktionsprozeß

eine Regel angewendet wird, wird dieser Wert um Eins erhöht.

- **Strafwert:** Bei Durchführung eines Lernprozesses werden Regeln "bestraft", die für eine "schlechte" Steuerung verantwortlich sind. Als Strafe wird dieser Wert um einen bestimmten Betrag erhöht. Je größer der Wert, desto weniger erfolgreich ist die Regel.

9.2 Auswahl und Anwendung der Regeln

In den meisten regelbasierten Systemen werden durch den Regelinterpreter solange Regeln angewendet, bis ein vorgegebenes Ziel erreicht (Problem gelöst) ist, keine neuen Fakten mehr erzeugt werden können, oder es keine Regel gibt, deren Bedingungen genügend erfüllt sind. Der implementierte Regelinterpreter geht folgendermaßen vor:

Die Menge der Regeln ist in mehrere gleichberechtigte Blöcke unterteilt. In jedem Block können bis zu 100 Regeln stehen. Die Regelmenge wird einmal von Anfang bis Ende vom Interpreter analysiert. Für die Regelblöcke werden beim Start des Programms Werte eingelesen, die angeben, zu wieviel Prozent die Bedingungen einer Regel erfüllt sein müssen, damit die Regel angewendet werden darf. Aus jedem Block wird höchstens eine Regel angewendet. Aus der Menge der Regeln eines Blockes, deren Bedingungen am besten erfüllt sind, wird die Regel angewendet, bei der der Wert

$\text{Alter} \cdot \text{Anwendungshäufigkeit} / \text{Strafwert}$

maximal ist. Anwendung einer Regel bedeutet, daß die Literale A_i des Aktionsteils der Regel in das STM als neue Fakten eingetragen werden. Zweckmäßigerweise sollten alle Regeln, die Entscheidungen über dasselbe Steuerelement treffen, in einem Block stehen. Dadurch wird gewährleistet, daß für dieses Steuerelement nicht mehrere Entscheidungen getroffen werden. Das Treffen mehrerer Entscheidungen für ein Steuerelement würde das Nachvollziehen des Entscheidungsprozesses erschweren. Die Eindeutigkeit der Entscheidungen wird aber dadurch gewährleistet, daß beim Eintragen einer Aktion (V,O,W) ein alter Eintrag für V im STM gelöscht wird, wenn durch das Eintragen der neuen Aktion die Variable V zwei qualitative oder zwei quantitative Werte besitzen würde. Das bedeutet auch, daß die Menge der Fakten bei Anwendung einer Regel widerspruchsfrei bleibt.

In der Version des regelbasierten Systems mit dem neuen Lernverfahren wurde die Regelauswahl modifiziert: Aus der Menge der Regeln eines Blockes, deren Bedingungen am besten erfüllt sind, wird die Regel mit dem kleinsten Strafwert angewendet.

9.3 Die Koppelung EXTRAN↔RBS

Es wird nun der Datenfluß zwischen dem Simulationsteil und dem RBS beschrieben. Die Koppelung von EXTRAN und dem RBS läßt sich graphisch wie folgt darstellen:

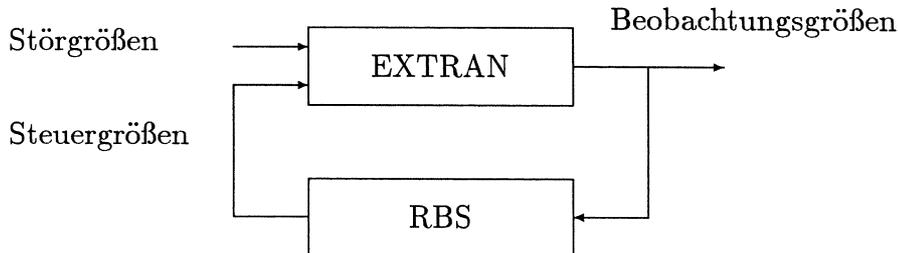


Abb. 10: Schema der Koppelung

EXTRAN simuliert den Abwassertransport in diskreten Schritten. Nach einer bestimmten Anzahl von Simulationsschritten wird das RBS aufgefordert, die Entscheidungen über die Steuerelemente im Kanalnetz zu treffen. Zu diesem Zweck werden dem RBS Fakten zur Verfügung gestellt, welche (zusammen mit der Regelmenge) die Grundlage für den Deduktionsprozeß bilden.

Eine Situation im Kanalnetz wird in EXTRAN durch viele verschiedene Beobachtungsgrößen charakterisiert (z.B. Wasserstände, Durchflüsse, Pumpenstufen). Aus den aktuellen Werten dieser Größen werden die an das RBS zu übertragenden Fakten gebildet¹⁷. Folgende Arten von Beobachtungsgrößen können übertragen werden:

- Wasserstände in Schächten
- Wasserstände in Rohren (Oberes oder unteres Ende einer Haltung¹⁸)
- Durchflüsse (durch Pumpen, Wehre oder Haltungen)

Diese Daten werden gemittelt über die letzte Minute vor dem aktuellen Zeitpunkt, um eventuell auftretende Oszillationen der Werte, bedingt z.B. durch numerische Instabilitäten, zu vermeiden.

¹⁷Dieses Verfahren entspricht der Bildung von Features bei der Mustererkennung.

¹⁸Im Modell EXTRAN besteht eine Haltung aus zwei Schächten, die durch ein Rohr verbunden sind, und evtl. einem Einzugsgebiet

Auch aus den Niederschlagsdaten hergeleitete Informationen (die Störgrößen¹⁹) werden übertragen : Dies sind die Niederschlagsmengen aus der Zeit vor dem aktuellen Zeitpunkt und in der Version mit dem neuen Lernverfahren auch Vorhersagedaten über die zu erwartenden Regenwasserzuflußmengen.

Vor der Simulation mit einem Kanalnetz wird eine Einheitsganglinie für den Zufluß in das System erstellt: Für einen einminütigen Blockregen mit 5 mm Niederschlagshöhe wird der Oberflächenabfluß zum vorgegebenen Kanalnetz berechnet. Aus dieser Zuflußwelle wird eine Einheitsganglinie erstellt. Sie gibt den zeitlichen Verlauf des Zuflusses für einen einminütigen Blockregen mit 1 mm Niederschlagshöhe und 1 ha undurchlässiger Fläche eines Einzugsgebietes wieder. Während einer Simulation wird aus den Daten über den gefallenen Regen mit der für das Kanalnetz erstellten Einheitsganglinie eine Zuflußvorhersage berechnet. Diese Vorhersage gibt näherungsweise an, wieviel Kubikmeter Regenwasser in den nächsten Minuten²⁰ pro Hektar undurchlässiger Fläche eines Einzugsgebietes in das System fließen wird.

Die Störgrößen, Regeninformation und Zuflußvorhersage, haben einen besonderen Stellenwert für eine Kanalnetzsteuerung. Anhand einer genauen Vorhersage über die zukünftige Zuflußmenge können bevorstehende Engpässe im System frühzeitig erkannt werden und geeignete Gegenmaßnahmen getroffen werden.

Das RBS wirkt über die Steuergrößen auf den Abwassertransport ein: Die Steuergrößen sind das Steuerintervall (die Zeitspanne zwischen zwei Entscheidungsprozessen) sowie die Entscheidungen über die Pumpenstufen²¹

Für den Datenfluß EXTRAN ↔ RBS existiert ein Übertragungsprotokoll: Den Namen der EXTRAN-Variablen werden dabei Namen von Variablen des RBS zugeordnet. Diese Liste legt damit die Beziehungen zwischen den Variablen in EXTRAN und den im RBS benutzten Variablen fest. Damit wird auch festgelegt, welche EXTRAN-Werte zum RBS übertragen werden und welche Entscheidungen zu EXTRAN übertragen werden.

Die Wahl der Informationen ist wie bei der automatischen Klassifikation ein sehr wichtiger Aspekt. Wenn dem RBS keine aussagekräftigen Informationen über den Kanalnetzzustand vorliegen, können auch nur eingeschränkt

¹⁹Bei einem realen Kanalnetz ist die Menge der Störgrößen größer: Dort muß z.B. auch die Menge der Schmutzwassereinleitungen berücksichtigt werden. Für EXTRAN wird angenommen, daß diese Einleitungen konstant sind.

²⁰Das Intervall wird vom Benutzer gewählt. Da der zukünftige Niederschlag nicht berücksichtigt werden kann, ist die Vorhersage umso genauer, je kleiner das Vorhersageintervall gewählt wird.

²¹Andere Steuerelemente als Pumpen müssen für eine Simulation mit EXTRAN durch Pumpen dargestellt werden. In der Realität sind Pumpen meist stufenlos, EXTRAN benutzt jedoch eine begrenzte Anzahl an Pumpenstufen.

brauchbare Steuerentscheidungen getroffen werden²².

10 Verbesserung der Steuerung durch Lernen

Lernen bedeutet für das RBS, daß durch Beobachten und Bewerten der Steuerung erkannt wird, ob die Steuerung unzulänglich war und wenn ja, daß die Regelmenge geändert wird, um diese Unzulänglichkeiten zu beheben. Dies geschieht in den folgenden Lernverfahren in mehreren Schritte

1. Bewertung der bisherigen Steuerung
2. Verbesserung der Steuerstrategie
3. Umsetzung der Verbesserungen an der Steuerung in neue Regeln

Für das Lernen werden noch weitere Bausteine für das RBS benötigt: Die Metaregeln und das Entscheidungsgedächtnis.

Metaregeln: Für den Lernprozeß gibt es *einen* Block mit speziellen Regeln, den *Metaregeln* (Abgrenzend dazu werden die übrigen Regeln als *Steuerregeln* bezeichnet). Mit diesen Regeln wird die Steuerung beurteilt: Durch den Bedingungsteil einer Metaregel wird festgelegt, wann ein unerwünschter Kanalnetzzustand vorliegt. Wenn also die Bedingungen einer Metaregel erfüllt sind, dann liegt ein Kanalnetzzustand vor, der durch eine Änderung der Steuerstrategie in Zukunft vermieden werden soll.

Die Syntax der Metaregeln und die Bedeutung ihrer Aktionen sind abhängig vom implementierten Lernverfahren und werden daher bei den jeweiligen Lernverfahren beschrieben. Metaregeln und Steuerregeln unterscheiden sich nicht nur in der Syntax und Ausführung, sondern auch in der Auswahl der Regeln. Im Block mit den Metaregeln wird jede Regel angewendet, deren Bedingungen (B_i s.o.) alle erfüllt sind.

Entscheidungsgedächtnis: Wenn zu einem bestimmten Zeitpunkt der Simulation erkannt wird, daß die Steuerung unzulänglich war, dann sollen die Steuerentscheidungen für zurückliegende Kanalnetzzustände korrigiert werden. Dazu benötigt man das Wissen, wie die Steuerentscheidungen in früheren Entscheidungsintervallen getroffen wurden. Dazu werden die

²²Ein Beispiel: Soll durch eine Regel die Auslastung zweier Speicherbecken miteinander verglichen werden, so sind Wasserstandsinformationen nur dann ausreichend, wenn für beide Becken die gleichen Wasserstands-Volumen Beziehungen gelten.

Nummern aller Steuerregeln, die angewendet wurden, im Entscheidungs-gedächtnis (decision memory = DM) gespeichert. Aufgrund der Dimen-sionierung des DM kann nur eine bestimmte Anzahl von Entscheidungs-schritten gespeichert werden. Weiter zurückliegende Informationen über Steuerentscheidungen stehen einem Lernverfahren nicht zur Verfügung.

10.1 Das bisherige Lernverfahren

Dieses Verfahren wurde von NEUMANN entwickelt und später in einigen Teilen modifiziert.

10.1.1 Der Strafprozeß

Während einer Simulation wird durch die Metaregeln festgestellt, wann ein Fehler in der Steuerung vorliegt: Ein Zustand²³, der zur Anwendung einer Metaregel führt, gilt dabei als unerwünscht. Eine Metaregel hat die Form

$$B_1, B_2, \dots, B_n \rightarrow (V, 'TOO', v) (STRAF, ' =R', s) (ZURUK, ' =R', z)$$

Die Bedingungen B_i haben die gleiche Bedeutung wie die Bedingungen der Steuerregeln.

Zur Aktion $(V, 'TOO', v)$: Durch den Variablennamen V wird festgelegt, welches Steuerelement schlecht gesteuert wurde und damit für dem unerwünschten Zustand verantwortlich ist. Die Variable V darf die Prädikate ' LOW' oder ' HIGH' besitzen. Hat V den Wert ' LOW', dann wurde der zugehörigen Steuervariablen ein zu geringer Steuerwert zugeordnet. Zu den Aktionen $(STRAF, ' =R', s)$ und $(ZURUK, ' =R', z)$: Der Werte der Variablen STRAF bestimmt, wieviele Strafpunkte eine Steuerregel erhält, die dem schlecht gesteuerten Bauwerk in den letzten ZURUK Entscheidungsschritten einen Wert zugewiesen hat. Eine Metaregel gibt also die Fehlerursache (z.B. 'P1WAS TOO LOW'), eine Strafpunktzahl und die Anzahl der Zeitschritte, die der Fehler zurückverfolgt werden soll, an. Durch die Angabe der Fehlerquelle wird gleichzeitig auch bestimmt, wie die Steuerstrategie geändert werden soll.

Beispiel einer Metaregel:

$$(SCH2 >R 500) \rightarrow (P2WAS TOO LOW) (STRAF =R 10) (ZURUK =R 4)$$

Das bedeutet: Wenn der Wert der Variablen SCH2 größer als 600 ist, dann wurde der Variablen P2 in den letzten 4 Zeitschritten ein zu geringer Wert zugewiesen. Dafür verantwortliche Regeln erhalten maximal 10 Strafpunkte.

²³Als Zustand oder Kanalnetzzustand wird der Zustand der Variablen des STM, die eine Kanalnetzsituation beschreiben, verstanden.

Anhand der Informationen im DM werden alle Regeln bestraft, die zu der Fehlsteuerung beigetragen haben. Regeln des letzten Zeitschrittes erhalten die volle Strafpunktzahl STRAF. Weiter zurückliegende Entscheidungen werden geringer bestraft. Für jeden Block von Steuerregeln, dessen Regeln bestraft und verändert werden dürfen, wird eine Liste von Variablennamen eingelesen. Diese Variablen werden im folgenden als Blockvariablen bezeichnet. Bei der Neubildung von Regeln für einen Steuerregelblock dürfen nur die zugehörigen Blockvariablen im Bedingungsteil benutzt werden. Nach der Bestrafung einer Regel werden dann die Nummer der bestraften Steuerregel, die Nummer der Metaregel, die für die Bestrafung verantwortlich ist, die Höhe der Strafe und die zur Zeit der Anwendung dieser Regeln gültigen Werte der Blockvariablen gespeichert. Diese Strafdaten werden zur Neubildung von Regeln benötigt.

Der Bestrafungsprozeß findet während eines Simulationslaufes statt. Er beeinflußt (durch Änderung der Strafwerte) die Regelauswahl des Inferenzprozesses, das Löschen von Regeln, wenn der Regelspeicher zu voll wird und natürlich die Änderung der Steuerstrategie.

10.1.2 Konstruktion neuer Regeln

Bisher wurde also die Steuerung bewertet und eine Lösung des Problems bestimmt. Jetzt muß diese Lösung in Regeln umgesetzt werden. Dies geschieht nach einem Simulationsvorgang anhand der während der Bestrafung von Regeln gespeicherten Daten. Für jede Steuerregel, die in einem Block steht, dessen Regeln bestraft werden dürfen, werden folgende Schritte unternommen:

1. Wahl einer Bewertungsregel und Bestimmung der Strafpunkte und die Anzahl der Strafen, die die Steuerregel durch diese Metaregel erhalten hat. Ist keine weitere Metaregel mehr vorhanden, dann endet die Neubildung von Regeln für diese Steuerregel.
2. Wenn die Anzahl der Strafen eine Mindestanzahl unterschreitet oder die Strafpunkte einen Schwellwert nicht überschreiten, dann weiter bei 1.
3. Aus den Strafdaten für diese Regel, d.h. den Zuständen, in denen die Steuerregel angewendet und durch die Metaregel bestraft wurde, werden für jede Blockvariable Mittelwert und mittlere Streuung berechnet.
4. Aufgrund der Strafdaten für die alte Steuerregel wird eine neue Regel durch folgende Arbeitsschritte erzeugt:

- (a) Wahl einer quantitativen Blockvariablen, für die der Quotient aus Streuung und Mittelwert MW kleiner als ein bestimmter Grenzwert ist (Gilt $MW = 0$, dann muß die Streuung auch Null betragen). Ist keine solche Blockvariable mehr vorhanden, dann weiter bei c).
- (b) Trage in den Bedingungsteil der neuen Regel das Intervall $[MW - \text{Streuung}, MW + \text{Streuung}]$ ein (z.B. ' $H01 \geq R 10$ $H01 < R 20$ '). Weiter bei a).
- (c) Wähle eine qualitative Blockvariable. Sei W der am häufigsten aufgetretene Wert in den Strafdaten für diese Regel. Liegt die relative Häufigkeit unterhalb eines vorgegebenen Grenzwertes, dann wähle eine andere Variable. Ist keine solche Blockvariable mehr vorhanden, dann weiter bei e).
- (d) Trage die Bedingung "Blockvariable gleich W" (z.B. ' $ZP1 = 2$ ') in den Bedingungsteil der neuen Regel ein. Weiter bei c).
- (e) Übernahme des Konklusionsteils der alten Regel, wobei der neue Steuerwert durch Steigerung oder Verminderung des alten Steuerwertes der bestraften Regel bestimmt wird, je nachdem, ob der alte Wert als zu hoch oder zu niedrig bewertet wurde. Weiter bei Schritt 1.

10.1.3 Löschen von Regeln

Wenn kein Platz im Regelspeicher vorhanden ist, um eine Regel einzutragen, werden einige Steueregeln gelöscht. Es sollen nur Regeln gelöscht werden, die keinen (oder nur einen geringen) Beitrag zur Steuerung leisten: In der Regelmenge wird die Regel gelöscht, für die der Quotient

Anwendungshäufigkeit / Alter

minimal ist. Haben mehrere Regeln diesen minimalen Wert, dann wird die Regel gelöscht, für die der Quotient

Strafe / Anwendungshäufigkeit

maximal ist. Das Löschen der Regeln wurde für die RBS-Version mit dem neuen Lernverfahren übernommen.

10.2 Anmerkungen zum alten Lernverfahren

Der aktuelle Zustand des Kanalnetzes wird durch die Werte der Beobachtungs- und Störgrößen, die in das Kurzzeitgedächtnis eingetragen werden, charakterisiert. Jede Variable in EXTRAN und auch im RBS

kann nur endlich viele Werte annehmen. Dies ist dadurch bedingt, daß zum einen alle möglichen Werte einer Variablen innerhalb eines Intervalls liegen (z.B. kann der Wasserstandspegel in einem Rohr nur Werte zwischen Null und dem Rohrdurchmesser annehmen) und zum anderen alle Werte in digitaler Form vorliegen, also nur auf einige Dezimalen genau sind. Sei M die Anzahl der Steuerelemente und Z die Menge aller möglichen Zustände des Kanalnetzes, charakterisiert durch die Werte der Beobachtungs- und Störgrößen²⁴, die im STM den Zustand des Kanalnetzes beschreiben.

Steuerung bedeutet nun, daß jedem Element von Z ein Steuervektor (v_1, \dots, v_M) mit Entscheidungen über die M Steuerelemente zugeordnet ist. Z läßt sich in Klassen eingeteilt, so daß die Steuervektoren zweier Elemente einer Klasse übereinstimmen und zwei Elemente aus verschiedenen Klassen unterschiedliche Steuervektoren besitzen.

Das Lernen läßt sich damit so formulieren: Durch den Strafprozeß wird bestimmt, ob die Steuerentscheidungen eines Zustandes geändert werden müssen. Änderung der Entscheidungen bedeutet, daß die Werte einiger Komponenten des zugehörigen Steuervektors geändert werden. Für das i -te Steuerelement gibt es eine Menge $Z_i \subseteq Z$ von Zuständen, deren Steuerentscheidung über s_i geändert werden soll²⁵. Durch Ausnutzung des Wissens über die Zusammenhänge im Kanalnetz werden den Elementen in Z_i neue Entscheidungen über s_i zugeordnet.

Da die Menge der Steuerregeln in Blöcke eingeteilt wird, und in einem Block die Regeln für *ein* Steuerelement stehen, werden Regeln erzeugt, die über genau ein Steuerelement eine Entscheidung treffen. Die Komponenten der Steuervektoren werden also einzeln betrachtet.

Sei $E_i = \{e_{i1}, \dots, e_{in_i}\}$ die Menge aller möglichen Steuerentscheidungen für das i -te Steuerelement. Jede der Mengen Z_i läßt sich derart in Mengen Z_{i1}, \dots, Z_{in_i} unterteilen, daß in einer Menge Z_{ij} alle Zustände enthalten sind, denen die Entscheidung e_{ij} für das Steuerelement s_i zugeordnet wurde.

Es werden neue Steuerregeln geschaffen, so daß es für jedes $z \in Z_{ij}$ eine neue Regel gibt, deren Bedingungen durch z erfüllt werden und durch deren Aktionsteil die i -te Steuervariable den Wert e_{ij} erhält.

Im bisherigen Lernverfahren wird durch den Bedingungsteil einer Metaregel ein Problem im Kanalnetz beschrieben. Sind die Bedingungen einer Metaregel für den aktuellen Zustand erfüllt, dann tritt das Problem im Netz auf. Durch den Aktionsteil wird festgelegt, wie dieses Problem vermieden werden kann, indem angegeben wird, welcher Steuervariablen ein falscher

²⁴Nach den vorhergehenden Betrachtungen besitzt Z endlich viele Elemente.

²⁵Wenn die Entscheidungen über bestimmte Steuerelemente nicht durch einen Lernprozeß geändert werden sollen, dann sind die zugehörigen Z_i leer. Im folgenden wird oBdA. angenommen, daß Z_i nicht leer ist

Wert zugewiesen wurde. Es wird auch angegeben, wie weit diese falschen Entscheidungen zurückverfolgt werden sollen und wie die Entscheidungen korrigiert werden sollen. In den Metaregeln steckt also das gesamte Wissen über die Bildung der Z_{ij} $i = 1, \dots, M, j = 1, \dots, n_i$.

Es sei jetzt ein Steuerelement s_i und eine Entscheidung e_{ij} $j \in (1, \dots, n_i)$ gewählt. Sei $S = Z_{ij}$. Durch jedes Element in S wird ein Kanalnetzzustand beschrieben. Interpretiert man die Elemente als Punkte im \mathbb{R}^p , dann gibt es in diesem Raum Bereiche, in denen die Punktdichte höher ist als in ihrer Umgebung. Dies sind Bereiche, für die die Steuerung von s_i korrigiert wurde. In diesen Bereichen soll für s_i die Steuerentscheidung e_{ij} getroffen werden. Das Problem liegt nun darin, diese natürlichen Gruppen (siehe Analyse der Punktdichte) zu lokalisieren und durch Regeln zu beschreiben.

Das bisherige Verfahren teilt die Menge S folgendermaßen in Klassen ein: In einer Klasse stehen alle Elemente aus S , die durch Anwendung derselben Metaregel in Z eingetragen wurden. Für jede dieser Klassen wird dann nach dem schon angegebenen Algorithmus genau eine Steuerregel erstellt.

Diese Vorgehensweise geschieht unter der Annahme, daß alle Zustände in S , die aufgrund der Anwendung derselben Metaregel in Z eingetragen werden, eine natürliche Gruppe im Zustandsraum bilden. Außerdem darf diese Gruppe nicht verzweigt sein, damit sie sich durch genau eine neue Regel hinreichend genau beschreiben läßt.

Diese recht scharfen Annahmen treffen in vielen Fällen nicht zu und daher beschreiben die erzeugten Regeln im allgemeinen nur unzulänglich die Änderung der Steuerentscheidungen der Elemente von Z . Dies wird durch ein Beispiel verdeutlicht: Mit der Anfangsregelmenge zum selbständigen Entwickeln einer Steuerstrategie durch das bisherige Lernverfahren wurden die Ereignisse 1 und 3 simuliert.

Die Entscheidungen für die Steuervariable P6 wurden durch die Metaregel
 (SCH11 \geq R 600)(SCH14 $<$ R 600) \rightarrow (P6WAS TOO LOW)(STRAF =R
 -10)(ZURUK =R 4)

bewertet. Es werden nun die Blockvariablen SCH6 und SCH11 für die Steuervariable P6 betrachtet. Diese Variablen enthalten die mittleren Wasserstände der letzten Minute für die Speicherschächte 6 und 11.

In der Abbildung 11 sind die Zustände (charakterisiert durch die Werte der beiden Blockvariablen), in denen die Regel

(DUMMY =R 0) \rightarrow (P6 =R 0)

bestraft wurde, als Punkte des \mathbb{R}^2 dargestellt. Mit dem bisherigen Verfahren wurde eine neue Steuerregel erstellt. Das eingezeichnete Rechteck kennzeichnet den Geltungsbereich der neuen Regel für die Variablen SCH6

und SCH11. Innerhalb des Rechtecks liegen 90% aller Punkte. Die restlichen Zustände werden nicht durch die Regel erfaßt, sie werden aber zur Berechnung der Streuung für die einzelnen Variablen mit einbezogen. Es werden also mehrere Zustände nicht durch die Regel erfaßt. Der Geltungsbereich der Regel hätte wesentlich kleiner gewählt werden können, um die innerhalb des Rechtecks liegenden Punkte zu überdecken.

Zustände mit der Entscheidung P6 = R 1

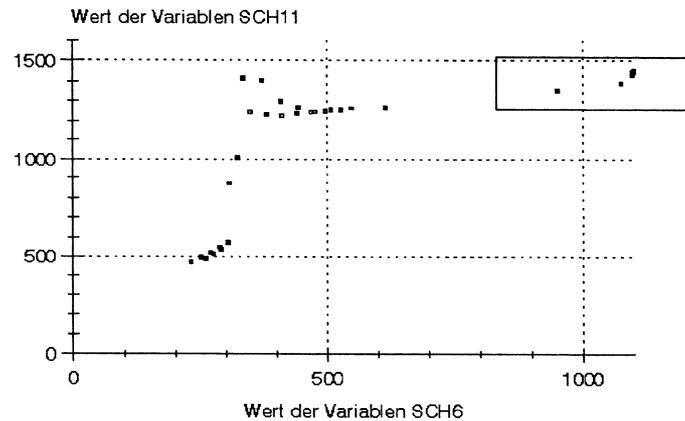


Abb. 11: Darstellung des Gültigkeitsbereiches einer neuen Regel, erstellt mit dem alten Lernverfahren.

Das bisherige Verfahren hat noch eine weitere erwähnenswerte Eigenschaft im Zusammenhang mit der Umsetzung der Punkte in neue Regeln: Verschiebt man die Punkte parallel zu der Achse einer Variablen in Richtung des Nullpunktes, dann wird der Quotient aus Mittelwert und Streuung für diese Variable größer. Wenn der Quotient größer als der vorgegebene Schwellwert ist, wird die Variable im Bedingungsteil der neuen Regel nicht berücksichtigt. Die Bildung der neuen Regeln ist also nicht invariant gegenüber einer Verschiebung der Punkte²⁶.

10.3 Das neue Lernverfahren

In den Anmerkungen zum alten Lernverfahren wurde deutlich, daß das Problem des Umsetzens der geänderten Steuerentscheidungen in neue Regeln bisher vernachlässigt wurde. Mit dem neuen Lernverfahren soll geprüft werden, ob mit der Verwendung von Klassifikationsverfahren dieses Problem gelöst werden kann.

Gleichzeitig wird eine andere Methode vorgestellt, wie für einen schlecht gesteuerten Zustand eine neue Steuerentscheidung bestimmt werden kann.

²⁶Dies ist zum Beispiel interessant, wenn man den Wasserstand in einem Schacht nicht mehr durch die Höhe über der Schachtsohle, sondern durch die Höhe über NN mißt.

10.3.1 Der Strafprozeß

Es existiert weiterhin ein Block mit Metaregeln, in dem während des Inferenzprozesses alle Regeln angewendet werden, deren Bedingungen zu 100% erfüllt sind. Wird eine Metaregel ausgeführt, dann bedeutet das auch weiterhin, daß ein unerwünschter Systemzustand vorliegt. Im Aktionsteil einer Metaregel wird durch eine Variable dieser Zustand beschrieben. Beispiel einer Metaregel:

(SCH2 >R 500)(SCH11 >R 600)(SCH14 >R 600) -> (ENTLA TOO LOW)

Der Bedingungsteil hat die gleiche Bedeutung wie bei den Steuerregeln. Im Aktionsteil sind nur der Operator 'TOO' und die Werte 'HIGH' und 'LOW' erlaubt.

Die Variablen, die im Aktionsteil einer Metaregel auftreten, werden im folgenden als Bewertungsvariablen bezeichnet. Die Metaregeln bewerten die Zustände des Kanalnetzes, nicht aber die Steuerung der einzelnen Steuerelemente. Sie enthalten damit auch nicht mehr das Wissen, wie bei gegebenen Problemen die Steuerung geändert werden soll.

Durch die Auswertung der Metaregeln im Inferenzprozeß wird eine Liste der aktuellen Probleme im Kanalnetz erzeugt. Bsp:

SCH2	TOO	LOW
SCH6	TOO	HIGH
ENTLA	TOO	LOW
⋮	⋮	⋮

Das Wissen, wie die Steuerung verändert werden soll, ist in einer Tabelle enthalten, die die Beziehungen zwischen den Bewertungsvariablen und den Steuervariablen festlegt. In ihr steht, auf welche Art und Weise die einzelnen Steuerelemente zur Lösung der auftretenden Probleme beitragen können. Bsp.:

	P1	P2
SCH1	2	-1
SCH2	0	3
SCH3	1	1

Ein positiver Tabelleneintrag bedeutet, daß die Entscheidungen über die zugehörige Steuervariable zu niedrig²⁷ waren, wenn die Bewertungsvariable den Wert 'LOW' besitzt. Ist der Eintrag negativ, so ist die Beziehung

²⁷Eine höhere Entscheidung bedeutet eine größere Pumpenstufe und damit i.a. eine größere Förderleistung

zwischen Bewertungs- und Steuervariable umgekehrt: Hat die Bewertungsvariable den Wert 'HIGH', dann sollten die Entscheidungen über die Steuervariable nach oben korrigiert werden. Ist der Eintrag Null, so hat das Steuerelement keinen Einfluß auf die Lösung des Problems.

Diese Tabelle wird benutzt, um festzustellen, ob die Leistungen der Pumpen in den vorangegangenen Entscheidungsintervallen zu hoch oder zu niedrig waren und so die Probleme im Kanalnetz verursachten: Sei $B = \{b_1, \dots, b_m\}$ die Menge aller Bewertungsvariablen, die in den Aktionsteilen der Metaregeln auftreten, $\{s_1, \dots, s_M\}$ die Menge der Steuervariablen, deren Regeln durch den Lernprozeß verändert werden sollen. $T = (t_{ji}) \in \mathbb{R}^{m \times M}$ sei die Matrix mit den Einflußwerten. Sei

$$w_i := \sum_j d_j \cdot t_{j,i} \quad i \in (1, \dots, M)$$

mit $J := \{j \mid b_j \text{ trat im Aktionsteil einer angewendeten Metaregel auf}\}$ und $d_j = 1$, wenn b_j zu niedrig, $d_j = -1$, wenn b_j zu hoch.

Für w_i gilt folgendes:

- $w_i > 0 \rightarrow$ die Entscheidungen über s_i waren zu niedrig
- $w_i = 0 \rightarrow$ die Entscheidungen über s_i waren korrekt
- $w_i < 0 \rightarrow$ die Entscheidungen über s_i waren zu hoch

Mit Hilfe des DM werden nun die Regeln bestraft, die in den letzten Intervallen Entscheidungen über die Steuervariablen getroffen haben, deren Werte w_i ungleich 0 sind, und deren Entscheidungen korrigiert werden können. Um zu wissen, wie weit diese Entscheidungen zurückverfolgt werden sollen, existiert eine weitere Tabelle, die angibt, wie weit die Regeln über eine Steuervariable zurückverfolgt werden sollen, wenn diese maßgeblich an der Lösung eines Problem es beteiligt ist. Wird eine Regel bestraft, werden in einer Datei der korrigierte Steuerwert, und die zur Zeit der Anwendung dieser Regeln gültigen Werte der Blockvariablen gespeichert.

Die Strafe für die Regeln über eine Steuervariable richtet sich nach den Werten in T : Unter allen Problemen, zu deren Lösung die Änderung der Entscheidungen für eine Variable s_i beiträgt, wird das Problem b_j bestimmt, für das der Betrag des Wertes t_{ji} maximal ist. Dieser Wert ist der Strafwert.

Eine Pumpe kann gleichzeitig zur Lösung mehrerer Probleme beitragen und andere Probleme verschärfen. Daher wird durch eine Mittel-Zweck Analyse mit Hilfe von T abgewogen, wie die Entscheidungen für die einzelnen Steuervariablen geändert werden sollen, um einen Großteil der Probleme zu lösen. Es erfolgt also eine koordinierte Steuerungsänderung.

10.3.2 Entwicklung neuer Regeln

Sei ein Steuerelement s_i $i \in (1, \dots, M)$ und eine Entscheidung e_{ij} $j \in (1, \dots, n_i)$ gewählt und $S = Z(i, j)$ die Menge aller Kanalnetzzustände, bei denen der Steuervariablen s_i durch das Lernverfahren die Entscheidung e_{ij} zugeordnet wurde. Diese Zustände werden durch die Blockvariablen des Blockes, in dem die Regeln über s_i stehen, charakterisiert. Für das neue Lernverfahren wird vorausgesetzt, daß alle Variablen, die im Lernprozeß zur Beschreibung eines Zustandes dienen, quantitative Werte besitzen (Insbesondere also die Blockvariablen). Dies ist aber keine Einschränkung, da die Beobachtungs-, Stör- und Steuergrößen sämtlich quantitative Größen sind.

Die Umsetzung der Elemente aus S in neue Regeln soll mit Hilfe von Klassifikationsverfahren präziser geschehen als bisher. Ein Verfahren, welches S in Regeln umsetzt, soll folgende Eigenschaften besitzen (wobei die Elemente aus S wieder als Punkte im \mathbb{R}^p interpretiert werden):

Das Verfahren soll die Bereiche mit hoher Punktdichte finden (natürliche Gruppen). Es muß unabhängig von der Form dieser natürlichen Gruppen sein und es muß diese Gruppen auch hinreichend genau durch Regeln beschreiben. Wenn Bereiche mit geringer Punktdichte vorhanden sind, dann kann es sein, daß die Steuerung dieser Bereiche teilweise gut und teilweise schlecht war. Die Ursache können unzureichende Fakten über den Zustand des Netzes sein (zwei Zustände, die das RBS als gleich ansieht, sind evtl. grundverschieden). Um also gewisse Ungenauigkeiten auszugleichen, soll das Verfahren Gebiete mit geringerer Punktdichte ausschließen.

Um diese Forderungen zu erfüllen, werden die Punkte mit dem Verfahren von Wishart zur Analyse der absoluten Punktdichte in Zusammenhangskomponenten zerlegt:

- a Für alle $x_k \in S$ wird die Anzahl c_k der Punkte $x_i \in S$ berechnet, die in der d -Umgebung von x_k liegen, d.h. für die gilt: $d_{k,i} < d$.
- b Alle Punkte x_k , für die $c_k < c$ gilt, werden aus S entfernt. Es werden also nur noch die Punkte betrachtet, die in einem Bereich mit einer Mindestpunktdichte liegen.
- c Mit den restlichen Punkten aus S (mit $c_k \geq c$) wird mit der Single-Linkage- Methode eine Gruppierung der Stufe d erzeugt.

Seien $x, y \in S$ mit $x = (x_1, \dots, x_p), y = (y_1, \dots, y_p)$ gegeben. Die Distanz $d(x, y)$ zweier Punkte (Kanalnetzzustände) wird wie folgt berechnet²⁸:

²⁸Siehe [12] Seite 212-213

$$d(x, y) = \left(\frac{\sum_{i=1}^p w_i (x_i - y_i)^2}{\sum_{i=1}^p w_i} \right)^{1/2} \quad w_i \in \mathbb{R}_+, \sum_{i=1}^p w_i \neq 0$$

Die Gewichte w_i für die einzelnen Blockvariablen werden vom Benutzer bestimmt. Der Fachmann, der die Daten für den Lernprozeß erstellt, kann mit diesen Gewichten den Einfluß verschiedener Blockvariablen steuern. Auf eine Normierung der Daten wurde verzichtet, da nicht abgeschätzt werden kann, welchen Einfluß diese Normierung auf die Daten besitzt. Die verschiedenen Blockvariablen sind zwar nicht unbedingt direkt vergleichbar (z.B. Wasserstände \leftrightarrow Durchflußvolumina), aber durch die Gewichte kann der Benutzer hier einschreiten. Das Verfahren von Wishart benutzt zwei Parameter: Zum einen wird eine Mindestpunktdichte c vorgegeben, sowie eine Distanzschranke d . Auch mit diesen Werten kann die Neubildung von Regeln beeinflusst werden.

Die Analyse der Punktdichte liefert eine Liste von Zusammenhangskomponenten. Sei eine einzelne Zusammenhangskomponente (ZK) gewählt. Es wird das kleinste Mehrfachintervall betrachtet, welches alle Punkte der ZK enthält. Dieses Mehrfachintervall (Quader) wird in kleinere Quader zerlegt. Für jeden dieser kleineren Quader wird geprüft, ob ein Punkt der ZK in diesem Quader liegt. Es werden nur noch die Quader betrachtet, die einen Punkt der ZK enthalten. Danach wird versucht, die Quader zu größeren Quadern zusammenzufassen. Dies geschieht, da jeder Quader zur Bildung einer Regel führt.

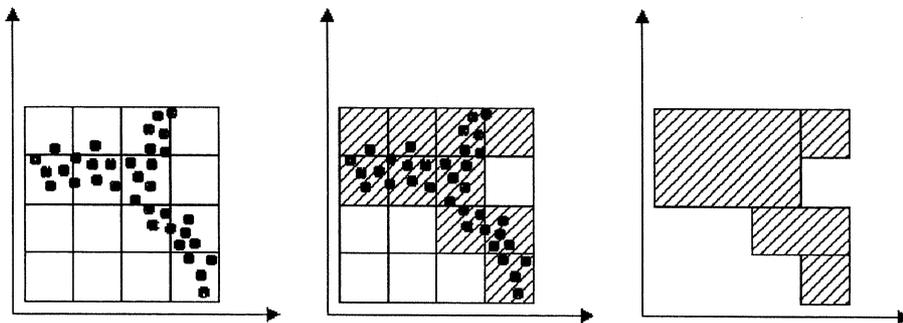


Abb. 12: Beispiel für die Beschreibung einer Zusammenhangskomponente durch Quader

Ist dieser Vorgang beendet, erhält man eine Liste von Quadern, welche die ZK ausreichend genau beschreiben. Jeder der Quader aus der Liste bildet den Geltungsbereich einer neuen Regel.

10.4 Anmerkungen zu den beiden Lernverfahren

Es soll jetzt anhand eines Beispiels verdeutlicht werden, wie sich durch den Einsatz des Klassifikationsverfahrens die Umsetzung der Steuerungsänderung in neue Regeln gegenüber dem alten Lernverfahren verbessert hat. Dazu wurde die in der Anlage 3 angegebene Basisregelmenge zur selbständigen Entwicklung einer Steuerstrategie benutzt. Für die Steuervariable P3 wurden als Blockvariablen SCH6 und SCH11 gewählt. Dann wurde mit dem neuen Lernverfahren einmal das erste und einmal das dritte Niederschlagsereignis simuliert. Durch den Bestrafungsprozeß wurde eine Menge von Zuständen (Punkten) erzeugt, denen die neue Steuerentscheidung 'P3 =R 1' zugeordnet wurde.

Es wird jetzt (graphisch) dargestellt, wie die beiden Lernverfahren diese Zustände handhaben. Die Zustände werden als Punkte des \mathbb{R}^2 interpretiert.

Bei einem Maximalwert von 0.1 für den Quotienten aus Mittelwert und Streuung wurde von dem alten Verfahren keine neue Regel erzeugt, da bei beiden Variablen der Quotient zu groß war. Nach Erhöhung des Schwellwertes auf 0.4 wurden beide Blockvariablen in den Bedingungsteil einer neuen Regel aufgenommen. Der Bereich, den diese Regel durch ihren Bedingungsteil beschreibt, ist in in Abbildung 13 durch ein Rechteck gekennzeichnet.

Zustände mit der Entscheidung P3 =R 1

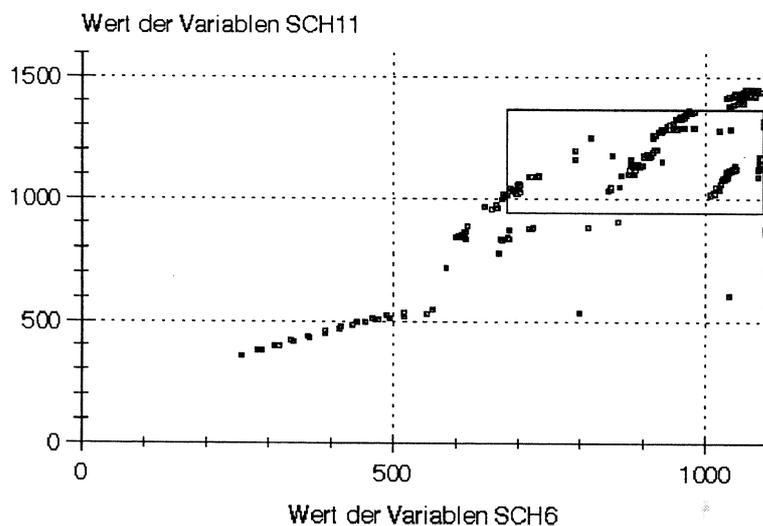


Abb. 13: Klassifikation der Punkte durch das alte Lernverfahren.

Deutlich erkennbar ist, daß mehrere Bereiche mit höherer Punktdichte nicht durch die neue Regel erfaßt werden.

In die Graphiken für das neue Lernverfahren sind nur die kleinsten Rechtecke eingezeichnet, die die Zusammenhangskomponenten enthalten.

Dies geschieht aus Gründen der Übersicht. Die tatsächlich erzeugten Regeln beschreiben die Bereiche hoher Punktdichte noch genauer, als die in den Graphiken eingezeichneten Rechtecke.

Für das neue Lernverfahren wurden drei verschiedene Parameter-Kombinationen getestet:

Zuerst wurde die Kombination Dichte $c = 3$ und Distanzschranke $d = 60$ untersucht. Diese Kombination wurde auch in den späteren Simulationen mit dem neuen Lernverfahren verwendet.

Zustände mit der Entscheidung P3 = R 1

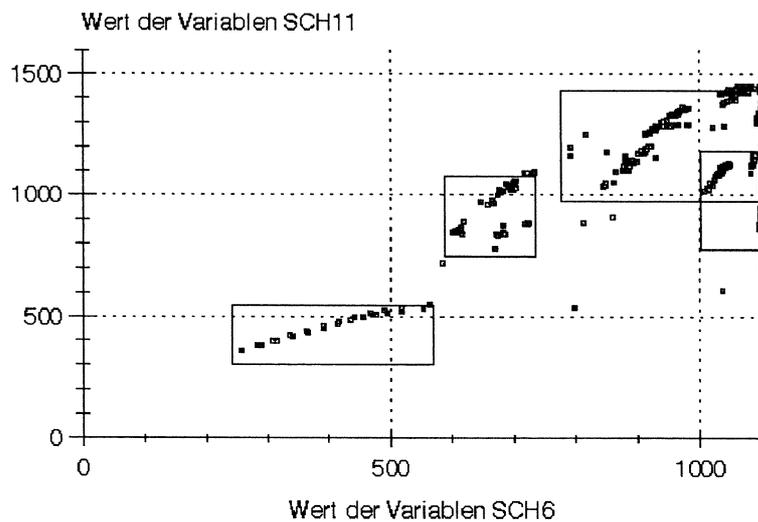


Abb. 14: Klassifikation der Punkte durch das Verfahren von Wishart. Dichte = 3 ,
Distanzschranke = 60

Deutlich zu sehen ist, daß einige "Ausreißer" nicht klassifiziert werden.

Eine Verringerung der Distanzschranke (auf $d = 10$) bewirkt laut Wishart eine Vergrößerung der Anzahl von Zusammenhangskomponenten. Dies ist

auch bei der untersuchten Punktmenge zu beobachten.

Zustände mit der Entscheidung $P3 = R 1$

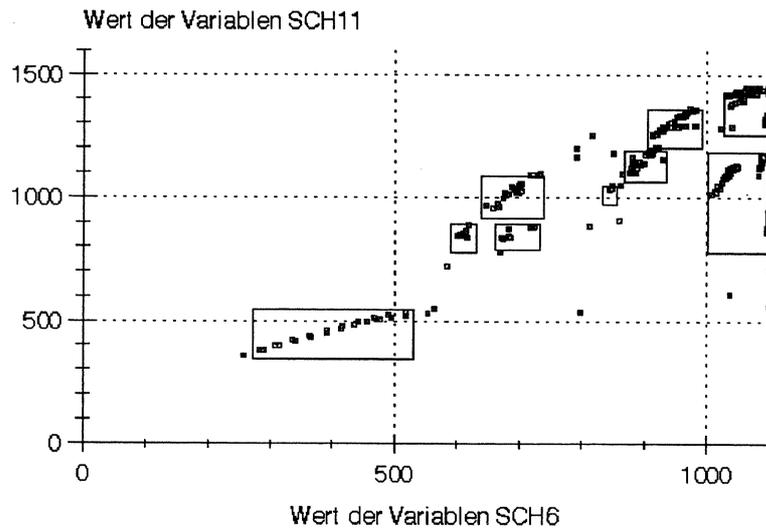


Abb. 15: Klassifikation der Punkte durch das Verfahren von Wishart. Dichte = 3 ,
Distanzschranke = 30

Schließlich wurde noch die Kombination $c = 6$ und $d = 60$ untersucht. Durch eine Vergrößerung des Parameters c wird der Verkettungseffekt stärker berücksichtigt (und vermieden). Da aber die Zusammenhangskomponenten gut separiert sind, vergrößert sich ihre Anzahl nicht. Es werden jedoch mehr Randpunkte von Zusammenhangskomponenten ausgeschlos-

sen.

Zustände mit der Entscheidung P3 = R 1

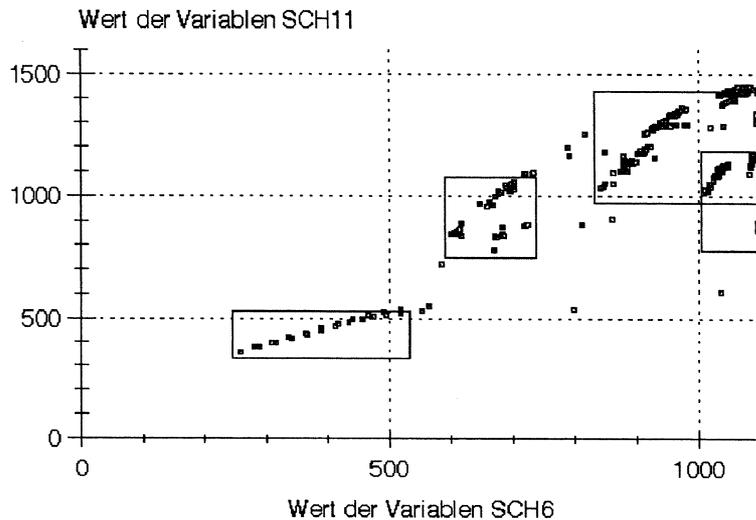


Abb. 16: Klassifikation der Punkte durch das Verfahren von Wishart. Dichte = 6 ,
Distanzschranke = 60

Als Fazit dieser Betrachtung läßt sich sagen, daß der Einsatz des Verfahrens von Wishart genau das bewirkt, was von ihm erwartet wird. Die Änderung der Steuerentscheidungen wird wesentlich besser durch neue Regeln beschrieben. Der Nachteil dieser Methode ist, daß gegenüber dem alten Lernverfahren wesentlich mehr neue Regeln innerhalb eines Lernschrittes erzeugt werden. Dadurch stößt man schneller an die Grenzen des Regelspeichers²⁹.

In beiden Lernverfahren sind das Erkennen schlecht gesteuerter Zustände und das Zuordnen neuer Entscheidungen eng miteinander verbunden. Bei dem bisherigen Verfahren steht schon in der Metaregel, welches Steuerelement anders gesteuert werden soll, um in Zukunft den unerwünschten Zustand zu vermeiden. Gleichzeitig wird auch angegeben, wie diese Änderung erfolgen soll. Indem durch eine Metaregel die Steuerung genau eines Steuerelementes geändert wird, wird versucht, jedes Problem für sich durch eine Veränderung der Steuerung zu beheben. Treten mehrere verschiedene Probleme gleichzeitig auf, werden also in einem Inferenzprozeß mehrere Metaregeln angewendet, so erfolgt keine koordinierte Änderung der Steuerstrategie³⁰. Um diesen Aspekt zu berücksichtigen, dienen die Metaregeln im neuen Lernverfahren nur noch der Erkennung der Probleme im Kanalnetz. Nach der Untersuchung aller Metaregeln, werden die Einflüsse

²⁹Wie sich später zeigt, kann das auch ein Vorteil sein

³⁰Es kann daher vorkommen, daß die Lösung eines Problems ein anderes Problem verschärft.

der Steuerelemente auf die einzelnen Probleme analysiert, um festzustellen, wie die bisherige Steuerstrategie geändert werden muß, um die Probleme zu vermeiden.

Die Änderung der Steuerentscheidungen für bestimmte Zustände erfolgt bei beiden Verfahren unter Verwendung von speziellem Wissen über die Zusammenhänge im Kanalnetz. Ein Versuch, mit Methoden der automatischen Klassifikation die Änderung der Steuerstrategie zu bestimmen, ist folgender:

Die Zustände im Kanalnetz seien charakterisiert durch die Werte der Beobachtungsvariablen. Die Zustände werden so in Klassen eingeteilt, daß allen Objekten einer Klasse dieselbe Steuerentscheidung zugeordnet ist und zwei Objekten aus verschiedenen Klassen auch verschiedene Steuerentscheidungen zugeordnet sind. Wird durch die Bewertung der Steuerung festgestellt, daß ein Objekt einer anderen Klasse zugeordnet werden muß, dann liegt das Problem darin, diese neue Klasse zu bestimmen. In der Praxis kann das Problem so gelöst werden:

Während einer Simulation werden nicht nur Zustände gespeichert, die schlecht gesteuert wurden, sondern auch die Zustände, deren Steuerung als gut bewertet wurde.

Sei S die Menge aller Zustände, für die eine schlechte Steuerentscheidung getroffen wurde und G die Menge aller "gut" gesteuerten Zustände. Sei d ein Maß für die Distanz zwischen zwei Zuständen (Objekten). Für jedes $x \in S$ wird das Element $y \in G$ gesucht, für das $d(x, y)$ minimal ist. Dem Zustand x wird dann die Steuerentscheidung von y zugeordnet³¹.

Dieses Verfahren führt aber nur dann zu einer Verbesserung der Steuerung, wenn gewährleistet ist, daß das gesuchte Objekt $y \in G$ eine Entscheidung besitzt, die, auf x übertragen, zu einer Verbesserung der Steuerleistung führt. Es kann aber vorkommen, daß in G kein Objekt existiert, dem die für x optimale Steuerentscheidung zugeordnet ist. Dies ist zum Beispiel der Fall, wenn allen Objekten aus G die gleiche Steuerentscheidung zugeordnet ist wie dem Objekt x . Dieses Verfahren führt also zu einer anderen, aber nicht unbedingt besseren Steuerung.

Ein weiterer Bereich des Lernens ist der Übergang von Zuständen mit ihren geänderten Steuerentscheidungen zu neuen Regeln. In der derzeitigen Form des RBS werden durch die Bedingungssteile der Regeln Mehrfachintervalle beschrieben. D.h., eine Menge von Zuständen (Punkten) mit der gleichen Steuerentscheidung muß durch einen oder mehrere mehrdimensionale Quader $Q = ([x_1, y_1] \times [x_2, y_2] \times \dots \times [x_p, y_p])$ ($p =$ Anzahl der Blockvariablen des aktuellen Blockes) beschrieben werden. Diese Quader können dann in Regeln umgesetzt werden.

³¹Siehe Single-Linkage Verfahren zur hierarchischen Klassifikation

Das bisherige Verfahren, durch Mittelwerte und Streuungen einen Quader zur näherungsweise Beschreibung der Punktmenge zu konstruieren, ist sehr einfach, aber häufig ungenau. Bei einer kugelförmigen oder elliptischen Form der betrachteten Punktmenge kann es ordentliche Ergebnisse liefern, nicht aber für langgestreckten, verzweigte oder nicht zusammenhängende Punktmenge. Da in den Bedingungsteil nur Variablen aufgenommen werden, für die der Quotient aus Streuung und Mittelwert kleiner als ein vorgegebener Grenzwert ist, kann es passieren, daß einzelne (oder sogar alle) Variablen nicht berücksichtigt werden. In diesem Fall wird die Steuerung in viel größerem Umfang geändert, als durch den Bestrafungsprozeß gefordert wurde.

Im neuen Lernverfahren wird die Punktmenge zuerst in Zusammenhangskomponenten (ZK) zerlegt, die dann durch mehrere Quader (folglich auch durch mehrere neue Regeln) beschrieben wird. Es hat sich gezeigt, daß die Darstellung der ZK durch eine Menge von Quadern zwar sehr genau sein kann (je größer n , desto genauer), aber sehr aufwendig in der Berechnung ist und sehr viel Speicherplatz (im Regelspeicher) verbraucht. Dies ist allerdings nur eine Frage der Hardware, mit der gearbeitet wird.

Aufgrund der bisherigen Betrachtungen wäre es günstig, eine Beschreibung für die Punktmenge zu finden, die die Struktur der Menge hinreichend genau wiedergibt und die sich durch wenige Regeln ausdrücken läßt. Eine Alternative zu der oben genannten Verwendung von Quadern wäre die Einführung trennender Hyperebenen:

Die Zustände, denen eine neue Steuerentscheidung zugeordnet wurde, seien derart in Klassen eingeteilt, daß allen Elementen einer Klasse die gleiche Steuerentscheidung zugeordnet ist. Diese Klassen lassen sich dann weiter zerlegen in Zusammenhangskomponenten. Für je zwei ZK wird dann eine Hyperebene konstruiert, die diese beiden ZK trennt [11].

Da Hyperebenen den Zustandsraum in zwei Bereiche teilen, kann im STM des RBS als Faktum angegeben werden, auf welcher Seite einer Hyperebene sich ein Objekt befindet. Dadurch kann die Zugehörigkeit eines Objektes zu einer Klasse durch seine Lage bezüglich der trennenden Hyperebenen geprüft werden. Es kann also auf eine genaue Darstellung der Form einer

ZK verzichtet werden.

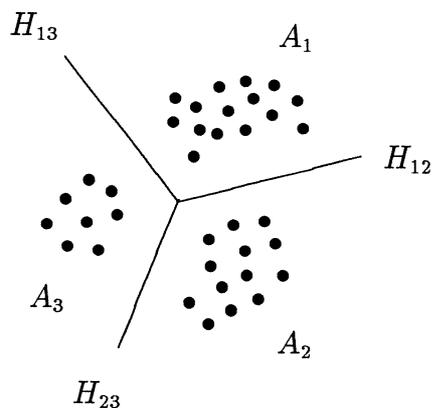


Abb. 17: Beispiel für die Trennung dreier Zusammenhangskomponenten durch Hyperebenen

Dieses Verfahren funktioniert leider nur dann, wenn alle ZK kompakt sind. Bei verzweigten ZK lassen sich zwei Klassen evtl. nicht durch eine Hyperebene trennen. Will man dann zwei Klassen durch mehrere Hyperebenen trennen, muß man wieder die spezielle Form der ZK berücksichtigen. Außerdem ist es sehr schwierig, die einzelnen Hyperebenen zu bestimmen. Daher wurde dieser Ansatz nicht weiter verfolgt.

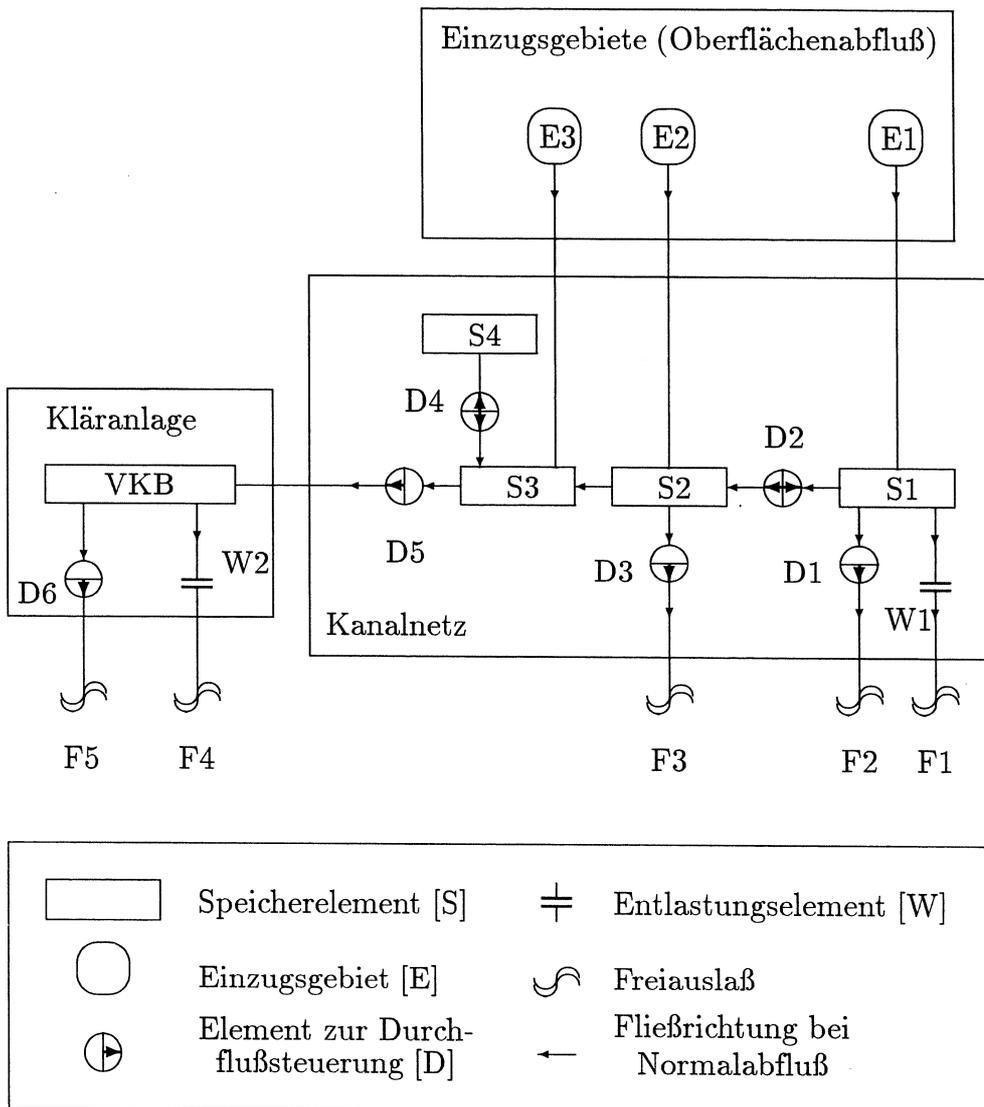


Abb. 18: Allgemeine Kanalnetzdarstellung

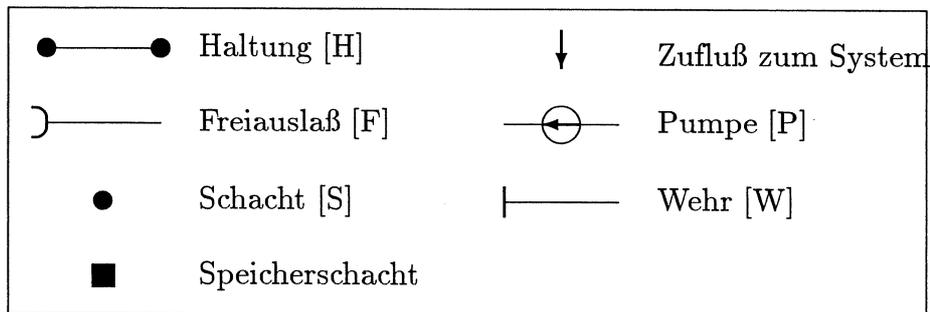
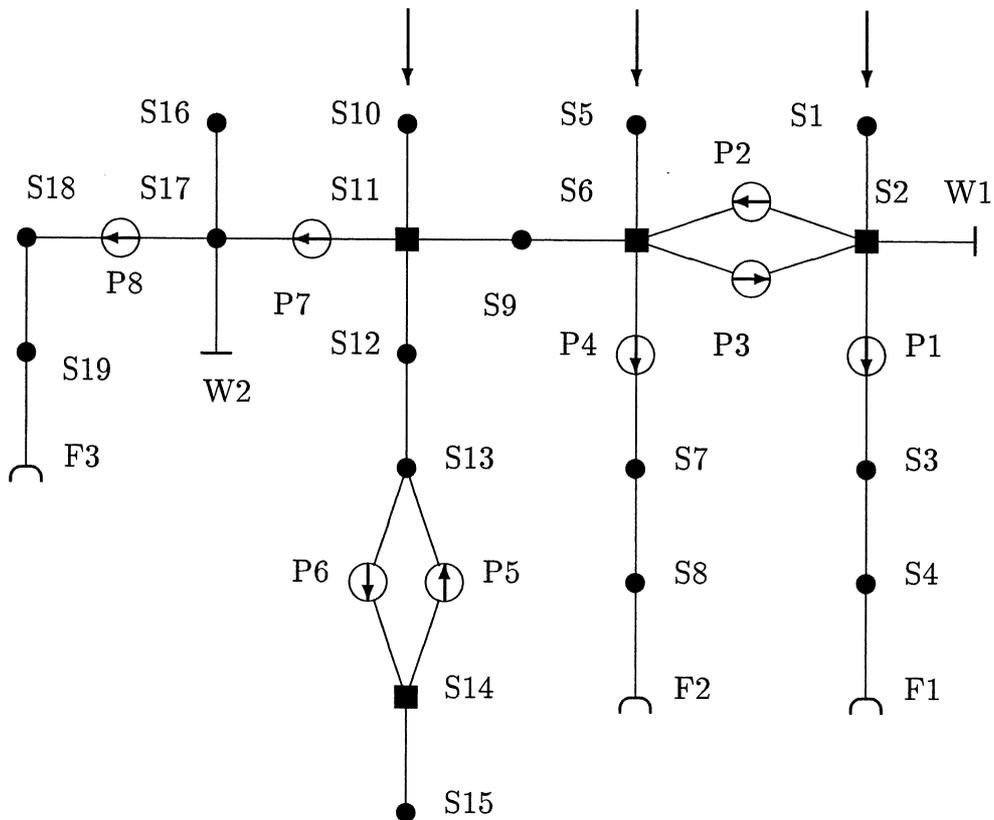


Abb. 19: EXTRAN-Darstellung des Kanalnetzes

11 Simulationen

Durch die folgenden Simulationen soll versucht werden, die beiden Lernverfahren in der Praxis zu beobachten.

11.0.1 Das Kanalnetz

Das Kanalnetz ist ein fiktives, für diese Arbeit entworfenes Kanalnetz. Abbildung 18 zeigt eine allgemeine Darstellung des Kanalnetzes. EXTRAN benutzt eine spezielle Beschreibung des Netzes, sie wird in Abbildung 19 dargestellt³². Auf diese EXTRAN-Darstellung beziehen sich die in der Diplomarbeit benutzten Namen und Numerierungen von Bauwerken des Kanalnetzes.

Das Kanalnetz wurde so entworfen, daß möglichst flexibel auf Niederschlagsereignisse reagiert werden kann. Es gibt sowohl aktive als auch passive Bauwerke zur Entlastung an mehreren Stellen im Kanalnetz (Pumpen, Wehre), um bei einer Überlastung des Netzes das Überstauen von Schächten vermeiden zu können. Bis auf Pumpe 8 sind alle Pumpen so dimensioniert, daß sie innerhalb von 20 - 30 Minuten den angrenzenden Speicherschacht füllen bzw. entleeren können. Damit ist gewährleistet, daß der Abwassertransport sehr flexibel gesteuert werden kann. Pumpe 8 fördert auf der höchsten Stufe gerade die doppelte Menge des Trockenwetterzuflusses. Dadurch soll simuliert werden, daß über die Kläranlage auch bei starker Belastung keine großen Wassermengen aus dem System fließen sollen, um die Klärleistung zu erhalten. Das Kanalnetz hat ein Gesamtspeichervolumen von ca. 32.000 m³. Davon entfallen ca. 27000 m³ auf fünf Speicherschächte, welche ein annähernd gleiches Speichervolumen haben. Es gibt daher im Netz genügend Speicherkapazität, mit deren Ausnutzung eine Verbesserung der Steuerleistung erreicht werden kann.

11.0.2 Die Kommunikation EXTRAN ↔ RBS

Die dem RBS zur Verfügung stehenden Informationen über die Situationen im Kanalnetz sowie die Informationen, die vom RBS zur Simulation übertragen werden sollen, werden in Kommunikationsdateien festgelegt. Sie sind in der Anlage angegeben.

³²Anlage 1 enthält die Kanalnetzdatei, wie sie von EXTRAN benutzt wird.

11.0.3 Die Regenereignisse

Die Regenereignisse sollten so gewählt werden, daß die Steuerleistung des RBS für diese Ereignisse durch einen Lernprozeß verbessert werden kann.

Die Ereignisse sollten daher das Kanalnetz nicht global überlasten. Für das Kanalnetz wird eine gleichmäßige Überregnung angenommen. Aufgrund der baulichen Gegebenheiten des Netzes (der Durchfluß zwischen den Speicherschächten 'Schacht-06' und 'Schacht-11' kann nicht gesteuert werden, und ohne Steuerung wird die Kapazität des Speicherschachtes SCH14 nicht genutzt), werden auch bei der gleichmäßigen Überregnung der Einzugsgebiete ohne Steuerung die Kapazitäten des Netzes optimal genutzt. Es wurden drei Regenereignisse konstruiert:

Nr.	Regensumme [mm]	Dauer [min]	Zuflußmenge zum Netz [m ³]
1	18,4	45	30759
2	13,1	20	18987
3	18,4	60	29339

Die zufließende Regenwassermenge liegt bei allen Regenereignissen unter der Speicherkapazität des gesamten Netzes. Aufgrund dieser Überlegungen sollten die konstruierten Regenereignisse für einen Lernprozeß geeignet sein.

Da bei dem durch die Metaregeln vorgegebenen Lernziel das zweite Regenereignis nur einen geringen Einfluß auf das Lernen hat³³, werden bei den folgenden Simulationen nur die Regenereignisse 1 und 3 verwendet.

³³Bei anderen Kanalnetzen und/oder anderen Lernzielen können schwache Regenereignisse durchaus eine größere Bedeutung besitzen.

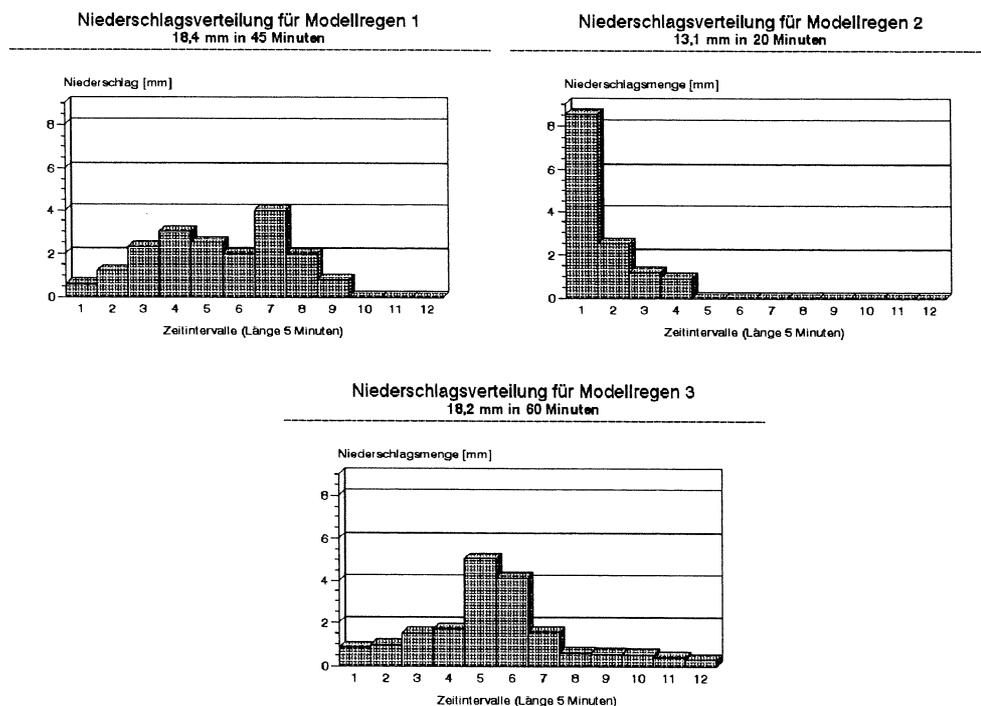


Abb. 20: Darstellung der Regenereignisse

11.0.4 Die Ausgangsregelmengen

Es wurden zwei Ausgangssteuerregelmengen entworfen: Die erste Regelmenge steuert den Abwassertransport nach lokalen Gesichtspunkten. Diese Steuerung ignoriert den Speicherschacht 'Schacht-14'. Die Pumpen zum Befüllen und Entleeren des Schachtes werden ständig auf der niedrigsten Stufe gesteuert. Durch einen Lernprozeß soll dabei gelernt werden, diesen Schacht zu nutzen. Es existieren außerdem keine Regeln zur Entlastung durch die Pumpen 1 und 4.

Die zweite Regelmenge steuert nur den Trockenwetterabfluß, d.h. alle Pumpen werden nur auf der untersten Stufe betrieben. Anhand dieser Regelmenge soll getestet werden, ob die Lernverfahren dazu geeignet sind, selbständig eine Steuerstrategie zu entwickeln.

Die Metaregeln für beide Verfahren sind darauf ausgerichtet, ein Überstauen der Speicherschächte zu vermeiden. Sind die Speicherschächte 2,11 und 14 fast voll (Auslastung mehr als 90%), dann (und nur dann) soll Wasser ungeklärt entlastet werden. Insgesamt soll durch einen Lernprozeß die Nutzung der Kapazitäten des Kanalnetzes bei einer Auslastung des Netzes verbessert werden. Es interessiert also der Zeitraum, in dem ein Regenwasserzufluß stattfindet. Die Metaregeln sind nicht darauf ausgerichtet, die Entleerung des Kanalnetzes nach einem Regenereignis zu lernen³⁴.

³⁴Dies hat vor allem technische Gründe, da besonders durch das alte Lernverfahren bei

Alle übrigen, für die Lernverfahren wichtigen Dateien werden in der Anlage aufgeführt.

11.0.5 Simulationen mit dem alten Lernverfahren

Ein Lernprozeß umfaßt mehrere Lernschritte. Ein Lernschritt besteht aus der Simulation der Regenerereignisse 1 und 3 und einer anschließenden Neubildung von Regeln.

Die Lernerfolge mit dem alten Verfahren lassen sich schnell beschreiben: Bei der ersten Anfangsregelmenge wurden 4 neue Regeln erzeugt, die sehr selten (1-2 mal in 40 Simulationen !) angewendet wurden. Nach wenigen Lernschritten kamen ausschließlich die Regeln der Anfangsregelmenge zur Anwendung. Daher ist keine Verbesserung der Steuerung zu verzeichnen. Bei der zweiten Anfangsregelmenge wurden mehr neue Regeln erzeugt als bei der ersten, aber wiederum wurden die neuen Regeln äußerst selten angewendet und es ist kein Lernerfolg erkennbar. Insbesondere wurde das Speicherbecken 14 nicht befüllt.

Die neuen Steuerregeln werden selten angewendet, weil das alte Lernverfahren den Wert

$\text{Alter} \cdot \text{Anwendungshäufigkeit} / \text{Strafwert}$

als Kriterium zur Auswahl einer Regel eines Blockes benutzt. Ältere, häufig benutzte Regeln werden dadurch zu stark bevorzugt. Da außerdem nur wenige Regeln erzeugt werden und damit der Regelspeicher nicht überlastet wird, werden auch keine Regeln gelöscht. Die Methode zur Regelauswahl kann man als sehr konservativ bezeichnen.

11.0.6 Simulationen mit dem neuen Lernverfahren

Für das neue Lernverfahren wurde auch die Regelauswahl modifiziert: Sind die Bedingungen mehrerer Regeln gleich gut erfüllt, dann entscheidet nur der Strafwert darüber, welche Regel angewendet wird. Die Methode der Regelauswahl ist wesentlich progressiver als die bisherige. Neugeschaffene Regeln werden bevorzugt, denn diese haben noch keine oder erst sehr wenige Strafpunkte erhalten. Da das neue Verfahren durchschnittlich mehr Regeln pro Lernschritt erzeugt, passiert es häufiger, daß der Regelspeicher voll ist und eine neue Regel nicht mehr eingetragen werden kann. In diesem Fall werden einige Regeln gelöscht, um Platz für neue Regeln zu schaffen. Das bedeutet, daß bei dem neuen Verfahren die Fluktuation innerhalb der Regelmenge sehr groß ist.

einer großen Anzahl von Metaregeln Dateien mit einer Größe von mehreren Megabyte entstehen.

Nun zu den Simulationsergebnissen. Zuerst werden die Ergebnisse für die Ausgangsregelmenge 1 (Verbesserung einer vorhandenen Strategie) untersucht:

Eine Teilaufgabe war, die Speicherkapazität des Schachtes 14 auszunutzen.

Speicherung von Abwasser in Schacht 14 Verbesserung einer lokalen Strategie

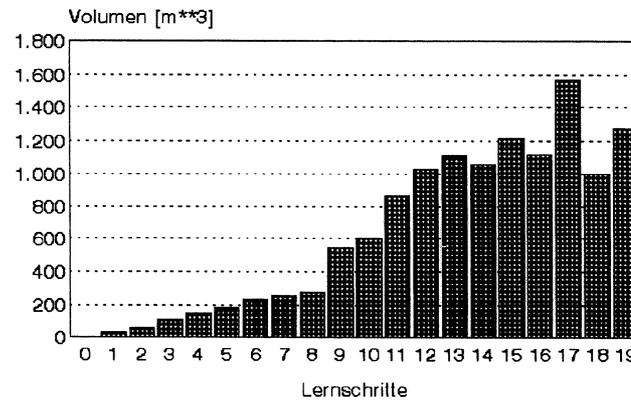
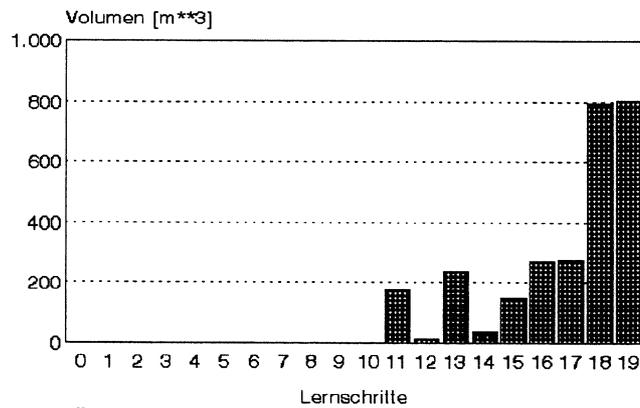


Abb. 21: Ergebnisse für Modellregen 1, erzielt mit Anfangsregelmenge 1

Wie die Graphik verdeutlicht, wächst die Wassermenge, die während eines Regenereignisses in dem Schacht gespeichert wird, fast monoton. Dies wurde dadurch begünstigt, daß schon sehr früh die Regel '(DUMMY) =R 0 -> (P6 =R 0)' aus dem Regelspeicher entfernt wurde. Schacht14 wurde nur befüllt, aber nicht entleert. Aufgrund der Metaregeln (Vermeidung von Überstau,...) sind die Menge der Entlastungen über Wehr 1 und das Überstauverhalten von Schacht 11 nützliche Informationen, um den Lernerfolg zu untersuchen:

Entlastung von Abwasser über Wehr 1 Verbesserung einer lokalen Steuerung



Überstauverhalten von Schacht 11 Verbesserung einer lokalen Steuerung

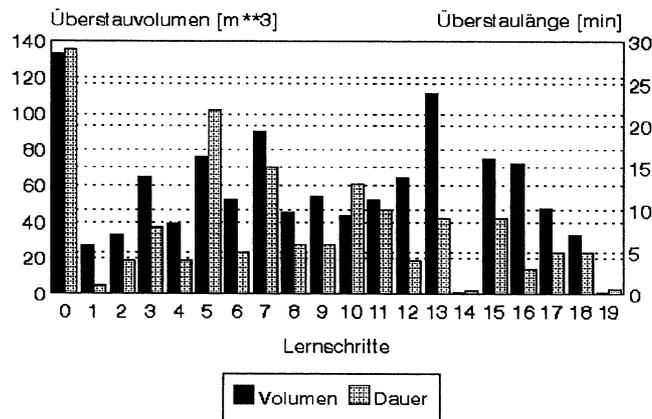
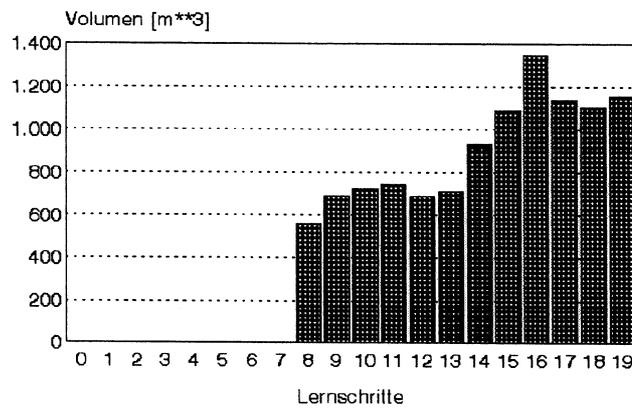


Abb. 22: Ergebnisse für Modellregen 1, erzielt mit Anfangsregelmenge 1

Diese Graphiken zeigen, daß keine kontinuierliche Verbesserung der Steuerung eintritt. Dies liegt vor allem an der großen Fluktuation in der Regelmenge. Innerhalb eines Lernschrittes wurden bis zu 20% aller Regeln gelöscht. Für die Entwicklung einer Steuerstrategie mit der zweiten Anfangsregelmenge ergibt sich ein ähnliches Bild:

Speicherung von Abwasser in Schacht 14 Entwicklung einer Steuerung



Überstauverhalten von Schacht 11 Entwicklung einer Steuerung

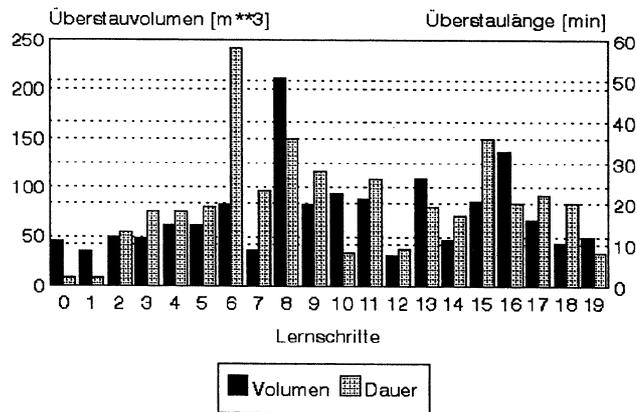


Abb. 23: Ergebnisse für Modellregen 1, erzielt mit Anfangsregelmenge 2

Entlastung von Abwasser über Wehr 1 Entwicklung einer Steuerung

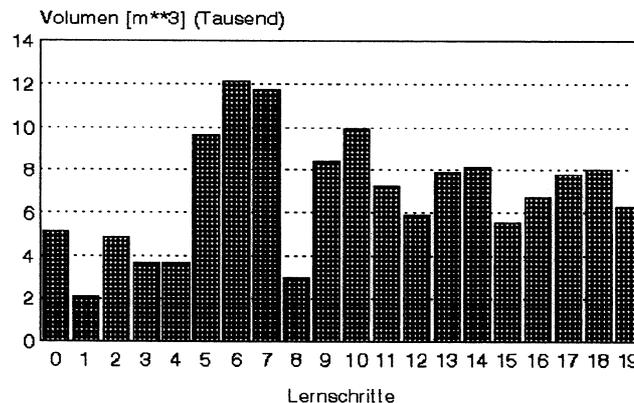


Abb. 24: Ergebnisse für Modellregen 1, erzielt mit Anfangsregelmenge 2

Das RBS lernt, Schacht 14 zu befüllen. Die Befüllung setzt erst nach mehreren Lernschritten ein, da die Regel

$(DUMMY =R 0) \rightarrow (P6 =R 0)$

erst so spät gelöscht wird. Ansonsten ist auch hier keine Kontinuität in der Steuerleistung zu erkennen.

Bei der Version des RBS mit neuen Lernverfahren gelten die meisten Regeln nur für einen sehr kleinen Bereich des Zustandsraumes. Dies bewirkt, daß viele Regeln nur selten angewendet werden und somit häufig ältere Regeln gelöscht werden. Wie sich bei der Befüllung von Schacht 14 gezeigt hat, kann dies positive Effekte haben, aber insgesamt werden viel zu viele Regeln gelöscht, um einen dauerhaften Lernerfolg zu erzielen.

12 Zusammenfassung

Die Steuerung von Kanalnetzen gewinnt in letzter Zeit immer mehr an Bedeutung. Dies liegt daran, daß die Anforderungen an den Abwassertransport ständig steigen. Die Steuerung eines Kanalnetzes ist eine zeitkritische und bei komplexen Kanalnetzen auch schwierige Angelegenheit. Deshalb wird am Institut für Wasserwirtschaft der Universität Hannover versucht, durch den Einsatz eines regelbasierten Systems die Steuerung zu automatisieren oder dem Fachmann Empfehlungen zur Steuerung zu geben. Das regelbasierte System trifft die Entscheidungen über die Steuerelemente eines Netzes mit Hilfe von Informationen über den Kanalnetzzustand und einer Menge von Regeln, die das Wissen über die Steuerung enthalten. Durch Anwendung von Regeln werden die Steuerentscheidungen getroffen. Da nicht immer die optimale Steuerstrategie bekannt ist, kann man das RBS

durch ein Lernmodul zu erweitern. Das RBS soll damit selbständig durch Beobachtung und Bewertung seiner Steuerstrategie neues Wissen über die Kanalnetzsteuerung sammeln und dadurch seine Steuerleistung zu verbessern.

Lernen bedeutet, daß das System für bestimmte Kanalnetzzustände neue Steuerentscheidungen trifft und diese Änderungen der Steuerstrategie durch neue Regeln ausdrückt.

Das Lernen läßt sich in die Schritte

1. Bewertung der Steuerung
2. Änderung von Steuerentscheidungen für bestimmte Kanalnetzzustände
3. Erzeugen neuer Regeln, die diese Änderung ausdrücken.

unterteilen. Der dritte Schritt bedeutet, daß man für eine Menge von Kanalnetzzuständen, denen neue Steuerentscheidungen zugeordnet wurden, neue Regeln schafft, die genau dann angewendet werden, wenn einer dieser Zustände eintritt. Die Menge der Zustände ist zunächst ungeordnet. Daher ist es schwierig, diese Zustände durch Regeln zu erfassen. Durch die Verwendung von Klassifikationsverfahren kann die Menge so in mehrere kleinere Teilmengen zerlegt werden, daß sich die einzelnen Teilmengen leichter durch Regeln beschreiben lassen.

Es wurde ein Lernverfahren entwickelt, welches eine Klassifizierung der Kanalnetzzustände, denen eine geänderte Steuerentscheidung zugeordnet wurde, durchführt, bevor diese in neue Regeln umgesetzt werden. Dieses neue Verfahren erlaubt eine sehr genaue Darstellung der Änderung der Steuerstrategie durch neue Regeln. Dazu ist jedoch eine große Anzahl von Regeln notwendig.

Die Leistung des Lernverfahrens wurde durch Simulation von Niederschlagsereignissen für ein fiktives Kanalnetz getestet. Dabei zeigte sich, daß andere Teile des Lernmoduls noch zu große Mängel aufweisen, um das Lernverfahren praktisch anzuwenden. Diese Mängel liegen vor allem in der großen Fluktuation in der Regelmenge. Diese könnte aber durch eine wesentliche Vergrößerung des Regelspeichers (bei geeigneter Hard- und Software) verringert werden. Hier sind also noch weitere Untersuchungen nötig, um die Probleme zu lösen.

Es wurde auch untersucht, ob mit Klassifikationsverfahren neue Steuerentscheidungen für schlecht gesteuerte Zustände gefunden werden können. Dies erwies sich jedoch als unbrauchbar: Durch diese Verfahren wird vorhandenes Wissen über die Zusammenhänge in einem Kanalnetz nicht

berücksichtigt und es ist nicht erkennbar, daß die mittels Klassifikation erzeugte Änderung der Steuerstrategie tatsächlich eine Verbesserung der Steuerleistung bewirkt.

Klassifikationsverfahren sind also nicht geeignet, Aufgaben zu lösen, für die fachspezifisches Wissen benötigt wird. Es hat sich aber herausgestellt, daß Klassifikation ein geeignetes Hilfsmittel ist, um die Handhabung unübersichtlicher Datenmengen zu erleichtern.

Literatur

- [1] Aldenderfer, Mark S.; Blashfield, Roger K. : *Cluster Analysis*. Sage Publications, Beverly Hills (Ca), USA, 1984
- [2] Bock, Hans Hermann : *Automatische Klassifikation*. Vandenhoeck & Ruprecht, Göttingen, 1974
- [3] Chambers, J.M. ; Gordesch, J. ; Klas, A. ; Lebart, L. ; Sint, P.P.: *Compstat Lectures 4*. Physica Verlag, Würzburg Wien, 1985
- [4] Cole, A.J (Hrsg.) : *Numerical Taxonomy*. Academic Press, New York, 1969
- [5] Deimer, Reinhard : *Unscharfe Clusteranalysemethoden*. Schulz-Kirchner Verlag, Idstein, 1986
- [6] Ernst, George W. ; Newell, Allen : *GPS: A Case Study in Generality and Problem Solving*. Academic Press, New York, 1969
- [7] Everitt, Brian : *Cluster Analysis*. Halsted Press, New York, 1986
- [8] Jain, Anil K. ; Dubes, Richard C. : *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs (NJ), USA, 1988
- [9] Müller, D: *Regelbasierte Systeme*. Vorlesung an der Universität Hannover, 1991
- [10] Neumann, Andreas : *Entwicklung eines lernenden Produktionensystems zur on-line Steuerung eines städtischen Kanalnetzes*. Diplomarbeit, Universität Hannover, 1985
- [11] Nillson, Nils J. : *Learning Machines*. McGraw-Hill, New York, 1965
- [12] Romesburg, H. Charles : *Cluster Analysis for Researchers*. Lifetime Learning Publications, Belmont (Ca), USA, 1984
- [13] Schilling, W. (Hrsg.) : *Real Time Control of Urban Drainage Systems* International Association on Water Pollution Research and Control, London 1989
- [14] Schneider, Stefan : *Bewertung und Verbesserung eines Expertensystems für die Steuerung von Kanalnetzen*. Diplomarbeit, Universität Hannover, 1990
- [15] Späth, Helmuth : *Cluster-Analyse-Algorithmen*. Oldenbourg Verlag, München, 1977
- [16] Spönemann, Peter : *Ein adaptives regelbasiertes System zur Abflußsteuerung*. Diplomarbeit, Universität Hannover, 1988

- [17] Therrien, Charles W. : *Decision Estimation and Classification*. John Wiley & Sons, New York, 1989
- [18] Winston Patrick, H. : *Künstliche Intelligenz*. Addison Wesley, Reading (MA), USA ,1987
- [19] Wishart, D : *Mode Analysis: A generalisation of nearest neighbour which reduces chaining effects*. In [4] , S. 282-311

Anlage 1
(Dateien, die von beiden Versionen benutzt werden)

Kanalnetzdatei für Simulation des Abwassertransportes mit EXTRAN

KRNETZ

31

TEZG E/HA L/ED H/D FRMD

NCOND NJUNC1 NJUNC2 ZP1 ZP2 GRL1 GRL2 LEN ROUG KLAS DEEP WIDE AFUL
 NCOND TEZG QIN AGES AUND ADAC RNSW ALPU ALPD VMUU VMUD VMDA RUBC STHE SPHI
 NCOND NJUNC1-X NJUNC1-Y NJUNC2-X NJUNC2-Y

HALTUNG-01SCHACHT-01SCHACHT-02 100. 98. 105. 105. 200. 1.5 1 1.

HALTUNG-01 30. 100. 50. 20. 1123

HALTUNG-01

HALTUNG-02SCHACHT-03SCHACHT-04 98.0 96.0 105. 105. 200. 1.5 1 1.5

HALTUNG-02 0. 123

HALTUNG-02

HALTUNG-03SCHACHT-05SCHACHT-06 100. 98. 105. 105. 200. 1.5 1 1.

HALTUNG-03 300. 100. 50. 20. 2123

HALTUNG-03

HALTUNG-04SCHACHT-07SCHACHT-08 98.0 96.0 105. 105. 200. 1.5 1 1.5

HALTUNG-04 0. 123

HALTUNG-04

HALTUNG-05SCHACHT-06SCHACHT-09 94. 91.0 105. 105. 300. 1.5 1 1.0

HALTUNG-05 0. 123

HALTUNG-05

HALTUNG-06SCHACHT-09SCHACHT-11 91.0 90.5 105. 105. 300. 1.5 1 1.0

HALTUNG-06 123

HALTUNG-06

HALTUNG-07SCHACHT-10SCHACHT-11 100. 98. 105. 105. 200. 1.5 1 1.0

HALTUNG-07 30. 100. 50. 20. 3123

HALTUNG-07

HALTUNG-08SCHACHT-12SCHACHT-11 95. 95. 105. 105. 100. 1.5 1 1.

HALTUNG-08 0. 123

HALTUNG-08

HALTUNG-09SCHACHT-13SCHACHT-12 95. 95. 105. 105. 100. 1.5 1 1.

HALTUNG-09 0. 123

HALTUNG-09

HALTUNG-10SCHACHT-15SCHACHT-14 100. 98. 105. 105. 200. 1.5 1 1.0

HALTUNG-10 0. 123

HALTUNG-10

HALTUNG-11SCHACHT-16SCHACHT-17104.9104.9 105. 105. 100. 1.5 1 0.09

HALTUNG-11 0. 123

HALTUNG-11

HALTUNG-12SCHACHT-18SCHACHT-19 95.0 93.0 105. 105. 200. 1.5 1 1.5

HALTUNG-12 0. 123

HALTUNG-12

JSTORE ZST1 ASTORE1 ZST2 ASTORE2 ZST3 ASTORE3 ZST4 ASTORE4

SCHACHT-02 94. 1000. 99. 1000.100.0 20. 105. 20.

SCHACHT-06 94. 1000. 99. 1000. 100. 20. 105. 20.

SCHACHT-11 90.5 1000. 95.5 1000. 96.5 20. 105. 20.

SCHACHT-14 94. 1000.99.00 1000. 100. 20. 105. 20.

SCHACHT-17 91. 1000. 96. 1000. 97. 20. 105. 20.

NRORIF NJUNC1 NJUNC2 KORI AORI CORI ZORI

NRPUMP NJUNC1 NJUNC2 KPUM VWELL VSUMP ZPUM

NRPUMP PRT1 PRT2 PRT3 PRT4 PRT5

NRPUMP VRATE12 VRATE23 VRATE34 VRATE45

NRPUMP VRATE21 VRATE32 VRATE43 VRATE54

PUMPE-01 SCHACHT-02SCHACHT-03 2

PUMPE-01	0.	1.	2.	3.	4.		
PUMPE-01		20.0		20.0		20.0	20.0
PUMPE-01		0.2		0.2		0.2	0.2
PUMPE-02	SCHACHT-02	SCHACHT-06				2	
PUMPE-02	0.03	1.	2.	3.	4.		
PUMPE-02		20.0		20.0		20.0	20.0
PUMPE-02		0.2		0.2		0.2	0.2
PUMPE-03	SCHACHT-06	SCHACHT-02				2	
PUMPE-03	0.	1.	2.	3.	4.		
PUMPE-03		20.0		20.0		20.0	20.0
PUMPE-03		0.2		0.2		0.2	0.2
PUMPE-04	SCHACHT-06	SCHACHT-07				2	
PUMPE-04	0.	1.	2.	3.	4.		
PUMPE-04		20.0		20.0		20.0	20.0
PUMPE-04		0.2		0.2		0.2	0.2
PUMPE-05	SCHACHT-14	SCHACHT-13				2	
PUMPE-05	0.	1.	2.	3.	4.		
PUMPE-05		20.0		20.0		20.0	20.0
PUMPE-05		0.2		0.2		0.2	0.2
PUMPE-06	SCHACHT-13	SCHACHT-14				2	
PUMPE-06	0.	1.	2.	3.	4.		
PUMPE-06		20.0		20.0		20.0	20.0
PUMPE-06		0.2		0.2		0.2	0.2
PUMPE-07	SCHACHT-11	SCHACHT-17				2	
PUMPE-07	0.36	1.	2.	3.	4.		
PUMPE-07		20.0		20.0		20.0	20.0
PUMPE-07		0.2		0.2		0.2	0.2
PUMPE-08	SCHACHT-17	SCHACHT-18				2	
PUMPE-08	0.36	0.5	0.6	0.7	0.8		
PUMPE-08		20.0		20.0		20.0	20.0
PUMPE-08		0.2		0.2		0.2	0.2

ITID KTID ATI1 ATI2 ATI3 ATI4 ATI5 ATI6 ATI7 WTID KOTI NITI

NRWEIR	NJUNC1	NJUNC2	KWEI	YCRE	YTOP	WLEN	COEF	IWEI	HWEI
WEHR-01	SCHACHT-02			1	5.	11.	2.	0.7	
WEHR-02	SCHACHT-17			1	5.	14.	2.	0.7	

JFREE IFRE HFRE
SCHACHT-04
SCHACHT-08
SCHACHT-19

JGATE IGAT HGAT

REGELBASIERTES SYSTEM ZUR ECHTZEITSTEUERUNG EINES KANALNETZES

VERSION GEKOPPELT MIT DEM SIMULATIONSMODELL EXTRAN

----- DATEI COMMU2='XPSSIM.DAT' N24=3 -----

LISTE DER AUSTAUSCHDATEN RBS -> SIMULATION:

ABFLUESSE = PUMPENLEISTUNG/ SCHIEBERLEISTUNG

PUMPE-01	P1
PUMPE-02	P2
PUMPE-03	P3
PUMPE-04	P4
PUMPE-05	P5
PUMPE-06	P6
PUMPE-07	P7
PUMPE-08	P8

1111111111

ANDERE MÖGLICHE DATEN

STINT

9999999999

AAAAAAAAAAAAAAAAAAAAAAAA

1.SPALTE	2.SPALTE
(XPS)	(SIM)

Anlage 2
(Dateien für die alte Version)


```

C*** DIES IST DIE DATEI FUER DEN STRAF- UND LERNPROZESS
C*** ZEILEN, DIE MIT 'C***' BEGINNEN, WERDEN ALS KOMMENTAR-
C*** ZEILEN BEHANDELT. ALS ERSTE INFORMATION WIRD DER NAME
C*** DER PARAMETERWERTDATEI EINGELESEN UND DIE VAR. IJLPMIN
C*** UND SVMIN. FORMAT: A12,1X,I5,1X,F10.2,1X,F10.2,1X,I5
C*** DIE PARAMETERWERTDATEI [PAWDAT],IJLPMIN,SVMIN,NUMMIN,LOGMIN
    PAWDAT.DAT  10   100.00   0.40   80
C*** ALS NAECHSTES WIRD DIE LISTE DER ZU BESTRAFENDEN PUMPEN
C*** EINGELESEN. DIE HIER ANGEGBENE REIHENFOLGE IST BINDEND FUER
C*** DEN GESAMTEN STRAF- UND LERNPROZESS. FORMAT 20(1X,A5)
C*** IN DER DARAUFGFOLGENDEN ZEILE (!!!) STEHEN DIE BEZEICHNUNGEN
C*** DER PUMPEN, WIE SIE IN DEN METAREGELN BENUTZT WERDEN.
    P1  P2  P3  P4  P5  P6  P7  P8
PIWAS P2WAS P3WAS P4WAS P5WAS P6WAS P7WAS P8WAS
C*** NUN WERDEN DIE WERTEBEREICHE DER LOGISCHEN VARIABLEN UND DIE
C*** OBER-/UNTERHALB BEZIEHUNGEN ZWISCHEN DEN PUMPEN UND ANDEREN
C*** LOGISCHEN VARIABLEN EINGELESEN.
C*** DAS FORMAT EINER ZEILE: A5,2(1X,A5),1X,2015
C*** ZUERST DER NAME DER LOG. VAR., DANN UNTER- UND OBERGRENZE UND
C*** SCHLIESSLICH DIE OBER- UND UNTERHALB-BEZIEHUNGEN
C***
      P1  P2  P3  P4  P5  P6  P7  P8
P1    0  4  1  1 -1  1 -1  1  1  1
P2    0  4  1  1 -1 -1  1 -1 -1 -1
P3    0  4 -1 -1  1  1 -1  1  1  1
P4    0  4  1 -1  1  1 -1  1  1  1
P5    0  4 -1  1 -1 -1  1 -1 -1 -1
P6    0  4  1 -1  1  1 -1  1  1  1
P7    0  4  1 -1  1  1 -1  1  1 -1
P8    0  4  1 -1  1  1 -1  1 -1  1
99999
C*** ZULETZT KOMMT DIE LISTE DER BLOCKVARIABLEN FUER JEDEN REGELBLOCK.
C*** FUER JEDEN BLOCK STEHEN ZWEI ZEILEN: IN DER ERSTEN ZEILE STEHEN
C*** DIE VARIABLENNAMEN, IN DER ZWEITEN ZEILE, OB DIESE VARIABLEN
C*** NUMERISCH ODER LOGISCH SIND. ' NUM' BEDEUTET NUMERISCH, ' LOG'
C*** BEDEUTET LOGISCHE VARIABLE. STEHT ALS ERSTER VARIABLENNAME
C*** 'NOPAR', SO BEDEUTET DAS, DASS FUER DIESEN BLOCK KEINE VARIABLEN-
C*** LISTE EXISTIERT. DIE ZWEITE ZEILE FUER DIESEN BLOCK HAT DANN KEINE
C*** BEDEUTUNG, MUSS ABER VORHANDEN SEIN.
C*** BEIDE ZEILEN HABEN DAS GLEICHE EINGABEFORMAT: 20(A5,1X)
C*** BLOCK 1
NOPAR

C*** BLOCK2
NOPAR

C*** BLOCK3
SCH2  SCH6 R1030
    NUM  NUM  NUM
C*** BLOCK 4
SCH2  SCH6 SCH11 R1030
    NUM  NUM  NUM  NUM
C*** BLOCK 5
SCH2  SCH6 SCH11 R1030
    NUM  NUM  NUM  NUM
C*** BLOCK 6
SCH2  SCH6 SCH11 SCH14 R1030
    NUM  NUM  NUM  NUM  NUM
C*** BLOCK 7
SCH11 SCH14 SCH17 R1030
    NUM  NUM  NUM  NUM

```

C*** BLOCK 8
SCH11 SCH14 SCH17 R1030
NUM NUM NUM NUM
C*** BLOCK 9
SCH6 SCH11 SCH17 R1030
NUM NUM NUM NUM
C*** BLOCK 10
SCH11 SCH17 R1030
NUM NUM NUM
C*** DAS ENDE
99999

C*** PRODUKTIONSDATEI 'INIPRO.DAT' (INITIALE PRODUKTIONEN)
C*** Anfangsregelmenge für das bisherige Lernverfahren

C*** ERLÄUTERUNG DER STRATEGIE:

C*** Es wird anfangs eine Strategie benutzt, die nicht aktiv (d.h. durch
C*** den Einsatz von Pumpen) in extremen Situationen Wasser entlastet.

C*** Es soll erst der Stauraum ausgenutzt werden, dann

C*** soll entlastet werden.

(SCH6 >=R 500) (SCH6 <R 600) -> (P2WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH6 >=R 600) (SCH2 <R 500) -> (P2WAS TOO HIGH) (STRAF =R -10)
(ZURUK =R 4)

(SCH6 >=R 600) (SCH2 <R 500) -> (P3WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH11 >=R 600) (SCH14 <R 600) -> (P6WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH11 >=R 600) (SCH14 <R 600) -> (P5WAS TOO HIGH) (STRAF =R -10)
(ZURUK =R 4)

(SCH14 >=R 600) (SCH11 <R 600) -> (P5WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P1WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P4WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P7WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P8WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

AAAAA

C*** BLOCK 1 INITIALISIEREN DER HILFSVARIABLEN

C*** DUMMY ist AUTOMATISCH ZU JEDEM ZEITPUNKT (=R 0) GESETZT IN 'INWORK'

C*** (= TRUE)

(DUMMY =R 0) -> (HILF1 = FALSE)

BBBBB

C*** BLOCK 2 : BESTIMMUNG DES STEUERINTERVALLS (IN SEKUNDEN)

(R1020 =R 0) -> (STINT =R 300)

(R1020 >R 0) -> (STINT =R 60)

BBBBB

C*** BLOCK 3 STEUERUNG DER PUMPE 'PUMPE 1'

(DUMMY =R 0) -> (P1 =R 0)

BBBBB

C*** BLOCK 4 STEUERUNG DER PUMPE 'PUMPE 2'

(SCH2 <=R 50) -> (P2 =R 0)

(SCH2 >R 50) (SCH2 <=R 200) -> (P2 =R 1)

(SCH2 >R 200) (SCH2 <=R 300) -> (P2 =R 2)

(SCH2 >R 300) (SCH2 <=R 400) -> (P2 =R 3)

(SCH2 >R 400) -> (P2 =R 4)

BBBBB

C*** BLOCK 5 STEUERUNG DER PUMPE 'PUMPE 3'

(DUMMY =R 0) -> (P3 =R 0)

BBBBB

C*** BLOCK 6 STEUERUNG DER PUMPE 'PUMPE 4'

(DUMMY =R 0) -> (P4 =R 0)

BBBBB

C*** BLOCK 7 STEUERUNG DER PUMPE 'PUMPE 5'

C*** Entleerung des Beckens

(SCH14 =R 0) -> P5 =R 0)

(SCH14 >R 0) (SCH11 >C SCH14) -> (P5 =R 0)

(SCH14 >R 0) (SCH11 <C SCH14) (R1030 >R 0) -> (P5 =R 0)

(SCH14 >R 0) (SCH11 <C SCH14) (R1030 =R 0) -> (P5 =R 1)

BBBBB

C*** BLOCK 8 STEUERUNG DER PUMPE 'PUMPE 6'

(DUMMY =R 0) -> (P6 =R 0)

BBBBB

C*** BLOCK 9 STEUERUNG DER PUMPE 'PUMPE 7'

(SCH11 <=R 50) -> (P7 =R 0)

(SCH11 >R 50) (SCH11 <=R 350) (SCH17 <R 500) -> (P7 =R 1)

(SCH11 >R 350) (SCH11 <=R 500) (SCH17 <R 500) -> (P7 =R 2)

(SCH11 >R 350) (SCH11 <=R 500) (SCH17 >=R 500) -> (P7 =R 0)

(SCH11 >R 500) (SCH17 <=R 500) -> (P7 =R 3)

(SCH11 >R 500) (SCH17 >R 500) -> (P7 =R 0)

BBBBB

C*** BLOCK 10 STEUERUNG DER PUMPE 'PUMPE 8'

(SCH17 <=R 250) -> (P8 =R 0)

(SCH17 >R 250) (SCH17 <=R 350) -> (P8 =R 1)

(SCH17 >R 350) (SCH17 <=R 400) -> (P8 =R 2)

(SCH17 >R 400) (SCH17 <=R 450) -> (P8 =R 3)

(SCH17 >R 450) -> (P8 =R 4)

ENDE

C*** PRODUKTIONSDATEI 'INIPRO.DAT' (INITIALE PRODUKTIONEN)
C*** Anfangsregelmenge für das bisherige Lernverfahren

C*** ERLÄUTERUNG:

C*** Die Regelmenge dient nur zum Transport des Trockenwetterzuflusses.

C*** Die Steuerung eines Regenereignisses soll vom RBS entwickelt werden.

(SCH2 >=R 500) (SCH6 <R 600) -> (P2WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH6 >=R 600) (SCH2 <R 500) -> (P2WAS TOO HIGH) (STRAF =R -10)
(ZURUK =R 4)

(SCH6 >=R 600) (SCH2 <R 500) -> (P3WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH11 >=R 600) (SCH14 <R 600) -> (P6WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH11 >=R 600) (SCH14 <R 600) -> (P5WAS TOO HIGH) (STRAF =R -10)
(ZURUK =R 4)

(SCH14 >=R 600) (SCH11 <R 600) -> (P5WAS TOO LOW) (STRAF =R -10)
(ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P1WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P4WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P7WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

(SCH2 >=R 500) (SCH11 >=R 600)
(SCH14 >=R 600) -> (P8WAS TOO LOW) (STRAF =R -20) (ZURUK =R 4)

AAAAA

C*** BLOCK 1 INITIALISIEREN DER HILFSVARIABLEN

C*** DUMMY ist AUTOMATISCH ZU JEDEM ZEITPUNKT (=R 0) GESETZT IN 'INWORK'

C*** (= TRUE)

(DUMMY =R 0) -> (HILF1 = FALSE)

BBBBB

C*** BLOCK 2 : BESTIMMUNG DES STEUERINTERVALLS (IN SEKUNDEN)

(R1020 =R 0) -> (STINT =R 300)

(R1020 >R 0) -> (STINT =R 60)

BBBBB

C*** BLOCK 3 STEUERUNG DER PUMPE 'PUMPE 1'

(DUMMY =R 0) -> P1 =R 0)

BBBBB

C*** BLOCK 4 STEUERUNG DER PUMPE 'PUMPE 2'

(DUMMY =R 0) -> P2 =R 0)

BBBBB

C*** BLOCK 5 STEUERUNG DER PUMPE 'PUMPE 3'

(DUMMY =R 0) -> (P3 =R 0)

BBBBB

C*** BLOCK 6 STEUERUNG DER PUMPE 'PUMPE 4'

(DUMMY =R 0) -> P4 =R 0)

BBBBB

C*** BLOCK 7 STEUERUNG DER PUMPE 'PUME 5'

(DUMMY =R 0) -> P5 =R 0)

BBBBB

C*** BLOCK 8 STEUERUNG DER PUMPE 'PUMPE 6'

(DUMMY =R 0) -> (P6 =R 0)

BBBBB

C*** BLOCK 9 STEUERUNG DER PUMPE 'PUMPE 7'

(DUMMY =R 0) -> (P7 =R 0)

BBBBB

C*** BLOCK 10 STEUERUNG DER PUMPE 'PUMPE 8'

(DUMMY =R 0) -> (P8 =R 0)

ENDE

Anlage 3
(Dateien für die neue Version)

REGELBASIERTES SYSTEM FUER ECHTZEITSTEUERUNG EINES KANALNETZES
VERSION GEKOPPELT MIT DEM SIMULATIONSMODELL EXTRAN

----- DATEI COMMU1='SIMXPS.DAT' N22=3 -----

LISTE DER AUSTAUSCHDATEN SIMULATION -> EXPERTENSYSTEM :

WASSERSTAENDE IN DEN SCHAECHTEN

SCHACHT-02	SCH2
SCHACHT-06	SCH6
SCHACHT-11	SCH11
SCHACHT-12	SCH12
SCHACHT-13	SCH13
SCHACHT-14	SCH14
SCHACHT-17	SCH17

1111111111

WASSERSTAENDE AM ROHR OBEN

1111111111

WASSERSTAENDE AM ROHR UNTEN

1111111111

ABFLUESSE

PUMPE-01	P1
PUMPE-02	P2
PUMPE-03	P3
PUMPE-04	P4
PUMPE-05	P5
PUMPE-06	P6
PUMPE-07	P7
PUMPE-08	P8

WEHR-01 WEHR1

WEHR-02 WEHR2

HALTUNG-02 FREI1

HALTUNG-04 FREI2

HALTUNG-12 FREI3

1111111111

Regendaten

5

10

15

1111111111

Vorhersagedaten über den Zufluss

10

20

30

9999999999

AAAAAAAAAAAAAAAAAAAAAAAA

ORGANISATION DER DATEI:

ERSTE SPALTE ZWEITE SPALTE

BEZEICHNUNG IN DER SIMULATION BEZEICHNUNG IM XPS

Gegenüber der bisherigen Version werden auch Zuflussvorhersagedaten zum RBS übertragen. Eine Zahl n bedeutet, dass eine Vorhersage für die nächsten n Minuten übertragen wird.

C*** DIES IST DIE DATEI FUER DEN STRAF- UND LERNPROZESS DES NEUEN VERFAHRENS.
 C*** ZEILEN, DIE MIT 'C***' BEGINNEN, WERDEN ALS KOMMENTARZEILEN BEHANDELT.
 C*** Als erste Information wird der Name der Parameterwertdatei eingelesen,
 C*** die die Menge der Zustände speichert, deren Steuerentscheidungen geändert
 C*** wurden.
 C*** Dazu noch eine Distanzschranke für die Zusammenhangskomponenten, die
 C*** Mindestanzahl der Punkte in einer Umgebung eines Punktes und eine
 C*** Prozentangabe für die Vereinigung kleiner Quader zu einem größeren Quader.
 C*** Der vierte Wert bedeutet folgendes: Wenn der Wert 0 ist, dann werden als
 C*** Gültigkeitsbereiche der neuen Regeln gerade die kleinsten Quader gewählt,
 C*** die die konstruierten Zusammenhangskomponenten einschliessen.
 C*** Ist der Wert 1, dann werden die Quader noch weiter zerlegt, um die Punkte
 C*** einer ZK noch genauer durch Regeln zu erfassen.
 C*** Näheres dazu in der Diplomarbeit und in den Quelltexten von NEUPRO.
 C*** FORMAT: A12,4(1X,I5)

C*** ...AAAAXIIIIIXIIIIIXIIIIIXIIII

PAWDAT.DAT 60 3 80 0

C*** ALS NÄCHSTES WIRD DIE LISTE DER ZU BESTRAFENDEN PUMPEN
 C*** EINGELESEN. DIE HIER ANGEGEBENE REIHENFOLGE IST BINDEND FUER
 C*** DEN GESAMTEN STRAF- UND LERNPROZESS. FORMAT 20(A5,1X)
 C*** In der darauffolgenden Zeile (!!!) stehen die Untergrenzen der Stufen,
 C*** in der Zeile danach die Obergrenzen der Pumpen, wie sie in den
 C*** Tabellen unten benutzt werden.

P1	P2	P3	P4	P5	P6	P7	P8
0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	4

C*** In der folgenden Tabelle bedeutet $x(i,j) > 0$, dass eine Steuerentscheidung
 C*** für ein Steuerelement j erhöht werden soll, wenn die Bewertungsvariable
 C*** den Wert 'LOW' besitzt.
 C*** Die zweite Zeile enthält die Gewichte. Je grösser der Betrag des
 C*** Gewichtes, desto mehr Einfluss hat eine Steuervariable auf die Lösung
 C*** des Problems, dass durch die zugehörige Bewertungsvariable beschrieben
 C*** wird.

C*** Format:1.Zeile: A5,20(1X,I5) 2.Zeile: 5X,20(1X,I5)

C*** 3.Zeile: 5X,20(1X,I5)

	P1	P2	P3	P4	P5	P6	P7	P8
SCH2	0	-5	5	0	1	-1	0	0
	20	20	20	20	30	30	30	30
SCH6	0	3	-3	0	1	-1	0	0
	30	30	30	30	30	30	30	30
SCH11	0	1	-1	0	3	-3	-1	-1
	30	30	30	30	30	30	30	30
SCH14	0	1	-1	0	-5	5	0	0
	30	30	30	30	30	30	30	30
SCH17	0	0	0	0	0	0	1	-1
	30	30	30	30	30	30	30	30
ENTLA	5	0	0	5	0	0	1	1
	30	30	30	30	30	30	30	30

99999

C*** ZULETZT KOMMT DIE LISTE DER BLOCKPARAMETER FUER JEDEN STEUERREGELBLOCK.
 C*** FUER JEDEN BLOCK STEHEN DREI ZEILEN: IN DER ERSTEN ZEILE STEHEN
 C*** DIE VARIABLENNAMEN, IN DER ZWEITEN ZEILE, OB DIESE VARIABLEN
 C*** NUMERISCH ODER LOGISCH SIND. ' NUM' BEDEUTET NUMERISCH, ' LOG'
 C*** BEDEUTET LOGISCHE VARIABLE. IN DER DRITTEN ZEILE STEHEN GEWICHTSFAKTOREN,
 C*** WELCHE BEIM LERNPROZESS FÜR DIE DISTANZFUNKTION NOTWENDIG SIND, DA DIE
 C*** DISTANZ ZWISCHEN ZWEI OBJEKTEN (ZUSTÄNDEN) DURCH DEN GEWICHTETEN, MITTLEREN
 C*** EUKLIDISCHEN ABSTAND BERECHNET WIRD. STEHT ALS ERSTER VARIABLENNAME
 C*** 'NOPAR', SO BEDEUTET DAS, DASS FUER DIESEN BLOCK KEINE BLOCKVARIABLEN
 C*** EXISTIEREN. DIE ZWEITE UND DRITTE ZEILE FUER DIESEN BLOCK SIND DANN

```
C*** OHNE BEDEUTUNG, MÜSSEN ABER VORHANDEN SEIN.
C*** Für den derzeitigen Lernprozeß dürfen die Variablen nur numerisch sein!
C*** DIE ERSTEN BEIDEN ZEILEN HABEN DAS GLEICHE EINGABEFORMAT: 20(A5,1X)
C*** DIE DRITTE ZEILE HAT DAS FORMAT 20(I5,1X)
C*** BLOCK 1
NOPAR

C*** BLOCK2
NOPAR

C*** BLOCK3
SCH2 SCH6 VH030
NUM NUM NUM
5 3 10
C*** BLOCK 4
SCH2 SCH6 SCH11 VH030
NUM NUM NUM NUM
5 5 4 10
C*** BLOCK 5
SCH2 SCH6 SCH11 VH030
NUM NUM NUM NUM
5 5 4 10
C*** BLOCK 6
SCH2 SCH6 SCH11 SCH14 VH030
NUM NUM NUM NUM NUM
1 5 4 2 10
C*** BLOCK 7
SCH11 SCH14 SCH17 VH030
NUM NUM NUM NUM
5 5 2 10
C*** BLOCK 8
SCH11 SCH14 SCH17 VH030
NUM NUM NUM NUM
5 5 2 10
C*** BLOCK 9
SCH6 SCH11 SCH17 VH030
NUM NUM NUM NUM
1 5 3 10
C*** BLOCK 10
SCH11 SCH17 VH030
NUM NUM NUM
1 5 10
C*** DAS ENDE
99999
```

C*** PRODUKTIONSDATEI 'INIPRO.DAT' (INITIALE PRODUKTIONEN)
C*** Anfangsregelmenge für das neuentwickelte Lernverfahren

C*** ERLÄUTERUNG DER STRATEGIE:

C*** Es wird anfangs eine Strategie benutzt, die nicht aktiv (d.h. durch
C*** den Einsatz von Pumpen) in extremen Situationen Wasser entlastet.

C*** Es soll erst der Stauraum ausgenutzt werden, dann

C*** soll entlastet werden.

C*** PRODUKTIONSDATEI 'INIPRO.DAT' (INITIALE PRODUKTIONEN)

(SCH2 >=R 500) -> (SCH2 TOO HIGH)

(SCH6 >=R 600) -> (SCH6 TOO HIGH)

(SCH11 >=R 600) -> (SCH11 TOO HIGH)

(SCH14 >=R 600) -> (SCH14 TOO HIGH)

(SCH17 >=R 500) -> (SCH17 TOO HIGH)

(SCH2 >=R 500) (SCH11 >=R 600) (SCH14 >=R 600) -> (ENTLA TOO LOW)

AAAAA

C*** BLOCK 1 INITIALISIEREN DER HILFSVARIABLEN

C*** DUMMY ist AUTOMATISCH ZU JEDEM ZEITPUNKT (=R 0) GESETZT IN 'INWORK'

C*** (= TRUE)

(DUMMY =R 0) -> (HILF1 = FALSE)

BBBBB

C*** BLOCK 2 : BESTIMMUNG DES STEUERINTERVALLS (IN SEKUNDEN)

(VH030 =R 0) -> (STINT =R 300)

(VH030 >R 0) -> (STINT =R 60)

BBBBB

C*** BLOCK 3 STEUERUNG DER PUMPE 'PUMPE 1'

(DUMMY =R 0) -> (P1 =R 0)

BBBBB

C*** BLOCK 4 STEUERUNG DER PUMPE 'PUMPE 2'

(SCH2 <=R 50) -> (P2 =R 0)

(SCH2 >R 50) (SCH2 <=R 200) -> (P2 =R 1)

(SCH2 >R 200) (SCH2 <=R 300) -> (P2 =R 2)

(SCH2 >R 300) (SCH2 <=R 400) -> (P2 =R 3)

(SCH2 >R 400) -> (P2 =R 4)

BBBBB

C*** BLOCK 5 STEUERUNG DER PUMPE 'PUMPE 3'

(DUMMY =R 0) -> (P3 =R 0)

BBBBB

C*** BLOCK 6 STEUERUNG DER PUMPE 'PUMPE 4'

(DUMMY =R 0) -> (P4 =R 0)

BBBBB

C*** BLOCK 7 STEUERUNG DER PUMPE 'PUMPE 5'

C*** Entleerung des Beckens

(SCH14 =R 0) -> P5 =R 0)

(SCH14 >R 0) (SCH11 >=C SCH14) -> (P5 =R 0)

(SCH14 >R 0) (SCH11 <C SCH14) (VHO30 >R 0) -> (P5 =R 0)

(SCH14 >R 0) (SCH11 <C SCH14) (VHO30 =R 0) -> (P5 =R 1)

BBBBB

C*** BLOCK 8 STEUERUNG DER PUMPE 'PUMPE 6'

(DUMMY =R 0) -> (P6 =R 0)

BBBBB

C*** BLOCK 9 STEUERUNG DER PUMPE 'PUMPE 7'

(SCH11 <=R 50) -> (P7 =R 0)

(SCH11 >R 50) (SCH11 <=R 350) (SCH17 <R 500) -> (P7 =R 1)

(SCH11 >R 350) (SCH11 <=R 500) (SCH17 <R 500) -> (P7 =R 2)

(SCH11 >R 350) (SCH11 <=R 500) (SCH17 >=R 500) -> (P7 =R 0)

(SCH11 >R 500) (SCH17 <=R 500) -> (P7 =R 3)

(SCH11 >R 500) (SCH17 >R 500) -> (P7 =R 0)

BBBBB

C*** BLOCK 10 STEUERUNG DER PUMPE 'PUMPE 8'

(SCH17 <=R 250) -> (P8 =R 0)

(SCH17 >R 250) (SCH17 <=R 350) -> (P8 =R 1)

(SCH17 >R 350) (SCH17 <=R 400) -> (P8 =R 2)

(SCH17 >R 400) (SCH17 <=R 450) -> (P8 =R 3)

(SCH17 >R 450) -> (P8 =R 4)

ENDE

C*** PRODUKTIONSDATEI 'INIPRO.DAT' (INITIALE PRODUKTIONEN)
C*** Anfangsregelmenge für das neuentwickelte Lernverfahren

C*** ERLÄUTERUNG:

C*** Die Regelmenge dient nur zum Transport des Trockenwetterzuflusses
C*** Die Steuerung eines Regenereignisses soll vom RBS entwickelt werden.
C*** PRODUKTIONSDATEI 'INIPRO.DAT' (INITIALE PRODUKTIONEN)

(SCH2 >=R 500) -> (SCH2 TOO HIGH)

(SCH6 >=R 600) -> (SCH6 TOO HIGH)

(SCH11 >=R 600) -> (SCH11 TOO HIGH)

(SCH14 >=R 600) -> (SCH14 TOO HIGH)

(SCH17 >=R 500) -> (SCH17 TOO HIGH)

(SCH2 >=R 500)(SCH11 >=R 600)
(SCH14 >=R 600) -> (ENTLA TOO LOW)

AAAAA

C*** BLOCK 1 INITIALISIEREN DER HILFSVARIABLEN
C*** DUMMY ist AUTOMATISCH ZU JEDEM ZEITPUNKT (=R 0) GESETZT IN 'INWORK'
C*** (= TRUE)

(DUMMY =R 0) -> (HILF1 = FALSE)

BBBBB

C*** BLOCK 2 : BESTIMMUNG DES STEUERINTERVALLS (IN SEKUNDEN)

(VH030 >R 0) -> (STINT =R 60)

(VH030 =R 0) -> (STINT =R 300)

BBBBB

C*** BLOCK 3 STEUERUNG DER PUMPE 'PUMPE 1'

(DUMMY =R 0) -> P1 =R 0)

BBBBB

C*** BLOCK 4 STEUERUNG DER PUMPE 'PUMPE 2'

(DUMMY =R 0) -> P2 =R 0)

BBBBB

C*** BLOCK 5 STEUERUNG DER PUMPE 'PUMPE 3'

(DUMMY =R 0) -> (P3 =R 0)

BBBBB

C*** BLOCK 6 STEUERUNG DER PUMPE 'PUMPE 4'

(DUMMY =R 0) -> P4 =R 0)

BBBBB

C*** BLOCK 7 STEUERUNG DER PUMPE 'PUMPE 5'

(DUMMY =R 0) -> P5 =R 0)

BBBBB
C*** BLOCK 8 STEUERUNG DER PUMPE 'PUMPE 6'

(DUMMY =R 0) -> (P6 =R 0)

BBBBB
C*** BLOCK 9 STEUERUNG DER PUMPE 'PUMPE 7'

(DUMMY =R 0) -> (P7 =R 0)

BBBBB
C*** BLOCK 10 STEUERUNG DER PUMPE 'PUMPE 8'

(DUMMY =R 0) -> (P8 =R 0)

ENDE