Die probabilistische Ähnlichkeitsanfragesprache QSQL2

Sascha Saretz und Sebastian Lehrack
Brandenburgischer Technische Universität Cottbus
Institut für Informatik
Postfach 10 13 44
D-03013 Cottbus, Germany
{ssaretz, slehrack}@informatik.tu-cottbus.de

ABSTRACT

Die quantenlogik-basierte probabilistische Ähnlichkeitsanfragesprache QSQL2 soll vorgestellt werden. Dabei liegt das Hauptaugenmerk auf der Formulierung von Anfragen, welche "unsicher" sind, also nicht nur die traditionelle Boolesche Werte **wahr** und **falsch** annehmen können. QSQL2 kann Ungenauigkeiten sowohl auf Relationenebene als Eintrittswahrscheinlichkeiten, als auch auf Prädikatebene als Relevanzwahrscheinlichkeiten modellieren. Zusätzlich bietet die Sprache die Eigenschaft einer Booleschen Algebra, womit bekannte Äquivalenzen für die Anfragen nutzbar sind.

1. MOTIVATION

Im traditionellen relationalen Modell von Codd [4] sind Tupel entweder in einer Relation enthalten oder nicht. Im Gegensatz dazu können die Ansätze probabilistischer Datenbanken unpräzisen Daten verarbeiten, wobei jede mit einer Eintrittswahrscheinlichkeit annotiert ist. Es ist also *unsicher*, ob ein Tupel in einem bestimmten Datenbankzustand (*mögliche Welt*) vorkommt oder nicht [5].

Ein anderer Typ von Unsicherheiten sind Ähnlichkeitsprädikate wie "Größe≈1.80m". Sie drücken unsichere Beziehungen zwischen Tupeln aus.

Dies sind zwei unterschiedliche Arten Unsicherheit zu formalisieren. Unsere Sprache QSQL2 erlaubt es beide Arten zu nutzen, was dem Nutzer mehr Freiheiten beim Stellen von Anfragen bietet. Des Weiteren beachtet die QSQL2 die Gesetze der Booleschen Algebra, womit viele für den Nutzer sehr intuitive Äquivalenzen anwendbar sind.

Um diese erweiterten Möglichkeiten zu verstehen, betrachten wir im nächsten Abschnitt zunächst eine Klassifikation von Anfragearten.

2. ANFRAGETYPEN

Wir wollen zunächst eine Klassifikation unterschiedlicher Anfrageklassen erstellen. Mit diesen sollen semantische Unterschiede zwischen Anfragen deutlich gemacht werden. Die Entwicklung dieser Klassifikation ist in [13] zu finden.

23^{rd} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 31.05.2011 - 03.06.2011, Obergurgl, Austria.

Um die Klassifikation aufzubauen identifizieren wir zwei signifikante Kriterien der Ausdrucksmächtigkeit der Anfragesprache und der darunterliegenden relationalen Datenbasis:

(i) das Einbauen von Konzepten der Ungenauigkeit und Ähnlichkeit durch Ähnlichkeitsprädikate und

(ii) Tupel, welche Konfidenzwerte besitzen.

Wir benennen die Erfüllung einer dieser beiden Kriterien mit dem Term *unsicher*. Dies bedeutet, dass wir sichere oder unsichere Anfragen auf sicheren oder unsicheren relationalen Daten anwenden. Es ist dabei darauf zu achten, dass die Begriffe *sicher* und *unsicher* auch mit anderen Bedeutungen genutzt werden. In unserem Zusammenhang nutzen wir den Begriff *unsicher* für den Datenmodellierungsaspekt. Wenn ein Nutzer nicht weiß, welche die korrekte Instanz oder der richtige Wert seiner Daten ist, kann er diese mit einem Konfidenzwert annotieren, welcher die Eintrittswahrscheinlichkeit darstellt.

Indem die beiden Klassifikationsdimensionen orthogonal angewendet werden, erhält man vier Anfrageklassen. Diese Klassen werden im Folgenden kurz beschrieben.

(i) Sichere Anfragen auf sicheren Daten (`CQonCD`)
Die Klasse `CQonCD` (Certain Queries on Certain Data) enthält alle Anfragen, welche durch Boolesche Bedingungen auf deterministischen relationalen Daten erzeugt werden. Diese Anfragen können durch traditionelle relationale Anfragesprachen wie den relationalen Kalkül, die relationale Algebra und SQL gestellt werden. Die folgenden drei Klassen enthalten `CQonCD` vollständig.

(ii) Unsichere Anfragen auf sicheren Daten (`UQonCD`)
Die Klasse `UQonCD` (Uncertain Queries on Certain Data) steht für Anfragen, welche Ungenauigkeiten und Vagheit unterstützen indem Ähnlichkeitsprädikate genutzt werden können. Diese Prädikate basieren auf einer sicheren Datengrundlage. Das Evaluationsergebnis einer solchen Anfrage kann durch einen score-Wert aus dem Intervall $[0,1]$ angegeben werden, welches den Grad der Erfüllung darstellt.

(iii) Sichere Anfragen auf unsicheren Daten (`CQonUD`)
Die Anfragen der Klasse `CQonUD` sind typisch für probabilistische Datenbanken mit Possible-Worlds-Semantik (siehe Abschnitt 3.2). Diese Anfragen nutzen Boolesche Bedingungen auf unsicheren Daten mit einem Konfidenzwert aus dem Intervall $[0,1]$.

(iv) Unsichere Anfragen auf unsicheren Daten (`UQonUD`) Wenn man die Possible-Worlds-Semantik (iii) durch Bedingungen mit Ähnlichkeitsprädikaten (ii) kombiniert, erhält man eine Anfrage einer Klasse mit erweiterter Ausdruckskraft. In `UQonUD` können Ähnlichkeitsbedingungen auf Daten genutzt werden, welche nur in einem bestimmten unsicheren Datenbankzustand gegeben sind. Die Klasse `UQonUD` umfasst die ersten drei Klassen.

Wir werden sehen, dass QSQL2 eine Vielzahl von Anfragen aus allen vier Klassen auswerten kann und somit eine große Bandbreite für die Nutzung von unsicheren Anfragen bietet.

3. DATEN- UND ANFRAGEMODELL

Nun soll das grundlegende Datenmodell der Anfragesprache QSQL2 beschrieben werden. Es kombiniert zwei Wahrscheinlichkeitsarten: (i) eine Relevanzwahrscheinlichkeit gegen eine Anfrage und (ii) eine Eintrittswahrscheinlichkeit für ein Datenobjekt.

3.1 Relevanzwahrscheinlichkeit

Um die Relevanzwahrscheinlichkeit z. B. einer `UQonCD`-Anfrage auszudrücken, nutzen wir die probabilistische Interpretation eines geometrischen Retrievalmodells, welches auf dem quadrierten Kosinus-Ähnlichkeitsmaß basiert [12]. Die Hauptidee unseres Retrievalmodells ist die Anwendung von Vektorräumen, welche auch aus der Quantenmechanik oder Quantenlogik bekannt sind, um Anfrageauswertung in Datenbanken zu betreiben. Hier wollen wir eine Idee der grundlegenden Prinzipien vermitteln. Für diesen Zweck sind Zusammenhänge zwischen Konzepten aus der Anfrageauswertung und dem angewandten Retrievalmodell in Tabelle 1 dargestellt.

Das Retrievalmodell beschreibt die Auswertung eines einzelnen Tupels gegen eine gegebene Ähnlichkeitsanfrage. Wir beginnen unsere Beschreibung, indem wir uns ein Vektorraum vorstellen, welcher die Domäne für ein Tupel ist. Alle Attributwerte eines Tupels werden durch die Richtung eines entsprechenden Tupelvektors der Länge 1 ausgedrückt. Eine logik-basierte Bedingung korrespondiert zu einem spezifischen Vektorunterraum des Domänen-Vektorraums, auch *Bedingungsraum* genannt.

Das Resultat der Auswertung ist festgelegt durch den minimalen Winkel zwischen Tupelvektor und Bedingungsraum. Der quadrierte Kosinus dieses Winkels ist ein Wert aus dem Intervall $[0,1]$ und kann daher als Ähnlichkeitsmaß interpretiert werden. Wenn also ein Tupelvektor zum Bedingungsraum gehört, kann man diese Bedingung als vollständige Übereinstimmung interpretierten (mit einem Score-Wert von 1). Im Gegensatz dazu entspricht der rechte Winkel von 90° zwischen Tupelvektor und Bedingungsraum keiner Übereinstimmung, der Score-Wert ist 0.

Anfrageauswertung		**CQQL Modell**
Wertebereich $\mathrm{Dom}(t)$	$\leftrightarrow$	Vektorraum $\mathbf{H}$
angefragtes Tupel t	$\leftrightarrow$	Tupelvektor $\vec{t}$
Bedingung c	$\leftrightarrow$	Bedingungsraum $\mathbf{cs}[c]$
Auswertung $\mathrm{eval}^t(c)$	$\leftrightarrow$	quadrierter Kosinus des Winkels zwischen $\vec{t}$ und $\mathbf{cs}[c]$ $(\cos^2(\measuredangle(\vec{t}, \mathbf{cs}[c])))$

Table 1: Zusammenhänge zwischen Anfrageauswertung und dem Retrieval-Modell CQQL

In früheren Arbeiten [12, 11] entwickelten wir eine probabilistische Interpretation für unser Retrievalmodell, daher kann das geometrische Ähnlichkeitsmaß auch als Wahrscheinlichkeit der Relevanz aufgefasst werden. Aus diesem Grund kann man die folgenden bekannten Auswertungsregeln für Wahrscheinlichkeiten anwenden, wenn alle beteiligten Teilbedingungen c_1 und c_2 unabhängig sind:

$$\begin{aligned}
\mathrm{eval}^t(c) &:= \mathrm{SF}(t,c), \text{ wenn } c \text{ atomar ist}\\
\mathrm{eval}^t(c_1 \wedge c_2) &:= \mathrm{eval}^t(c_1) * \mathrm{eval}^t(c_2)\\
\mathrm{eval}^t(c_1 \vee c_2) &:= \mathrm{eval}^t(c_1) + \mathrm{eval}^t(c_2) -\\
&\qquad \mathrm{eval}^t(c_1) * \mathrm{eval}^t(c_2)\\
\mathrm{eval}^t(\neg c) &:= 1 - \mathrm{eval}^t(c).
\end{aligned}$$

Die *Berechungsfunktion* SF den Ähnlichkeitswert für atomare Ähnlichkeitsbedingungen berechnet, z. B. 'Ort ≈ Berlin'.

Um die Unabhängigkeit der Teilbedingungen zu erhalten benötigt man die folgende Einschränkung: *In einer gültigen Bedingung darf kein Attribut gegen mehr als eine Konstante in unterschiedlichen Ähnlichkeitsprädikaten angefragt werden.* Daher ist die Bedingung 'Ort≈Berlin ∧ Ort≈München' nicht in QSQL2 erlaubt. Die Ähnlichkeitsprädikate 'Ort ≈ Berlin' und 'Ort ≈ München' können somit nicht für einen festen Ort gleichzeitig zu 1 ausgewertet werden (vollständige Übereinstimmung), was auch der Intuition entspricht. Diese Einschränkung entspricht der Unabhängigkeitsannahme von Tupel-unabhängigen bzw. Block-unabhängigen probabilistischen Datenbanken, welche im folgenden Abschnitt näher erläutert werden.

3.2 Eintrittswahrscheinlichkeit

Die Possible-Worlds-Semantik wird von den meisten probabilistischen Datenbanken genutzt um Anfragen aus der Klasse `CQonUD` zu verarbeiten.

Als Grundlage dient eine Relation $R \subseteq \mathrm{Dom}(A_1) \times \ldots \times \mathrm{Dom}(A_n)$ eines Relationenschemas $\mathrm{attr}(R) = \{A_1, \ldots, A_n\}$, wobei A_i für ein Attribut steht. Dann definiert jede Tupelteilmenge von R einen eigenen Datenbankzustand, auch *Welt* von R genannt. Nehmen wir eine ein-attributige Relation $R = \{(1),(2)\}$ an. Für dieses Beispiel sind die möglichen Zustände oder möglichen Welten durch $R_{w_1} = \{(1),(2)\}$, $R_{w_2} = \{(1)\}$, $R_{w_3} = \{(2)\}$ und $R_{w_4} = \{\}$ gegeben. Eine dieser möglichen Welten repräsentiert die eine, welche in Realtität vorkommt. Allerdings ist unbekannt, welche dies genau ist. Um diese Unsicherheit zu meistern, nutzen wir ein Wahrscheinlichkeitsmaß über der Menge aller möglichen Welten, welches aus einer probabilistischen Tabelle abgeleitet ist. Wir nennen eine Welt mit einer Eintrittswahrscheinlichkeit höher als 0 eine *mögliche Welt* oder *possible world*.

Im Allgemeinen ist die Semantik der genutzten Wahrscheinlichkeitsmaße nicht vordefiniert. Um die Wahrscheinlichkeitsberechnung zu vereinfachen nutzen wir die Semantik der probabilistischen Block-unabhängigen Datenbanken [2] für QSQL2.

In probabilistischen Block-unabhängigen Datenbanken ist jedes Tupel t mit einem Ereignis $E[t]$ verknüpft, welches das

Vorkommen oder das Nichtvorhandensein eines Tupels t in der Realität ausdrückt. Insbesondere unterscheiden wir zwei Arten von Ereignissen und Tupeln. Auf der einen Seite betrachten wir *Basisereignisse* welche von *Basistupeln* abgeleitet sind, welche durch initiale probabilistische Relationen gegeben sind. Außerdem berücksichtigen wir *komplexe Ereignisse*, welche mit während der Anfrageverarbeitung erzeugen komplexen Tupeln verknüpft sind. Diese Ereignisse bestimmen die Eintrittswahrscheinlichkeit der Ergebnistupel.

Dabei sind Tupel aus einem Block disjunkt zueinander, Tupel aus unterschiedlichen Blöcken sind unabhängig zu einander. Durch diese Vereinfachung erhält man eine relativ einfache Berechnungsvorschrift für komplexe Ereignisse.

Wenn die zugrundeliegende Ereignisstruktur unabhängig ist, kann man die Wahrscheinlichkeiten eines komplexen Ereignistupels wie in [8] berechnen:

$$\begin{aligned}\Pr(E[t_1] \wedge E[t_2]) &:= \Pr(E[t_1]) * \Pr(E[t_2]) \\ \Pr(E[t_1] \vee E[t_2]) &:= \Pr(E[t_1]) + \Pr(E[t_2]) - \\ &\qquad \Pr(E[t_1] \wedge E[t_2]) \\ \Pr(\neg E[t_1]) &:= 1 - \Pr(E[t_1]).\end{aligned}$$

3.3 Kombinierter Wahrscheinlichkeitsraum

Schlussendlich kombinieren wir die eingeführten probabilistischen Modelle, um beliebige Anfragen aus der Klasse `UQonUD` verarbeiten zu können. Dies wird getan, indem die Wahrscheinlichkeitsräume, welche Relevanz- und Eintrittswahrscheinlichkeiten repräsentieren, durch einen Produktwahrscheinlichkeitsraum vereinigt werden. Die Nutzung eines Produktwahrscheinlichkeitraumes kann durch die Klassifikation der Anfrageklasse `UQonUD` gerechtfertigt werden.

Wir nehmen also zuerst ein gegebenes Tupel als Datenbasis an, welches mit einer Eintrittswahrscheinlichkeit annotiert ist. Dann wenden wir *zusätzlich* eine Ähnlichkeitsbedingung an, um eine Relevanzwahrscheinlichkeit auf dieser Datenbasis zu erzeugen. So verhindern wir das Vermischen oder Überlappen von beiden Eingabewahrscheinlichkeiten. Somit nehmen wir an, dass beide Wahrscheinlichkeitsmaße unabhängig voneinander und in den kombinierten Produktwahrscheinlichkeitsraum eingebettet sind.

4. ANFRAGEN IN QSQL2

Um Ideen zu verdeutlichen und Beispielanfragen anzugeben wollen wir ein laufendes Beispiel einführen. Es ist ein vereinfachter Verbrechenslöser, welcher an ein Beispiel vom Trio Projekt [15] angelehnt ist. Die Datenwerte sind aus [13]. Es gibt eine deterministische Tabelle *Criminals* (abgekürzt *crim*, Tabelle 2), welche ein Dossier von registrierten Kriminellen enthält. Des Weiteren gibt es eine probabilistische Tabelle *Observations* (abgekürzt *obs*, Tabelle 3) mit Zeugenaussagen und den zugehörigen Konfidenzen.

Die Datei der Kriminellen enthalt die Attribute `name`, `status`, `sex`, `age` und `height` jeder registrierten Person, wobei die Domänen für die Attribute `status` und `sex` {free, jail, parole} und {female, male} sind.

Die Aufzeichnung der Beobachtungen beinhaltet die Zeugenaussagen für ein spezielles Verbrechen, so dass jeder Zeuge nur genau eine Person mit entsprechenden Geschlecht (`obs_sex`), geschätztem Alter (`obs_age`) und geschätzter Größe (`obs_height`) sah. Jedes Aussagentupel in `obs` ist mit einem Konfidenzwert annotiert, welcher als Eintrittswahrscheinlichkeit aufgefasst werden kann. Wir nehmen an, dass Beobachtungen von unterschiedlichen Zeugen unabhängig voneinander sind (z. B. die Tupel t_6 und t_7) und dass die Aussagen eines Zeugen disjunkt sind. Zum Beispiel kann nur maximal eins der Tupel t_4, t_5 und t_6 der Zeugin *Amber* der Wahrheit entsprechen.

Criminals (crim)					
TID	name	status	sex	age	height
t_1	Bonnie	jail	female	21	1.63
t_2	Clyde	free	male	32	1.83
t_3	Al	free	male	47	1.76

Table 2: Deterministische Informationen über registrierte Kriminelle

Observation (obs)					
TID	witness	obs_sex	obs_age	obs_height	Pr
t_4	Amber	male	30	1.85	0.3
t_5	Amber	male	35	1.90	0.3
t_6	Amber	female	25	1.70	0.3
t_7	Mike	female	20	1.60	0.7
t_8	Carl	female	30	1.80	0.9

Table 3: Zeugenaussagen annotiert mit Konfidenzwerten

Anfragen bzgl. Klassifikation: Als erstes sollen Beispiele für die Anfrageklassen aus Abschnitt 2 erfolgen. Ein typisches Beispiel für ein `CQonCD`-Anfrage ist *"Bestimme alle Kriminellen, welche den Status 'free' haben"*. Diese Anfrage ist in QSQL2 und SQL gleich, da keine Ähnlichkeitsprädikate oder probabilistischen Relationen benötigt werden (Listing 1).

Eine `UQonCD`-Anfrage ist z. B. *"Bestimme alle Kriminellen, welche den Status 'free' haben und deren Altern ungefähr 30 ist"*. Listing 2 zeigt das entsprechende Anfrage in QSQL2-Syntax. Die Vagheit ("ungefähr") wird durch ein Ähnlichkeitsprädikat ($\approx$) umgesetzt.

Als ein komplexeres Beispiel betrachten wir *"Bestimme alle Kriminellen, welche möglicherweise beobachtet wurden. Dies bedeutet, dass das Alter in einem Intervall von zehn Jahren um das beobachtete Alter liegt und dass das beobachtete Geschlecht passend ist"*. Dieses Beispiel enthält eine Boolesche Bedingung, während die Relation `Observation` probabilistisch ist (Listing 3).

Als ein Beispiel für eine Anfrage aus der Klasse `UQonUD` wollen wir eine Variante der letzten `CQonUD`-Anfrage (Listing 3) betrachten: *"Bestimme alle Kriminellen, welche möglicherweise beobachtet wurden. Dies bedeutet, dass das Alter ähnlich zum beobachteten Alter ist und dass das beobachtete Geschlecht passend ist"* (Listing 4). In dieser Anfrage kommt sowohl ein Ähnlichkeitsprädikat ($\approx$), als auch eine probabilistische Relation (`Observation`) vor.

```
SELECT name FROM Criminals C
WHERE C.status = 'free'
```

Listing 1: `CQonCD`-Anfrage

```
SELECT name FROM Criminals C
WHERE C.status = 'free' and C.age ≈ 30
```

Listing 2: UQonCD-Anfrage

```
SELECT name FROM Criminals C, Observation O
WHERE C.sex = O.sex and C.age > O.obs_age-5
  and C.age < O.obs_age+5
```

Listing 3: CQonUD-Anfrage

Logische Anfragen: Ein großer Vorteil von QSQL2 ist, dass das zugrunde liegende theoretische Fundament eine Boolesche Algebra bildet, also viele bekannte mathematische Äquivalenzen wie z. B. Distributivität, Idempotenz und Absorption erfüllt sind. An dieser Stelle sollen einige dieser logischen Eigenschaften exemplarisch von QSQL2 für praxisrelevante Anfragen genutzt werden. Wie wir später noch in Abschnitt 5.4 sehen werden, erfüllen z. B. die Fuzzy-Datenbanken nicht alle diese logischen Eigenschaften, insofern sind einige der folgenden Anfragen trotz einfacher Syntax nicht selbstverständlich.

Oft macht es Sinn Implikationen der Form $A \rightarrow B$ auszudrücken, d.h. wenn die erste Aussage wahr ist, muss es die andere auch sein. Durch die bekannte Äquivalenz $A \rightarrow B \equiv \neg A \vee B$ kann man diesen Junktor auch auf Anfragen mit Relevanz- und Eintrittswahrscheinlichkeiten anwenden. Analog verhält es sich mit der Äquivalenz $A \leftrightarrow B$. Bei ihr sind im Booleschen Fall entweder beide Variablen wahr oder beide sind falsch. Durch die Umformung $A \leftrightarrow B \equiv A \rightarrow B \wedge B \rightarrow A \equiv (\neg A \vee B) \wedge (\neg B \vee A) \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$ kann diese Aussage auch äquivalent in QSQL2 ausgedrückt werden.

QSQL2 bietet ebenfalls gewichtete Junktoren. So macht es manchmal Sinn den Einfluss einer Teilbedingung herauf- oder herabzusetzen. In der Sprache gibt es deshalb jeweils eine gewichtete Konjunktion, ausgedrückt durch $and[\theta_1, \theta_2]$, und eine gewichtete Disjunktion, ausgedrückt mit $or[\theta_1, \theta_2]$. Die Gewichtsvariablen θ_i sind reelle Zahlen aus dem Intervall $[0, 1]$, wobei ein Gewicht von 0 überhaupt keinen Einfluss und ein Gewicht von 1 normalen Einfluss bedeutet.

Man könnte sich vorstellen, dass die Identifizierung der Verdächtigen durch die Zeugen nicht eindeutig war, weil das Verbrechen bei Dunkelheit geschehen ist. So kann man folgende Variante der UQonUD-Anfrage in QSQL2 stellen: *"Bestimme alle Kriminellen, welche möglicherweise beobachtet wurden. Dies bedeutet, dass das Alter ähnlich zum beobachteten Alter ist und dass die Größe ähnlich zur beobachteten Größe ist. Die Relevanz des beobachteten Größe ist doppelt so hoch wie die des geschätzten Alters."* (Listing 5).

```
SELECT name FROM Criminals C, Observation O
WHERE C.sex = O.sex and C.age ≈ O.obs_age
```

Listing 4: UQonUD-Anfrage

```
SELECT name FROM Criminals C, Observation O
WHERE C.height ≈ O.obs_height and[ 1, 0.5 ]
      C.age ≈ O.obs_age
```

Listing 5: Beispiel für gewichtete Anfrage

5. VERGLEICHBARE ANSÄTZE

In den letzten Jahren wurden viele probabilistische relationale Datenbankansätze vorgeschlagen [3, 2, 7, 8, 6, 10, 1]. Sie unterstützen alle die Verarbeitung von probabilistischen relationalen Daten, d.h. Anfragen aus der Klasse CQonUD.

Neben der Berechnungskomplexität ist die Ausdruckskraft ein signifikantes Vergleichsmerkmal. Im Folgenden werden drei unterschiedliche Ansätze beschrieben, wie probabilistische Datenbanken um Ähnlichkeitsprädikate erweitert werden können.

5.1 Ähnlichkeitsprädikate als Built-In-Prädikate

Fuhr und Rölleke schlagen vor die Bewertungsfunktion eines Ähnlichkeitsprädikates durch eine separate probabilistische Relation umzusetzen [8]. Diese Relation für eine Ähnlichkeitsfunktion (SF-Relation) ersetzt das Ähnlichkeitsprädikat und wird durch ein Join in die Anfrage integriert.

Leider gibt es bei diesem Ansatz ein Problem bei der Konstruktion der Ähnlichkeitsfunktion SF. Die Funktion repräsentiert ein Ähnlichkeitsprädikat, aber bzgl. der Auswertung ist es kein unabhängiges Konzept, sondern unterliegt den selben Regeln wie alle probabilistische Relationen. So müssen die Tupel unabhängige Basisereignisse bilden, damit man geeignete Aggregationsfunktionen anwenden kann. Die Unabhängigkeit der Tupel in einer SF-Relation ist aber nicht gegeben. Fuhr und Rölleke schlagen daher vor nur Anfragen zu nutzen, in denen keine Tupel aus der selben SF-Relation kombiniert werden. Deshalb kann keine SF-Relation mehr als einmal in einer Anfrage vorkommen und Projektionen können nicht mehr beliebig genutzt werden.

5.2 Ähnlichkeitsprädikate als Wahrscheinlichkeit von Relationen

Der letzte Ansatz nutzte Ähnlichkeitsprädikate wie probabilistische Relationen, welche *während* der Anfrageauswertung eingebaut werden. Im Gegensatz dazu schlagen Dalvi und Suciu [6] vor, die Wahrscheinlichkeiten für die genutzten Ähnlichkeitsprädiakte *vor* der eigentlichen Anfrageauswertung auszuwerten. Die Ergebnisse dieser Vorberechnung werden als Eintrittswahrscheinlichkeiten den Relationen zugewiesen, auf welche die Ähnlichkeitsprädikate verweisen.

Dieser Ansatz arbeitet nur auf Anfragen mit konjunktivverknüpften Ähnlichkeitsprädiaten. Schon bei einer einfachen Disjunktion von Ähnlichkeitsprädiaten, welche sich auf unterschiedliche Relationen beziehen, ist es nicht mehr möglich, die Auswertung der disjunktiven Ähnlichkeitsbedingung aufzuspalten und hinunter in die entsprechenden Relationen zu schieben.

5.3 Ähnlichkeitsprädikate auf Attributebene

In anderen Modellen wie [1, 10] können Wahrscheinlichkeiten auch auf Attributebene modelliert werden. In diesem Fall ist es möglich, die Auswertung der Ähnlichkeitsprädikate in den abgefragten Attribut *vor* der eigentlichen An-

frageauswertung zu speichern. Wie beim letzten Ansatz aus Abschnitt 5.2 funktioniert dies nur bei konjunktiv verknüpften Ähnlichkeitsprädikaten, weil die Wahrscheinlichkeit eines Tupels konjunktiv aus den Wahrscheinlichkeiten der jeweiligen Attributwerte berechnet wird. Deshalb können nicht alle komplexen (z. B. disjunktiven) Kombinationen von Ähnlichkeitsprädikaten ausgewertet werden.

5.4 Fuzzy-Datenbanken

Fuzzy-Datenbanken wie FSQL [9] können ebenfalls unsichere Anfragen auf einer unsicheren Datengrundlage bewerkstelligen, allerdings sind sie kein probabilistisches Modell. Die entsprechenden Tupel-Konfidenzwerte werden einfach ohne Rücksicht auf die Semantik der Teilbedingungen aggregiert. Es findet also keine Überprüfung auf Korrelationen statt, was das Ergebnis verfälschen kann.

Außerdem bildet die Fuzzylogik [16] keine Boolesche Algebra, da bekannte Äquivalenzen wie Idempotenz und Distributibität nicht erfüllt sind. Aufgrund des Fehlens dieser elementaren Eigenschaft sind Fuzzy-Datenbanken für uns nicht geeignet. Einen ausführlichen Vergleich zwischen Fuzzy- und Quantenlogik wird in [14] gegeben.

Wir fassen zusammen, dass im Gegensatz zu QSQL2 die anderen Ansätze [8, 6, 1, 10, 9] nicht beliebige, logik-basierte Ähnlichkeitsbedingungen beherrschen.

6. ZUSAMMENFASSUNG

In dieser Arbeit wurde die quantenlogik-basierte probabilistische Ähnlichkeitsanfragesprache QSQL2 vorgestellt. Ihre Grundlagen wurden kurz dargelegt, ihre Syntax an Beispielen anschaulich gemacht und ihre Besonderheiten demonstriert.

Im Gegensatz zu probabilistischen Datenbanken ist die Integration und Nutzung von Ähnlichkeitsprädikaten in mehr Fällen möglich. Die zusätzlichen Eigenschaften einer Booleschen Algebra wie Idempotenz oder Distributibität ermöglichen bessere Resultate als z. B. bei Fuzzylogik-basierte Sprachen. Das mathematisch Fundament ermöglicht die Interpretation der Ergebnisse als Wahrscheinlichkeiten, was sie anschaulicher und verständlicher macht.

Danksagung: Diese Arbeit wurde durch die Förderung SCHM 1208/11 – 1 der Deutschen Forschungsgemeinschaft (DFG) unterstützt.

Literatur

[1] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom. Trio: A System for Data, Uncertainty, and Lineage. In *32nd International Conference on Very Large Data Bases. VLDB 2006 (demonstration description)*, September 2006.

[2] D. Barbara, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.*, 4(5):487–502, 1992.

[3] R. Cavallo and M. Pittarelli. The theory of probabilistic databases. In P. M. Stocker, W. Kent, and P. Hammersley, editors, *VLDB*, pages 71–81. Morgan Kaufmann, 1987.

[4] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.

[5] N. N. Dalvi, C. Ré, and D. Suciu. Probabilistic Databases: Diamonds in the Dirt. *Commun. ACM*, 52(7):86–94, 2009.

[6] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[7] D. Dey and S. Sarkar. A probabilistic relational model and algebra. *ACM Trans. Database Syst.*, 21(3):339–369, 1996.

[8] N. Fuhr and T. Rölleke. A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. *ACM Trans. Inf. Syst.*, 15(1):32–66, 1997.

[9] J. Galindo, A. Urrutia, and M. Piattini. *Fuzzy Databases: Modeling, Design and Implementation*. Idea Group Publishing, Hershey, USA, 2006.

[10] C. Koch. MayBMS: A system for managing large uncertain and probabilistic databases. *Managing and Mining Uncertain Data*, 2008.

[11] S. Lehrack, S. Saretz, and I. Schmitt. QSQL^p: Eine Erweiterung der probabilistischen Many-World-Semantik um Relevanzwahrscheinlichkeiten. In T. Härder, W. Lehner, B. Mitschang, H. Schöning, and H. Schwarz, editors, *BTW*, volume 180 of *LNI*, pages 494–513. GI, 2011.

[12] S. Lehrack and I. Schmitt. A Probabilistic Interpretation for a Geometric Similarity Measure. In *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ECSQARU '11, June 2011.

[13] S. Lehrack and I. Schmitt. A unifying probability measure for logic-based similarity conditions on uncertain relational data. In *Proceedings of the 1st Workshop on New Trends in Similarity Search*, NTSS '11, pages 14–19, New York, NY, USA, 2011. ACM.

[14] I. Schmitt, A. Nürnberger, and S. Lehrack. On the Relation between Fuzzy and Quantum Logic. In *Views on Fuzzy Sets and Systems from Different Perspectives, chapter 5*. Springer-Verlag, 2009.

[15] J. Widom. Trio: A system for data, uncertainty, and lineage. In *Managing and Mining Uncertain Data*, pages 113–148. Springer, 2008.

[16] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.