Zur unscharfen Klassifikation von Datenbanken mit fCQL

Andreas Meier, Christian Mezger, Nicolas Werro, Günter Schindler*
Universität Fribourg
Departement für Informatik
Rue Faucigny 2
CH-1700 Fribourg
andreas.meier@unifr.ch

Abstract

Betriebliche Informationssysteme weisen umfangreiche Datenbestände auf, die vorwiegend in relationalen Datenbanken verwaltet werden. Was oft fehlt, sind automatisierte Verfahren, um diese Bestände ohne grosse Restrukturierungen zu analysieren. Deshalb erweitern wir das relationale Datenbankschema durch ein Kontextmodell für unscharfe Klassifikation. Darauf aufbauend entwickeln wir die Sprache fCQL (fuzzy Classification Query Language), die unscharfe Abfragen mit Hilfe linguistischer Variablen an das erweiterte Datenbankschema zulässt und diese in SQL-Aufrufe an die Datenbank umwandelt. Damit geben wir dem Anwender ein Werkzeug des Data Mining in die Hand, damit er aufgrund einer vordefinierten unscharfen Klassifikation erweiterte Abfragen an seine Datenbestände starten und verbesserte Entscheidungsgrundlagen erhalten kann.

1 Motivation

Auf dem Weg zur Informationsgesellschaft ist man von einem Mangel an Daten in einen Überfluss (Information Overload) hineingeraten. Deshalb sind Unternehmen und Organisationen an Werkzeugen für die Datenanalyse interessiert, um weiterhin über Entscheidungsgrundlagen für das Geschäft zu verfügen. Von besonderem Interesse ist der Prozess des Knowledge Discovery in Databases (KDD), der wertvolle Informationen in teilweise umfangreichen Datenbanken extrahiert.

Das primäre Ziel eines KDD-Prozesses ist die Reduktion der Komplexität der Daten resp. das *Erkennen von Mustern in grossen Datenbeständen*. Klassische Verfahren wie Cluster Analyse oder Regressionsanalyse basieren meistens auf statistischen Verfahren. Sie setzen voraus, dass die Datenmengen oder Datenbanken numerische Werte resp. scharfe Datenwerte enthalten. Sobald die Daten selbst oder die Klassen, die in der Datenanalyse definiert sind und denen die Daten dann zur Komplexitätsreduktion zugeordnet werden, nicht mehr scharf definiert sind, versagen viele herkömmliche Datenanalyseverfahren.

In den Unternehmen enthalten die Datenbestände neben numerischen Daten auch nicht-numerische Informationen. Datenbankabfragesprachen setzen voraus, dass die Anfragen in der gleichen Feinheit und Genauigkeit formuliert werden, wie die Daten in den Datenbanken gespeichert sind. Relationale Abfragesprachen, wie z.B. SQL (Structered Query Language, siehe [Meier, 2003]), erlauben keine ungenau formulierten resp. unscharfen Abfragen.

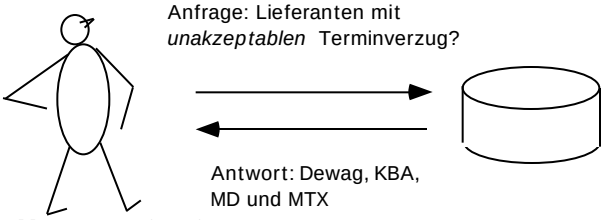

Abb. 1: Unscharfe Abfrage einer relationalen Datenbank

In Abbildung 1 ist eine unscharfe Abfrage illustriert, die mit dem vagen Begriff 'unakzeptabel' operiert (Tabellenbeispiel siehe Abbildung 3, Begriffsdefinition siehe Abbildung 6). Dieses kleine Anschauungsbeispiel illustriert das Bedürfnis, bei grossen Datenbeständen auch mit vagen und unpräzisen Anfragen operieren zu können. Auch im Falle der unscharfen Klassifikation gibt es viele praktische Beispiele aus dem Unternehmensalltag:

- *Customer Relationship Management*: In letzter Zeit hat das Management der Kundenbeziehungen und –prozesse an Bedeutung gewonnen. Dazu werden die Kunden gerne aufgrund bestimmter Merkmale resp. des Kaufverhaltens in Klassen oder Kundensegmente eingeteilt. Eine häufige Einteilung ist in A-, B- und C-Kunden, je nach der Kaufkraft des Kunden. Diese Klassenzugehörigkeit wird in den meisten Fällen scharf vorgenommen, d.h. jeder Kunde gehört zu genau einer Klasse. Sollen nun neben den abgeschlossenen Geschäften auch das Entwicklungspotenzial berücksichtigt werden, so kann ein *einzelnder Kunde nicht mehr eindeutig einem Kundensegment zugeordnet* werden. Das Analysieren des eventuell umfassenden Kundenbestandes und das Bilden unscharfer Kundensegmente drängt sich auf. Nur damit können geeignete Marketing-, Verkaufs- und After-Sales Serviceaufgaben gezielt und kostengünstig vorgenommen werden.

* Schweizerische Bundesbahnen SBB, Bern, Schweiz

- *Prüfung der Kreditwürdigkeit resp. des Risikos*: Versicherungsinstitute und Banken teilen ihre Kundschaft aus Risikoüberlegungen in diverse Klassen ein. So müssen für einen Kreditantrag das Alter, die Kaufkraft und weitere Merkmale geprüft werden. Anstelle scharfer Kreditwürdigkeits- oder Risikoklassen können unscharfe Klassen interessant sein, welche mit der Hilfe einer Zugehörigkeitsfunktion arbeiten. Aufgrund von Untersuchungen hat man festgestellt, dass bei *traditioneller Kreditwürdigkeitsprüfung und bei eindeutig unterschiedlichen Risiken durchaus dieselbe Gesamtwertung des Kunden* resultieren kann. Umgekehrt ist auch möglich, dass bei sehr ähnlichen Merkmalseigenschaften der Kunden unterschiedliche Gesamtbewertungen entstehen können.
- *Auswahl von Lieferanten*: Zur Auswahl von Lieferanten müssen Bewertungen vorgenommen werden. Herkömmliche resp. scharfe Klassifikationen der Lieferanten, z.B. nach *Liefertermin und Qualitätseigenschaften*, garantieren nicht, dass eine erfolgversprechende und längerdauernde Beziehung aufrecht erhalten werden kann. Die in den gespeicherten Lieferantendaten enthaltene, aber nicht oder noch nicht sichtbare Struktur wird durch unscharfe Klassenbildung sichtbar gemacht. Sie erlaubt, eine gezielte und effektive Behandlung der einzelnen Lieferantenklassen vorzunehmen.

Betriebliche Informationssysteme erzeugen eine Fülle von Daten. Im täglichen Geschäft aber vor allem auch zur Absicherung von Entscheidungen ist es notwendig, diese teilweise umfangreichen Datenbestände resp. Datenbanken zu analysieren. Ohne eine adäquaten KDD-Prozess mit unscharfen Datenanalysen riskiert man, auf wertvolle Informationen – Risiken wie Chancen – zu verzichten.

Im nächsten Abschnitt 2 wird der Zusammenhang zwischen der Datenbanktheorie und Fuzzy Logik aufgezeigt, der für den KDD-Prozess wichtig ist. Abschnitt 3 erläutert das Kontextmodell, das eine Klassifikation im Datenbankschema ermöglicht. Abschnitt 4 klärt den Begriff kontextredundanter Tupel und Abschnitt 5 führt linguistische Variablen im Datenbankschema ein. Abschnitt 6 zeigt wesentliche Merkmale der Sprache fCQL für unscharfe Klassifikation relationaler Datenbanken auf. Abschnitt 7 gibt einen Ausblick und illustriert weitere Fragestellungen.

2 Datenbanken und Fuzzy Logik

Bei Datenbanken, speziell auch bei relationalen Datenbanksystemen, werden die Merkmalswerte als eindeutig vorausgesetzt und Abfragen an die Datenbank ergeben klare Ergebnisse. Relationale Datenbanken lassen sich demnach wie folgt charakterisieren:

- Die Merkmalswerte in den Datenbanken sind *präzise*, d.h. sie sind eindeutig. Schon bei der Forderung der ersten Normalform verlangen wir, dass die Merkmalswerte atomar sind und aus einem wohldefinierten Wertebereich stammen. Impräzise Merkmalswerte wie der Terminverzug eines Lieferanten beträgt „2 oder 3 oder 4 Tage“ oder vage Merkmalswerte wie der Terminverzug beträgt „ungefähr 3 Tage“ sind nicht zugelassen.
- Die in einer relationalen Datenbank abgelegten Merkmalswerte sind *sicher*, d.h. die einzelnen Werte sind entweder bekannt oder sie sind unbekannt. Eine Ausnahme bilden die Nullwerte, d.h. Merkmalswerte, die nicht oder noch nicht bekannt sind. Darüber hinaus unterstützen uns die Datenbanksysteme keineswegs, etwa eine vorhandene stochastische Unsicherheit zu modellieren. Mit anderen Worten sind Wahrscheinlichkeitsverteilungen für Merkmalswerte ausgeschlossen; es bleibt schwierig auszudrücken, ob ein bestimmter Merkmalswert dem wahren Wert entspricht oder nicht.
- Abfragen an die Datenbank sind *scharf*. Sie haben immer einen dichtochomen Charakter, d.h. ein in der Abfrage vorgegebener Abfragewert muss mit den Merkmalswerten in der Datenbank entweder übereinstimmen oder nicht übereinstimmen. Eine Auswertung der Datenbank, bei der ein Abfragewert mit den gespeicherten Merkmalswerten „mehr oder weniger“ übereinstimmt, ist nicht zulässig.

Seit einigen Jahren werden Erkenntnisse aus dem Gebiet der unscharfen Logik (Fuzzy Logic) auf Datenmodellierung und Datenbanken angewendet, siehe z.B. [Bordogna and Pasi, 2000; Bosc and Kacprzyk, 1995; Chen, 1998; Petry, 1996; Pons *et al*. 2000]. Die meisten dieser Arbeiten sind theoretischer Natur; einige Forschungsgruppen haben allerdings versucht, mit Implementierungen die Nützlichkeit unscharfer Datenbankmodelle und Datenbanksysteme aufzuzeigen.

Bei der Datenmodellierung kann man sich ein grösseres Anwendungsfeld erschliessen, wenn unvollständige, vage oder impräzise Sachverhalte zugelassen werden. Mit Hilfe der Fuzzy Logic wurden verschiedene Modellerweiterungen vorgeschlagen, sowohl für das Entity Relationship Model wie für das Relationenmodell. Beispielsweise hat Chen in seiner Dissertation [Chen, 1992] die klassischen Normalformen der Datenbanktheorie zu unscharfen erweitert, indem er bei den funktionalen Abhängigkeiten Unschärfe zuliess (siehe dazu auch [Shenoi *et al*., 1992]). Viele unterschiedliche Vorschläge zu *unscharfen Datenmodellen für Datenbanken* finden sich in [Kerre and Chen; 1995].

Für die Erweiterung *relationaler Abfragesprachen mit unscharfer Logik* sind ebenfalls Untersuchungen angestellt worden. Beispielsweise schlägt [Takahashi, 1995] eine Fuzzy Query Language (FQL) auf der Basis des relationalen Domain Calculus [Ullman, 1982] vor. Die Sprache FQUERY von [Kacprzyk and Zadrozny, 1995] verwendet unscharfe Terme und ist im Microsoftprodukt Access prototypartig implementiert worden.

In unserer Arbeit über eine unscharfe Klassifikation und eine unscharfe Klassifizierungssprache fCQL (fuzzy Classification Query Language) wählen wir eine etwas andere Forschungsrichtung, die ursprünglich von [Schindler, 1998] aufgezeigt wurde: Wir beschränken uns auf

eine Erweiterung des relationalen Datenbankschemas, indem wir ein Kontextmodell für unscharfes Klassifizieren von Tabelleninhalten vorschlagen. Darauf aufbauend entwickeln wir die Sprache fCQL, die *unscharfe Abfragen mit Hilfe vordefinierter linguistischer Variablen* an das Datenbankschema zulässt und diese in SQL-Aufrufe an die darunterliegende (scharfe) Datenbank übermittelt. Damit vermeiden wir, die Datenbank mit grossem Aufwand zu einer unscharfen Datenbank zu migrieren [Meier, 1997] oder den Anwender mit einem unscharfen SQL zu konfrontieren [Finnerty and Shenoi, 1993]. Unscharfe Prädikate würden zu einer Vielzahl semantischer Effekte führen und ein Anwender müsste unterschiedliche Interpretationen vornehmen. Wir hingegen geben dem Anwender mit fCQL ein Werkzeug des Data Mining in die Hand, damit er aufgrund einer vordefinierten unscharfen Klassifikation an seine Datenbestände erweiterte Abfragen starten und verbesserte Entscheidungsgrundlagen berechnen kann.

3 Kontextmodell und Klassifikation

Umfangreiche Datenbestände sind oft unübersichtlich und deshalb schwierig zu analysieren und auszuwerten. Um aussagekräftige Informationen zu erhalten, muss der Anwender seine Bestände neu strukturieren und gegebenenfalls verdichten. Dazu sind verschiedene Methoden und Konzepte zum Aufbau und Betrieb eines Data Warehouse entwickelt worden. Auch gibt es Werkzeuge des Data Mining, um neue Erkenntnisse aus den Datenbeständen zu erschliessen.

Wir wählen den Ansatz eines Kontextmodells, um Klassen im relationalen Datenbankschema festlegen zu können. Für die Analyse und Bewertung einer grossen Anzahl von Lieferanten ist es beispielsweise sinnvoll, möglichst ähnliche Lieferanten in Klassen zusammenzufassen. Man erhält dann als Beispiel die Menge der „Lieferanten mit Qualitätsproblemen“ oder die Menge der „Lieferanten, bei denen die Geschäftsbeziehung ausgebaut werden sollte“. Eine solche Zusammenfassung von Lieferanten zu Klassen bedeutet eine Reduktion der Komplexität. Der Anwender kann damit seine Lieferantenbeziehungen übersichtlicher pflegen und analysieren, dank reduzierter Datenflut.

Darüber hinaus werden wichtige Eigenschaften der Lieferanten durch die Klassenbildung sichtbar gemacht. Dieses zusätzliche Wissen erlaubt dem Anwender, ganze Klassen gezielt und gesamthaft zu analysieren und die Beziehung unterschiedlicher Klassen herauszuarbeiten.

Bei der *Klassifikation von Objekten einer relationalen Datenbank* kann man zwischen scharfen und unscharfen Verfahren unterscheiden. Bei einer scharfen Klassifikation wird eine dichtochome Zuweisung der Datenbankobjekte zur Klasse vorgenommen, d.h. die Mengenzugehörigkeitsfunktion des Objekts zur Klasse beträgt 0 für nicht enthalten oder 1 für enthalten. Ein klassisches Verfahren würde daher einen Lieferanten der Klasse „Lieferanten mit Qualitätsproblemen“ oder der Klasse „Lieferanten, zu denen die Geschäftsbeziehung ausgebaut werden sollte“ zuordnen. Ein unscharfes Verfahren dagegen lässt für die Mengenzugehörigkeitsfunktion Werte zwischen 0 und 1 zu: Ein Lieferant kann z.B. mit einem Wert von 0.3 zur Klasse „Lieferanten mit Qualitätsproblemen“ gehören und gleichzeitig mit einer Zugehörigkeit von 0.7 zur Klasse der „Lieferanten, zu denen die Geschäftsbeziehung ausgebaut werden sollte“. Eine unscharfe Klassifikation ermöglicht daher eine differenzierte Interpretation der Klassenzugehörigkeit; man kann bei Datenbankobjekten einer Klasse zwischen Rand- oder Kernobjekten unterscheiden, zudem können Datenbankobjekte auch zu zwei unterschiedlichen Klassen gleichzeitig gehören.

Im fuzzy-relationalen Datenmodell mit Kontexten – verkürzter gesagt im Kontextmodell – ist jedem Attribut Aj, definiert auf einem Wertebereich D(Aj), ein Kontext zugeordnet. Ein Kontext K(Aj) ist eine Partition von D(Aj) in Äquivalenzklassen. Ein *relationales Datenbankschema mit Kontexten* besteht daher aus einer Menge von Attributen A=(A1,...,An) und einer Menge assoziierter Kontexte K=(K1(A1),...,Kn(An)) gemäss [Shenoi, 1995].

Für eine Bewertung von Lieferanten sollen materialspezifische Daten über die gelieferte Qualität und den Terminverzug aufgezeichnet werden. Das entsprechende Datenbankschema lfb(A,K) zur Lieferantenbewertung sei mit den Attributen

A = (Lieferant,Material,Qualität,Terminverzug)
und den Kontexten
K = (K(Lieferant),K(Material),K(Qualität),
K(Terminverzug))

spezifiziert. Dabei wird die Materialqualität mit den Begriffen D(Qualität)= {hoch,mittel,ausreichend,niedrig} beschrieben. Der Terminverzug sei die bisher aufgetretene Verspätung gegenüber dem zugesagten Liefertermin in Tagen. Für die Kontexte gelten die folgenden Partitionen:

K(Lieferant) = {{Lieferantennamen}}
K(Material) = {{Materialidentifikationsnummern}}
K(Qualität) = {{hoch,mittel},{ausreichend,niedrig}}
K(Terminverzug) = {{1,2,3,4,5},{6,7,8,9,10}}

Die Kontexte K(Lieferant) und K(Material) bestehen aus den breitesten Äquivalenzklassen, die den jeweiligen Wertebereichen entsprechen. Gemäss K(Qualität) wird für die Auswertung einer Abfrage z.B. unterstellt, dass die Qualitätsniveaus „ausreichend“ und „niedrig“ äquivalent sind. Zudem besagt K(Terminverzug), dass bei Abfragen die Unterscheidung zwischen einem Terminverzug z.B. von einem Tag und fünf Tagen unerheblich ist. In Abbildung 2 zeigen wir den Klassifikationsraum für unser einfaches Beispiel der Lieferantenbewertung. Durch die Partitionierung der Wertebereiche Qualität und Terminverzug erhalten wir die vier Äquivalenzklassen C1, C2, C3 und C4. Die inhaltliche Bedeutung der Klassen wird durch semantische Klassennamen sichtbar gemacht; so wird z.B. für die Klasse C4 der Name „Beziehung überprüfen“ gewählt. Eine Orientierung an den Äquivalenzklassen erleichtert eine sinnvolle Interpretation.

Es gehört zu den Aufgaben des Datenbankadministrators, geeignete Äquivalenzklassen in Zusammenarbeit mit den entsprechenden Fachspezialisten zu definieren.

Abb. 2: Klassifikationsraum aufgespannt durch Merkmale Qualität und Terminverzug

Bei einer Klassifikation müssen die ausgewählten Merkmale unabhängig sein. Dazu muss man gegebenenfalls eine Analyse der Daten vornehmen und die Korrelationskoeffizienten zwischen den Merkmalen berechnen. Einen weiteren Hinweis auf abhängige Merkmale erhält man durch das Aufdecken von transitiven Abhängigkeiten beim Datenbankentwurf.

4 Kontextredundante Tupel

Beim Relationenmodell spricht man von redundanten Tupeln (resp. Tabellen), wenn in den Tupelkomponenten mehrfach vorkommende Sachverhalte vorliegen, die ohne Informationsverlust herausgestrichen werden könnten. Für den Erhalt redundanzfreier Relationen ist die Theorie der Normalformen entwickelt worden (vgl. [Meier 2003] oder [Ullman, 1982]). Die Redundanz zweier Tupel im Kontextmodell ist weicher definiert: Zwei *Tupel t und t' heissen kontextredundant*, wenn alle Tupelkomponenten ti und t'i zur gleichen Äquivalenzklasse gehören. Kontextredundanzfreie Relationen erhält man durch Mischoperationen, die im Folgenden näher diskutiert werden.

lfb	Lieferant	Material	Qualität	Terminverzug
	BAW	802.025	ausr.	8
	Dewag	802.025	mittel	5
	Dewag	809.200	hoch	8
	KBA	802.025	ausr.	7
	KBA	809.200	ausr.	3
	KBA	840.024	niedrig	9
	MD	802.025	ausr.	8
	MD	809.200	mittel	4
	MTX	802.025	hoch	2
	MTX	840.024	hoch	4
	MAM	802.025	niedrig	7
	MAM	840.024	mittel	6
	ZT	809.200	hoch	8
	ZT	840.024	mittel	2

Abb. 3: Tabelle lfb zur Lieferantenbewertung

Ein Datenbankobjekt gehört zu einer Klasse, wenn sein Merkmalsvektor ins entsprechende Teilgebiet zeigt. Im vorgestellten Kontextmodell wird als Klassifikationsfunktion die mengentheoretische Vereinigung als Mischoperation gewählt. Diese Operation wird bei der Auswertung eines Ausdrucks der kontextbasierten Relationenalgebra (vgl. [Shenoi, 1995; Schindler, 1998]) durchgeführt, um kontextredundanzfreie Ergebnisrelationen zu erhalten.

Betrachten wir dazu ein Beispiel: Es soll eine Bewertung der in Abbildung 3 gezeigten Lieferanten für das Material 802.025 vorgenommen werden. Objekte sind die einzelnen Lieferanten, identifiziert durch das Merkmal Lieferant des zusammengesetzten Primärschlüssels. Die für eine Bewertung relevanten Merkmale sind Qualität und Terminverzug. Eine Klassifikation der Lieferanten für Material 802.025 liefert folgende Kombination von kontextbasierter Projektion Π und Selektion Σ (Die Operatoren der kontextbasierten Relationenalgebra werden durch griechische Grossbuchstaben ausgedrückt, in Anlehnung an die Arbeiten von [Shenoi, 1995]):

$$\Pi[\text{Lieferant,Qualität,Terminverzug,K()}](\Sigma[\text{Material}_{\sim K(\text{Material})}802.025](\text{lfb}))$$

Für die Auswertung der Abfrage wird der Kontext K(Material) durch den präzisen Kontext PK(Material)= {{802.025},{809.200},...} ersetzt. Das Ergebnis der Abfrage ist die kontextredundanzfreie impräzise Relation lfe in Abbildung 4, die eine scharfe Zuordnung der Objekte Lieferant zu den Klassen C1 und C4 in der Spalte des objektidentifizierenden Merkmals Lieferant ergibt.

lfe	Lieferant	Qualität	Terminverzug	Klasse
	{MTX, Dewag}	{hoch, mittel}	{2, 5}	C1: Beziehung ausbauen
	{KBA, MAM, BAW, MD}	{ausr. niedrig}	{7, 8}	C4: Beziehung überprüfen

Abb. 4: Scharfe Zuordnung der Lieferanten zu den Klassen C1 und C4

Dass die Auswertung einer Abfrage mit dem Kontextmodell nicht immer Ergebnisse im Sinne eines scharfen Klassifizierers erzeugt, zeigt eine materialunabhängige Bewertung der Lieferanten. Dazu betrachten wir folgende kontextbasierte Projektion der Relation lfb:

$$\Pi[\text{Lieferant,Qualität,Terminverzug,K()}](\text{lfb})$$

Die Ergebnistabelle lfs in Abbildung 5 illustriert, dass lediglich die Lieferanten BAW und MTX scharf klassifiziert wurden. Andererseits wurde z.B. der Lieferant MD der Klasse C1 und C4 zugeordnet. Dies bedeutet, dass die Mischoperation für den Lieferanten MD im Moment noch eine widersprüchliche Empfehlung liefert, nämlich „Beziehung ausbauen" (C1) und „Beziehung überprüfen" (C4). Die Auswertung einer Abfrage im Kontextmodell ordnet also nicht generell ein Datenbankobjekt einer und nur einer Klasse zu.

lfs	Lieferant	Qualität	Terminverzug	Klasse
	{***MTX***, ZT, MD, Dewag}	{hoch, mittel}	{2, 4, 5}	C1: Beziehung ausbauen
	{MAM, Dewag, ZT}	{hoch, mittel}	{6, 8}	C2: Terminverzug anmahnen
	{KBA}	{ausr.}	{3}	C3: Qualität besprechen
	{KBA, MAM, ***BAW***, MD}	{ausr., niedrig}	{7, 8, 9}	C4: Beziehung überprüfen

Abb. 5: Unscharfe Zuordnung der Lieferanten zu Klassen (ausser BAW und MTX)

Die Mehrfachzuweisung von Datenbankobjekten zu Klassen kann zu gegensätzlichen Handlungsempfehlungen führen. Um diese Unsicherheit zu reduzieren, wird das Kontextmodell mit Elementen aus der Theorie der unscharfen Mengen zu einer unscharfen Klassifikation weiterentwickelt.

5 Unscharfe Klassifikation mit linguistischen Variablen

Klassen repräsentieren verschiedene Zustände für ein bestimmtes Datenbankobjekt. Im Fall der mehrfachen Zuweisung eines Objekts zu unterschiedlichen Klassen kann man keine scharfe Abgrenzung zwischen den verschiedenen Zuständen vornehmen. Um unscharfe Klassen aus den bisher betrachteten scharfen Kontexten herleiten zu können, werden den Äquivalenzklassen zunächst verbale Begriffe zugeordnet. Mit diesen Begriffen verbindet sich dann eine Vorstellung über die zur Äquivalenzklasse gehörenden Elementen mit unscharfer Zuordnung.

Eine formale Beschreibung der Zuordnung von verbalen Begriffen zu Äquivalenzklassen ist mit dem Konzept der linguistischen Variablen möglich [Zimmermann, 1992]. Linguistische Variable haben als Werte keine Zahlen oder Verteilungen, sondern sprachliche Konstrukte resp. Terme (verbale Begriffe). Diese Terme werden inhaltlich durch unscharfe Mengen auf einer sogenannten Basisvariablen definiert. In Abbildung 6 ist als Beispiel die Definition des Merkmals Terminverzug als linguistische Variable auf der Basisvariablen „kumulierter Verzug“ gegeben. Der Definitionsbereich der Basisvariablen ist gleich dem Wertebereich D(Terminverzug). Die linguistische Variable hat die Termmenge T(Terminverzug)={akzeptabel,inakzeptabel} mit den verbalen Begriffen „akzeptabel“ und „inakzeptabel“ zur Beschreibung der beiden Äquivalenzklassen {1,2,3,4,5} resp. {6,7,8,9,10}.

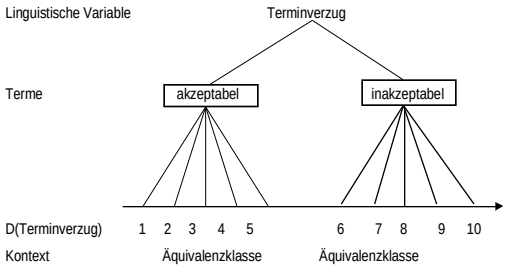

Abb. 6: Zuordnung von verbalen Begriffen zu Äquivalenzklassen

Die semantische Beschreibung unscharfer Klassen mit linguistischen Variablen erlaubt, dem menschlichen Verständnisvermögen entgegen zu kommen. Jeder Term wird durch eine unscharfe Menge repräsentiert, die durch Zugehörigkeitsfunktionen auf den Wertebereichen der Attribute definiert sind. Die vagen Terme zur Beschreibung der Äquivalenzklassen sind die Label der unscharfen Mengen. In Abbildung 7 ist ein Terminverzug gleichzeitig „akzeptabel“ und „inakzeptabel“ zum Grad von 0.5, d.h. die Zugehörigkeit von 5 Tagen zu den scharfen Mengen $\mu_{akzeptabel}$ und $\mu_{inakzeptabel}$ ist 0.5. Ein Terminverzug von 5 Tagen ist also nicht ausschliesslich akzeptabel oder inakzeptabel wie bei scharfen Klassen.

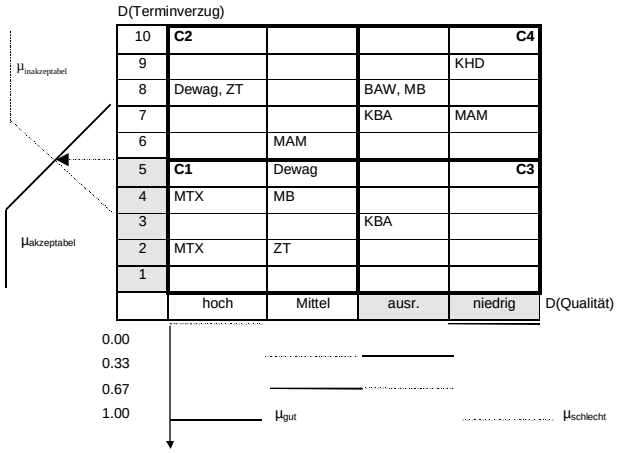

Abb. 7: Unscharfe Partitionierung der Wertebereiche mit Zugehörigkeitsfunktionen

Die Beschreibung der vagen Terme „akzeptabel“ und „inakzeptabel“ mit den Zugehörigkeitsfunktionen $\mu_{akzeptabel}$ und $\mu_{inakzeptabel}$ bewirkt, dass der Wertebereich D(Terminverzug) unscharf partitioniert wird. Analog wird auch der Wertebereich D(Qualität) durch die Terme „gut“ und „schlecht“ unterteilt. Dadurch entstehen im Kontextmodell Klassen mit kontinuierlichen Übergängen, d.h. unscharfe Klassen.

Die Klassenzugehörigkeit eines Objekts ergibt sich aus der Zugehörigkeit der Tupelkomponenten (Merkmalswerte) zu den unscharfen Mengen, deren Label die Klasse beschreiben. Die Objektzugehörigkeit von MTX z.B. mit dem Merkmalsvektor (4,hoch) bezogen auf den Term „akzeptabel“ des Merkmals Terminverzug entspricht dem Zugehörigkeitswert

$$M(MTX|akzeptabel) = \mu_{akzeptabel}(4).$$

Die Zugehörigkeit M(Oi|Ck) eines Objekts Oi zur Klasse Ck ergibt sich aus der *Aggregation über alle Terme der linguistischen Variablen*, die die Klasse definieren. Die Klasse C1 z.B. wird durch die Terme „akzeptabel“ und „gut“ beschrieben. Die Aggregation muss daher einer Konjunktion der einzelnen Zugehörigkeitswerte entsprechen. Dazu sind in der Theorie der unscharfen Mengen verschiedene Operatoren entwickelt worden, siehe z.B. [Zimmermann, 1992]. Eine generelle Operatorenklasse sind die triangularen Operatoren oder t-Normen mit dem Minimumoperator als Beispiel. Die Zugehörigkeit von MTX(4,hoch) zur Klasse C1 lautet mit diesem Operator

$$M(MTX(4,hoch)|C1) = \min\{\mu_{akzeptabel}(4),\mu_{gut}(hoch)\}.$$

Die Verwendung des Minimumoperators als Aggregationsoperator bedeutet, dass das Merkmal mit dem geringsten Zugehörigkeitswert für die Bestimmung der Zugehörigkeit eines Objekts zu einer Klasse massgebend ist. Bei einer Klassifikation von Lieferanten sei ein Lieferant bezüglich seines Terminverzuges akzeptabel, liefere jedoch eine schlechte Qualität. Eine Aggregation mit dem Minimumoperator würde den Lieferanten nur auf-

grund seiner Qualität als „schlechten“ Lieferanten einstufen. Erfolgt hingegen die Lieferantenbewertung durch menschliches Abwägen, dann würde man eine gewisse Kompensation zwischen schlechter Qualität und akzeptablem Terminverzug vornehmen. Mit anderen Worten wird die schlechte Qualität bei einer Einstufung des Lieferanten in vielen Fällen nicht das einzige Kriterium sein.

Menschliches Entscheidungsverhalten ist häufig durch kompensatorisches Abwägen gekennzeichnet. Dazu wurden in der Fuzzy Logic spezielle Operatoren wie das kompensatorische Und oder der sogenannte γ-Operator entwickelt, siehe dazu [Zimmermann, 1992; Chen, 1998]. Die unscharfe Ergebnisrelation lfu aus Abbildung 8, bei der die Tupelkomponenten unscharfe Mengen sind, zeigt ein Klassifizierungsergebnis mit einer Aggregation über den γ-Operator.

lfu	Lieferant	Qualität	Terminverzug	
	{ (MTX, 1.00), (ZT, 0.45), (MD, 0.45), (Dewag, 0.35) }	gut	akzeptabel	C1: Bez.ausb.
	{ (MAM, 0.40), (Dewag, 0.65), (ZT, 0.55) }	gut	inakzeptabel	C2: Ter.mahnen
	{ (KBA, 0.31) }	schlecht	akzeptabel	C3: Qual.bespr.
	{ (KBA, 0.69), (MAM, 0.60), (BAW, 1.00), (MD, 0.55)}	schlecht	inakzeptabel	C4: Bez.überprr.

Abb. 8: Unscharfe Klassen durch kompensatorische Aggregation mit dem γ-Operator

Die unscharfen Mengen in der Spalte Lieferant zeigen die unscharfe Zerlegung. So hat z.B. der Lieferant Dewag eine Zugehörigkeit von 0.35 zur Klasse „Beziehung ausbauen“ (C1) und 0.65 zur Klasse „Terminverzug anmahnen“ (C2). Bei der scharfen Klassifikation der Tabelle lfs aus Abbildung 5 hingegen wurde Dewag jeweils mit der Zugehörigkeit 1.0 den Klassen „Beziehung ausbauen“ und „Terminverzug anmahnen“ zugewiesen. Ein Vergleich mit dem Resultat aus Abbildung 8 illustriert, dass beim Lieferanten Dewag eher Terminprobleme und nicht Qualitätsfragen im Vordergrund stehen. Die Unsicherheit bezüglich der Pflege der Beziehung zum Lieferanten ist damit geklärt.

6 Sprache fCQL für unscharfe Klassifikation

In unserer Arbeit verzichten wir auf die Entwicklung unscharfer Abfragesprachen auf der Basis einer kontextbasierten Relationenalgebra resp. eines unscharfen Domain Calculus, wie sie verschiedentlich vorgeschlagen und teilweise implementiert wurden (vgl. dazu [Kacprzyk and Zadrozny, 1995; Shenoi, 1995; Takahashi, 1995]). Unser Sprachansatz beschränkt sich auf eine Klassifikationssprache mit dem Namen fCQL (fuzzy Classification Query Language), mit der unscharfe Klassen definiert und abgefragt werden können. Konkret verwenden wir SQL auf den darunterliegenden scharfen Datenbanken, um anschliessend eine unscharfe Ergebnisrelation zu erzeugen. Dazu ist es notwendig, mit SQL-Anweisungen kontextredundante Tupel aus den scharfen Datenbeständen zu selektieren und zu aggregieren.

Abfragesprache	Beispiel	Abfragetyp im Kontextmodell
SQL	**select** Lieferant **from** lfb **where** Terminverzug = 4	scharfe Abfrage durch präzise Kontexte mit einelementigen Äquivalenzkl.
MIQUEL	**select** Lieferant **from** lfb **where** Terminverzug ungefähr=4	unscharfe Abfrage durch scharfe Kontexte mit mehrelementigen Äquivalenzkl.
fCQL	**classify** Lieferant **from** lfb **classify** Lieferant **from** lfb **with class is** Terminproblem **classify** Lieferant **from** lfb **with** Terminverzug **is** akzeptabel **and** Qualität **is** gut	unscharfe Klassifikation durch vage Kontexte mit unscharfen Äquivalenzklassen

Abb. 9: Übersicht über kontextbasierte Sprachansätze für relationale Datenbanken

In Abbildung 9 diskutieren wir drei unterschiedliche Ansätze zu Abfragesprachen mit Kontextmodell. Die Sprache SQL kann als scharfe Abfragesprache mit präzisen Kontexten und einelementigen Äquivalenzklassen charakterisiert werden.

[Finnerty and Shenoi, 1993] haben in ihrer Arbeit die Abfragesprache MIQUEL konzipiert, um unscharfe Abfragen an eine relationale Datenbank richten zu können. Eine MIQUEL-Abfrage unterscheidet sich von einer gewöhnlichen SQL-Abfrage in der where-Klausel:

select Merkmale
from Tabellen
where Selektionsbedingung mit Kontextbezeichnung

Die einfachste Form einer *kontextbezogenen Bedingung* ist nach Finnerty und Shenoi ein Ausdruck der Form

<Attribut> <Kontext> <RelOperator> <Ausdruck>,

wie z.B. in Abbildung 9 die Bedingung „Terminverzug ungefähr = 4“. Mit anderen Worten besteht beim Sprachvorschlag MIQUEL die Abfragebedingung immer aus einem scharfen Zielwert und fakultativ aus einem Kontext. Die Sprache MIQUEL lässt also scharfe wie unscharfe Abfragen gleichzeitig zu.

Klassifikationsabfragen mit der Sprache fCQL (vgl. [Schindler, 1998; Meier *et al*., 2001]) operieren im Gegensatz zu MIQUEL auf der linguistischen Eben mit vagen Kontexten. Das hat den Vorteil, dass der Anwender keinen scharfen Zielwert und keinen Kontext kennen muss, sondern lediglich den Spaltennamen des objektidentifizierenden Merkmals und die Tabelle oder Sicht, in der die Merkmalswerte enthalten sind. Für eine gezielte Betrachtung einzelner Klassen kann der Anwender eine Klasse spezifizieren oder Merkmale mit einer verbalen Beschreibung ihrer Ausprägungen angeben. Klassifikationsabfragen arbeiten also mit verbalen Beschreibungen auf Merkmals- und Klassenebene:

classify Objekt
from Tabelle
with Klassifikationsbedingung

Die Sprache fCQL ist an SQL angelehnt, wobei die Projektionsliste bei der select-Klausel durch den Spaltennamen des zu klassifizierenden Objekts ersetzt wird.

Während die where-Klausel beim SQL eine Selektionsbedingung enthält, beschreiben wir mit der with-Klausel die Klassifikationsbedingung.

Als Beispiel einer fCQL-Abfrage erzeugt „**classify** Lieferant **from** lfb“ eine Klassifikation sämtlicher in der Tabelle lfb (aus Abbildung 3) vorhandenen Lieferanten. Mit „**classify** Lieferant **from** lfb **with class is** Terminproblem“ wird gezielt eine Klasse abgefragt. Verzichtet man auf die Definition der Klasse, so kann man mit den linguistischen Beschreibungen der Äquivalenzklassen eine bestimmte Objektmenge selektieren. Als Beispiel betrachten wir die Abfrage „**classify** Lieferant **from** lfb **with** Terminverzug **is** akzeptabel **and** Qualität **is** gut“. Diese Abfrage besteht aus dem Bezeichner des zu klassifizierenden Objekts (Lieferant), dem Namen der Grundtabelle (lfb), den kritischen Merkmalsnamen (Terminverzug resp. Qualität) sowie den Namen der vordefinierten Äquivalenzklassen (akzeptabel resp. gut). Im Anhang sind die Produktionsregeln der kontextfreien Grammatik aufgeführt.

7 Ausblick

An der Universität Fribourg haben wir einen Compiler für die Sprache fCQL entwickelt, siehe [Meier *et al.*, 2001; Savary, 2003]. Die Übersetzung einer Abfrage in fCQL resultiert nach einer lexikalischen und syntaktischen Analyse in der Generierung von SQL-Code. Für die lexikalische Analyse haben wir das Werkzeug JFlex (http://www.informatik.tu-muenchen.de/~kleing/jflex/) der TU München verwendet, das auf der bekannten Software Lex resp. JLex der Universität Princeton basiert. Zudem benutzen wir für die Syntaxprüfung die Software CUP (http://www.cs.princeton.edu/~appel/modern/java/JLex). Zur Codegenerierung mussten wir das Datenbankschema um drei beschreibende Tabellen für Kontexte, Klassen und Zugehörigkeitsfunktionen erweitern.

Im Moment testen wir unsere Implementierung an einem Datenbestand aus der Industrie. Die zuständige Marketingabteilung verfügt zwar über ein ausgereiftes Datawarehouse, doch möchte sie die Kunden- und Produktebestände mit Methoden der unscharfen Logik weiter analysieren und bewerten. Dank dem gewählten Ansatz mit unscharfen Klassen mussten wir für die Übernahme des Datenbestandes keine komplexe Datenbankmigration durchführen. Mit dem Feldtest erhoffen wir uns, dass die Klassifikationssprache fCQL den Werkzeugsatz für Data Mining und Knowledge Discovery in sinnvoller Weise ergänzt.

Literatur

[Bordogna and Pasi, 2000] Bordogna G., Paso G. (Eds.): *Recent Issues on Fuzzy Databases*. Physica-Verlag, 2000

[Bosc and Kacprzyk, 1995] Bosc P., Kacprzyk J. (Eds.): *Fuzzyness in Database Management Systems*. Physica-Verlag, 1995

[Chen, 1992] Chen G.: Design of Fuzzy Relational Databases Based on Fuzzy Functional Dependency. Ph.D. Dissertation Nr. 84, Leuven Belgium, 1992

[Chen, 1998] Chen G.: *Fuzzy Logic in Data Modeling – Semantics, Constraints, and Database Design*. Kluwer Academic Publishers, 1998

[Finnerty and Shenoi, 1993] Finnerty S., Shenoi S.: Abstraction-based Query Languages for Relational Databases. In: Wang P.P. (Ed.): *Advances in Fuzzy Theory and Technology*. Volume I, Bookwrights Durham, 1993, pp. 195-218

[Kacprzyk and Zadrozny, 1995] Kacprzyk J., Zadrozny S.: FQUERY for Access – Fuzzy Querying for a Windows-Based DBMS. In: [Bosc and Kacprzyk. 1995], pp. 415-433

[Kerre and Chen, 1995] Kerre E. E., Chen G.: An Overview of Fuzzy Data Models. In: [Bosc and Kacprzyk, 1995], pp. 23-41

[Meier, 1997] Meier A.: Datenbankmigration – Wege aus dem Datenchaos. *Praxis der Wirtschaftsinformatik*, Jhrg. 34, Nr. 194, S. 24-36

[Meier, 2003] Meier A.: *Relationale Datenbanken – Eine Einführung für die Praxis*. Springer Verlag, 2003

[Meier *et al*., 2001] Meier A., Savary C., Schindler G., Veryha Y.: Database Schema with Fuzzy Classification and Classification Query Language. *Proc. of the International Congress on Computational Intelligence – Methods and Applications*, CIMA 2001, University of Wales, Bangor U.K., June 19-22, 2001

[Petry, 1996] Petry F.E.: *Fuzzy Databases – Principles and Applications*. Kluwer Academic Publishers, 1996

[Pons *et al*., 2000] Pons O., Vila M. A., Kacprzyk J. (Eds.): *Knowledge Management in Fuzzy Databases*. Physica-Verlag, 2000

[Savary, 2003] Savary C.: Implementierung der unscharfen Klassifikationssprache fCQL als Erweiterung von SQL. Diplomarbeit, Departement für Informatik, Universität Fribourg, 2003

[Schindler, 1998] Schindler G.: *Fuzzy Datenanalyse durch kontextbasierte Datenbankanfragen*. Deutscher Universitäts-Verlag, 1998

[Shenoi *et al*., 1992] Shenoi S., Melton A., Fan L.T.: Functional Dependencies and Normal Forms in the Fuzzy Relational Database Model. *Information Sciences*, Vol. 60, 1992, pp. 1-28

[Shenoi, 1995] Shenoi S.: Fuzzy Sets, Information Clouding and Database Security. In: [Bosc and Kacprzyk, 1995], pp. 207-228

[Takahashi, 1995] Takahashi Y.: A Fuzzy Query Language for Relational Databases. In: [Bosc and Kacprzyk, 1995], pp. 365-384

[Ullman, 1982] Ullman J.D.: *Principles of Database Systems*. Computer Science Press, 1982

[Zimmermann, 1992] Zimmermann H.-J.: *Fuzzy Set Theory – and Ist Applications*. Kluwer Academic Publishers, 1992

Anhang

fCQL (fuzzy Classification Query Language)

<ClassificationQuery> → **classify** <Object> **from** <Relation> |
classify <Object> **from** <Relation>
with <ClassificationCondition>

<ClassificationCondition> → <ClassSelection> |
<AttributeSelectionList>

<AttributeSelectionList> → **(**<AttributeSelection>**)** |
<AttributeSelectionList> **or**
(<AttributeSelection>**)**

<AttributeSelection> → <AttributeConditionList>

<AttributeConditionList> → <AttributeCondition> |
<AttributeConditionList> **and**
<AttributeCondition>

<AttributeCondition> → <Attribute> **is** <EquivalenceClass> |
(<Attribute> **is** <EquivalenceClass>
or <EquivalenceClassList>**)**

<EquivalenceClassList> → <EquivalenceClass> |
<EquivalenceClassList> **or**
<EquivalenceClass>

<Attribute> → ColumnDefinition

<EquivalenceClass> → ColumnDefinition

<Relation> → RelationIdentifier | ViewIdentifier

<ClassSelection> → <ClassConditionList>

<ClassConditionList> → <ClassCondition> |
<ClassConditionList> **or**
<ClassCondition>

<ClassCondition> → **class is** <Description>

<Description> → ColumnDefinition

<Object> → ColumnDefinition