Datenintegration durch Objekt-Identifikation.

Die Zusammenführung von Datenquellen, die keine gemeinsamen Identifizierer enthalten.

Mattis Neiling[1]

Abstract

Data integration is an important topic in the information age. Although structural aspects are widely investigated, there is a lack of research on semantic discrepancies between data sources.

Data integration should be able to handle input errors such as erroneous data and misspellings. Also domain and data type mismatch, the problems of missing values and duplicated records need investigation. Object identification is essential for the task of integration, especially if keys are absent or incorrect. This approach utilizes the properties, which can be derived from the data sources for identification — the derivable attributes. For two sources, the values of the derivable attributes are compared and classified for pairs of records. A random sample of pairs is used for detecting similarities, rules or classification criteria. Different Statistical and Data Mining Techniques could be applied to classify the pairs of the sources in order to link them or not.

Keywords: database integration, record linkage, entity identification, derivable attributes, common properties, semantic discrepancies and similarities

[1]Freie Universität Berlin, Institut für Wirtschaftsinformatik, Garystr. 21, D–14195 Berlin, Internet: `http://www.wiwiss.fu-berlin.de/~mneiling`, Email: `Mattis.Neiling@wiwiss.fu-berlin.de`

Jetzt wächst zusammen,
was zusammengehört!
Willy Brandt

1 Einleitung

Fehlerfreie Daten gibt es nicht! Es ist eine Binsenweisheit, daß Datenquellen fehlerbehaftet sein können. Sowohl bei der Datenerfassung als auch bei der maschinellen Weiterverarbeitung, überall können sich Fehler einschleichen.

Wie sieht es in der Welt der Datenbanken und Informationssysteme aus? Um die Identifizierbarkeit von Datensätzen innerhalb einer Datenbank zu garantieren, werden aufwendige Mechanismen eingesetzt. In einem offenen Informationssystem wie dem Internet ist es ein aussichsloses Unterfangen, ein Analogon zu dem Kontrollmechanismus eines Datenbankmanagement-Systems zu installieren, ohne die Autonomie der Datenquellen zu stören. Deshalb wird es immer Fälle geben, wo man nicht genau sagen kann, ob sich zwei Datensätze auf ein und dieselbe Sache beziehen oder ob nicht! Es sei bemerkt, daß sich dieses Problem auch dann stellen kann, wenn eindeutige Identifizierer für die verschiedenen Datenbanken festgelegt wurden: Ohne Kontrollmechanismus wird es des Öfteren zu Fehlern kommen.

In einem föderierten Datenbanksystem stellt sich die Frage ebenso: Kann man verlangen, Datensätze eindeutig identifizieren zu können? Wohl eher nicht, sonst müßte man nicht zwischen *verteilten* und *föderierten Datenbanken* unterscheiden!

Nun, oft wird man sich damit behelfen müssen, Heuristiken für diese Entscheidung zu verwenden oder aber: *Record Linkage. Unter Record Linkage verstehen wir, daß die Datenintegration mit einer Bewertung verbunden wird.*

Die Objekt-Identität wird aus den Daten ermittelt, wobei konsequenterweise Fehler entstehen können!

Der hier verfolgte Ansatz geht davon aus, daß für die Feststellung der Objekt-Identität ein Lernprozeß notwendig ist. Es wird sich zumeist um überwachtes Lernen handeln (*supervised learnig*), da sich nur in Ausnahmefällen das Weltwissen über den Objekt-Bezug direkt angeben lassen kann.

Ziel der vorliegenden Arbeit ist es, ein gemeinsames Grundgerüst zu errichten, das für viele Problemstellungen sowohl der Datenintegration als auch der Objekt-Identifikation verwendbar ist.

Den Kern bildet die Modularisierung der einzelnen Schritte, die für die Integration nötig sind. Dadurch können — je nach Aufgabenstellung und Bedarf — einzelne Komponenten des Datenintegrations-Prozesses ausgetauscht werden. Schließlich soll der Vergleich verschiedener Lösungsansätze im Rahmen der hier dargestellten Herangehensweise ermöglicht werden.

Die Aufgabe der Objekt-Identifikation gibt es auch in der Bildverarbeitung, dort werden beispielsweise bestimmte Merkmale für beobachtete Objekte aus einem Bild oder einer Videosequenz abgeleitet, die dann für die Ermittlung der Identität verwendet werden (Huang and Russell 1998).

Es sei erwähnt, daß im Datenbank-Bereich der Begriff der SAME-Relation für das Datensatz-Identifikationsproblem gebräuchlich ist, siehe Kashyap and Sheth (1996) oder Müller (1997). Im Juli 1999 hat das DFG-Projekt DIAsDEM begonnen, dort sollen verschiedene Datenbanken und semistrukturierte Dokumente integriert werden, wobei Schemaerkennung und Objektidentifikation mittels Data Mining Techniken gelöst werden sollen.[2]

1.1 Was zeichnet Datenintegration aus?

Ziel der Datenintegration ist es, verschiedene Datenquellen zusammenzufassen. Dabei sollen die Datensätze, die zusammengehören (d.h. dasselbe Realwelt-Objekt betreffen) einander zugeordnet werden.

Es sich kann sich hierbei um die Anreicherung bzw. Ergänzung von Informationen in einer Datenquelle handeln. Möglich sind aber auch statistische Auswertungen durch das Zusammenfügen verschiedener Datenquellen wie z.B. administrativer Register.

Die Quellen (wie Datenbanken oder Online-Kataloge) enthalten Informationen über Realwelt-Objekte wie Personen, Bücher oder Krankheiten. Bei der Integration sollen diejenigen Informationen, die sich auf ein Objekt (z.B. eine Person) beziehen, in einem Datensatz zusammengefaßt werden. Diese Zusammenfassung soll weitestgehend automatisch erfolgen.

Im Allgemeinen muß davon ausgegangen werden, daß *kein* eindeutiger und einheitlicher Identifizierer (Schlüssel) in den Datenquellen vorhanden ist, der sich zur Ermittlung der Objekt-Identität eignen könnte.[3] Aufgrund dessen

[2] DIAsDEM: Datenintegration von Altlasten und semistrukturierten Dokumenten durch Einsatz von Mining Verfahren (gefördert von der Deutschen Forschungsgemeinschaft DFG); Ansprechpartner/Projektleiter: Dr. Myra Spiliopoulou, Humboldt-Universität zu Berlin, `http://www.wiwi.hu-berlin.de/~myra` und Dr. Stefan Conrad, Universität Magdeburg, `http://wwwiti.cs.uni-magdeburg.de/~conrad`

[3] Sofern ein gemeinsamer Schlüssel in beiden Datenquellen enthalten ist, der es uns ermöglicht, die Objekt-Identität festzustellen, ist die Zuordnung der Datensätze kein Problem: Mit diesem Schlüssel läßt sich die Datenintegration leicht bewerkstelligen. Lediglich Fehler in diesem Schlüssel können zu Problemen führen; dann ist abzuwägen, wie diese gelöst werden sollen, oft ist eine manuelle Bereinigung denkbar, der hier dargestellte Datenintegrations-Ansatz ist aber auch für diese Fälle möglich.

kann die Objekt-Identität nur aus den Attributausprägungen ermittelt werden. Das heißt, die Zuordnung der einzelner Datensätze ist nur mittels der gespeicherten Merkmale möglich, die in *beiden* Quellen enthalten sind. Diese *identifizierenden Merkmale* können aber auch aus den Datenquellen auf geeignete Weise abgeleitet werden, wie sich z.B. das Alter aus dem Geburtsdatum ergibt oder der Ort durch die Postleitzahl bestimmt ist. Denkbar ist aber auch die Einbeziehung weiterer Informationen (z.B. Metadaten oder externe Quellen), um einen Datensatz einem Objekt zuordnen zu können.

1.2 Information Integration — Datenintegration im Internet

Angenommen, es sei ein Programm verfügbar, das eine Suchanfrage gegen eine oder mehrere Internet-Quellen (Online-Datenbanken) starten kann und die Ergebnisse auf einem lokalen Rechner verfügbar macht. Dieses Programm übernimmt dann die automatische Extraktion von Daten aus diesen Internet-Quellen durch eine Folge systematischer Abfragen. Daher ist es möglich, aus den gewonnenen Daten der Datenbank ein "neues" Informationssystem aufzubauen.

Dieser Ansatz wird in (Bornhövd and Buchmann 1999) verfolgt, wo verschiedene Internet-Quellen (z.B. Autovermietungen) zusammengeführt werden, wobei die exakte Übereinstimmung *identifizierender Attribute* in den Datenquellen die Objekt-Identität und damit die Datensatz-Zuordnung liefert. Ein weiteres Beispiel ist das Projekt *JEDI* der GMD.[4]

In dieser Arbeit wird ein für die Datenintegration verwendbares Modell der Objekt-Identifikation dargestellt. Die Bausteine des Datenintegrations-Prozesses werden vorgestellt. Schließlich wird der Einsatz geeigneter Methoden für die Klassifikation dargestellt. Eine Fallstudie (ein Bibliothekskatalog) verdeutlicht den Prozeß der Datenintegration. Eine Diskussion und offene Forschungsaspekte bilden den Abschluß.

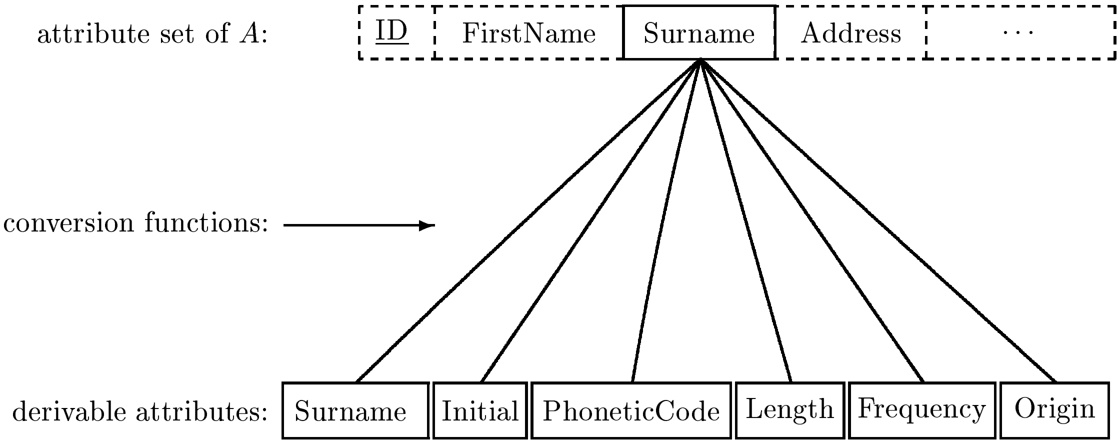
2 Ein Modell zur Objekt-Identifikation

2.1 Die Notation

U bezeichne das Universum der betrachteten wohlunterscheidbaren Objekte, über die in den Datenquellen A und B Daten gespeichert sind.

Definition 2.1 (Domain). X sei eine Menge von Attributen (*attributes, properties*) über U.

[4] *JEDI: Java Extraction and Dissemination of Information* der Gesellschaft für Mathematik und Datenverarbeitung (GMD) in Darmstadt, siehe (Huck, Fankhauser, Aberer, and Neuhold 1998)



Surname $\vdash$ Surname: $h :=$ `Identity` Surname $\vdash$ Initial: $h :=$ `Extract First Letter`
Surname $\vdash$ Length: $h :=$ `Count Letters` Surname $\vdash$ PhoneticCode: $h :=$ `Compute Phonetic Code`
(Surname$|$ $\mathcal{C}$) $\vdash$ Frequency: $h :=$ `Frequency In Source` A; $\mathcal{C}$ is the known frequency of surnames in A
(Surname$|$ $\mathcal{C}$) $\vdash$ Origin: $h :=$ `Decide Whether Domestic Surname`; $\mathcal{C}$ is knowledge of domestic surnames

Abbildung 1: Beispiele für die aus einem Nachnamen ableitbaren Attribute

Mit $\mathcal{D}_X$ bezeichnen wir die Menge aller Werte, die X annehmen kann ($\mathcal{D}_X$ ist die *Domain* von X).

Eine Attributmenge ist also die Zusammenfassung mehrerer Attribute (Für die Tabelle einer Datenbank ist dies eine Auswahl der Spalten).

Wie lassen sich die Attribute zweier Attributmengen miteinander vergleichen? Indem man sie auf gemeinsame Attribute zurückführt, die sog. ableitbaren Attributen. Dieses sind geeignete Attribute (Eigenschaften) auf dem Universum U.

Definition 2.2 (DERIVABILITY). Ein Attribut e mit Domain $\mathcal{D}_{\{e\}}$ ist aus einer Attributmenge X ableitbar genau dann (*derivable attribute*), wenn eine Abbildung $h_X : \mathcal{D}_X \to \mathcal{D}_{\{e\}}$ existiert, die die Domain $\mathcal{D}_X$ in die Domain $\mathcal{D}_{\{e\}}$ transformiert (*conversion mapping*).

Schreibweise: $X \vdash e$ oder $X \vdash \{e\}$.

Formal:

$$X \vdash e \quad :\Longleftrightarrow \big(\exists\, h_X : \quad \mathcal{D}_X \to \mathcal{D}_{\{e\}}\big). \tag{1}$$

Die Funktion h_X läßt sich auch als die Projektion des Wertebereiches $\mathcal{D}_X$ einer Attributmenge X in den Wertebereich $\mathcal{D}_{\{e\}}$ eines Attributes $\{e\}$ auffassen.

Es sei bemerkt, daß sich ein ableitbares Attribut möglicherweise erst durch das Zusammenspiel mehrerer Attribute unter Berücksichtigung von Kontext-Information ergibt. Sie muß also nicht explizit in den Datenquellen enthalten sein, es genügt, daß sich der Wert dieser Eigenschaft für die Datensätze auf eine geeignete Weise herleiten läßt. Formal läßt sich die Kontext-Information durch eine Parametrisierung der Konversionsabbildung h_X fassen:

Definition 2.3. Die Ableitbarkeit einer Attributes e mit Domain $\mathcal{D}_{\{e\}}$ aus einer Attributmenge X unter Verwendung von Kontext-Information $\mathcal{C}$ (Schreibweise $(X \mid \mathcal{C}) \vdash E$) definieren wir wie folgt:

$$(X \mid \mathcal{C}) \vdash e \quad :\Longleftrightarrow \big(\exists\, \mathcal{C}\; \exists\, h_{(X|\mathcal{C})} : \quad \mathcal{D}_X \to \mathcal{D}_{\{e\}}\big). \tag{2}$$

Die Konversionsabbildung schreiben wir dann auch als $a^{'} := h_X(a \mid \mathcal{C})$, sie wird durch den Kontext $\mathcal{C}$ parametrisiert. Sprechweise: Der Wert $a^{'}$ ergibt sich aus a unter h_X *gegeben* $\mathcal{C}$. Als Kurzschreibweise verwenden wir auch für die Ableitbarkeit bei gegebener Kontext-Information $X \vdash e$.

Die Definition der Ableitbarkeit wird auf eine Menge ableitbarer Attribute ausgedehnt:

Definition 2.4. Die Ableitbarkeit einer Attributmenge $E = \{e_1, \dots, e_k\}$ mit Domain $\mathcal{D}_E = \mathcal{D}_{\{e_1\}} \times \dots \times \mathcal{D}_{\{e_k\}}$ aus einer Attributmenge X (Schreibweise $X \vdash E$) definieren wir wie folgt:

$$X \vdash E \quad :\Longleftrightarrow \big(X \vdash e_i \quad \forall\, i = 0, \dots, k\big). \tag{3}$$

Unser Ziel ist die Verknüpfung der Datenquellen A und B. Für die Zuordnung einzelner Datensätze müssen wir die Attributmengen der beiden Datenquellen auf vergleichbare, korrespondierende Attribute untersuchen. Als korrespondierend fassen wir zwei Attributmengen auf, wenn sich aus Ihnen *ein und dasselbe* Attribut ableiten läßt (resp. eine ganze Attributmenge).

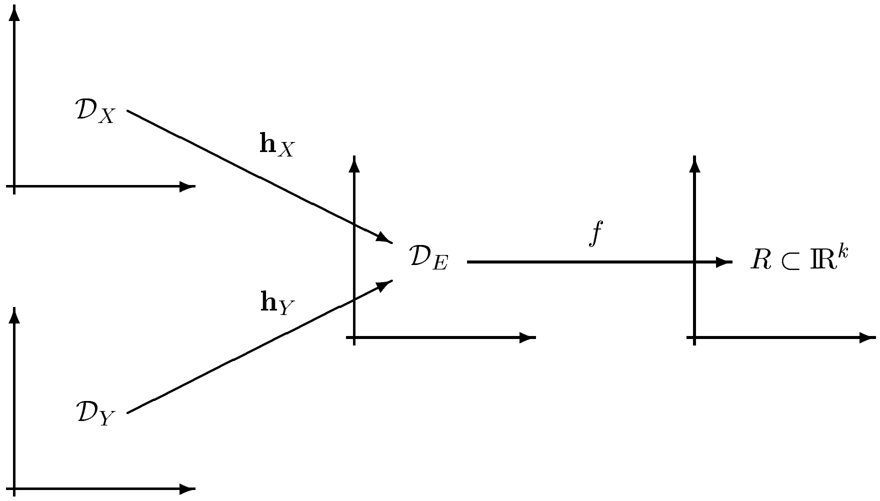
Definition 2.5 (Correspondence). X, Y und E seien Attributmengen über dem Universum von Objekten U.

Dann definieren wir die Korrespondenz der Attributmengen X und Y bezüglich E (Schreibweise $X \sim Y$ w.r.t. E; *X and Y are correspondent w.r.t. E*)) wie folgt:

$$X \sim Y \text{ w.r.t. } E \quad :\Longleftrightarrow \quad X \vdash E \,\wedge\, Y \vdash E. \tag{4}$$

Die Begriffe der Ableitbarkeit und Korrespondenz verdeutlicht das

Beispiel 2.6. U sei die Menge aller Personen. In der Datenquelle A sei das Attribut $A.GebDat$ für das Geburtsdatum enthalten, in B das aktuelle Alter $B.Alter$. Die Transformation der beiden Domains in die Domain der



$\mathcal{D}_X$,$\mathcal{D}_Y$:	the domains of two attribute sets X and Y
$\mathcal{D}_E$:	the domain of the corresponding attributes
$R \subset \mathrm{I\!R}^k$:	the comparison space
$\mathbf{h}_X$,$\mathbf{h}_Y$:	the conversion mappings
f:	the comparison function at $\mathcal{D}_E \times \mathcal{D}_E$

$\dim \mathcal{D}_X = m, \ \dim \mathcal{D}_Y = n, \ \dim \mathcal{D}_E = d \geq \dim R = k$

Abbildung 2: Die Abbildungen für die Objekt-Identifikation. Aus den Werten der Attribute zweier Datensätze werden die Werte der korrespondierenden Attribute bestimmt und diese schließlich miteinander verglichen.

Eigenschaft $e = Alter$ einer Person ergibt sich hier wie folgt:

$$h_X : \mathcal{D}_X \to \mathcal{D}_{\{Alter\}}, \quad h_X(date \mid today) := Year(today - date),$$
$$h_Y : \mathcal{D}_Y \to \mathcal{D}_{\{Alter\}}, \quad h_Y(age) := Identity = age,$$

wobei $X := \{A.GebDat\}$, $Y := \{B.Alter\}$, $\mathcal{D}_X := DATE$, $\mathcal{D}_Y, \mathcal{D}_{\{Alter\}} := \{0, 1, 2, 3, \ldots\}$ und $\mathcal{C}$ den Kontext (das zum Zeitpunkt der Bestimmung aktuelle Datum *today*) bezeichnen. Also haben wir $(X \mid \mathcal{C}) \vdash \{Alter\} \wedge Y \vdash \{Alter\}$, d.h. $X \sim Y$ w.r.t. *Alter*.

Auf den ableitbaren Attributen $(e_1, \ldots, e_k)$ können wir die einzelnen Datensätze der Datenquellen miteinander vergleichen: Hierzu leiten wir für jeden Datensatz $a \in A$ und $b \in B$ die Werte der ableitbaren Attributen $(e_1, \ldots, e_k)$ ab, und erhalten so die Werte $a' = (a'_1, a'_2, \ldots, a'_k)^{\mathrm{T}}$ und $b' = (b'_1, b'_2, \ldots, b'_k)^{\mathrm{T}}$, die wir dann direkt miteinander vergleichen können.

Definition 2.7 (Comparison Function). $E = \{e_1, e_2, \ldots, e_k\}$ sei eine Menge von $k \geq 1$ ableitbaren Attributen zweier Attributmengen X und Y, d.h. $X \sim Y$ w.r.t. E.

The Work Flow for Data Integration

Step 1: Preprocessing

- Select derivable attributes from data sources
- Construct conversion mappings
- Define an appropriate comparison function
- Choose the method and criterion for classification (e.g. Decision Tree & Entropy)

Step 2: Sampling

- Select random sampled pairs of tuples
- Denote whether the tuples refers to the same object

Step 3: Learning

- Learning the parameters of the classification function from the sample

Step 4: Validation

- Estimate the accuracy (e.g. error rates) of the classification from testing sample
- Go to Step 1, 2 or 3, if results are poor

Step 5: Integration

- Compute the comparison values for each pair of elements from the sources
- Classify the pairs
- Resolve contradictory cases
- Merge the sources

Abbildung 3: Der Prozeß der Datenintegration

Dann können wir auf dem Kreuzprodukt der Domain $\mathcal{D}_E = \mathcal{D}_{\{e_1\}} \times \dots \times \mathcal{D}_{\{e_k\}}$ von E mit sich selbst eine Vergleichsfunktion f (*comparison function*) definieren, deren Werte in einem Vergleichsraum $R \subset \mathrm{I\!R}^k$ (*comparison space* liegen:

$$f : \mathcal{D}_E \times \mathcal{D}_E \to R \subset \mathrm{I\!R}^k, \quad f(a', b') = \big(f_1(a'_1, b'_1), \dots, f_k(a'_k, b'_k)\big)^{\mathrm{T}},$$

wobei a' und b' die Werte der ableitbaren Attributen zweier Datensätze $a \in A$, $b \in B$ sind, die miteinander verglichen werden sollen. Die Werte a' und b' werden komponentenweise mittels der Abbildungen $h_{X,1}, \dots, h_{X,k}$ und $h_{Y,1}, \dots, h_{Y,k}$ gebildet. Der Wert a' eines transformierten Datensatzes berechnet sich nach $a' = (a'_1, a'_2, \dots, a'_k)^{\mathrm{T}} = \big(h_{X,1}(a), h_{X,2}(a), \dots, h_{X,k}(a)\big)^{\mathrm{T}}$ für ein $a \in A$, für $b \in B$ analog. Die Funktion f läßt sich damit auch schreiben als $f(a', b') = \big(\mathbf{h}_X(a), \mathbf{h}_Y(b)\big)$, wobei $\mathbf{h}_X := (h_{X,1}, \dots, h_{X,k})^{\mathrm{T}}$ bezeichne.

Bemerkung 2.8. Die einzelnen Komponenten $f_i, i = 1, \dots, k$ der Vergleichsfunktion f können entweder kategorielle (nominale oder ordinale) Werte $(0, 1, 2, \dots, c_k,\ c_k \geq 1)$ haben, gemäß der Definition 2.7 sind aber auch kontinuierliche Werte zulässig, wie sie beispielsweise durch ein Abstandsmaß (eine Metrik) entstehen können.

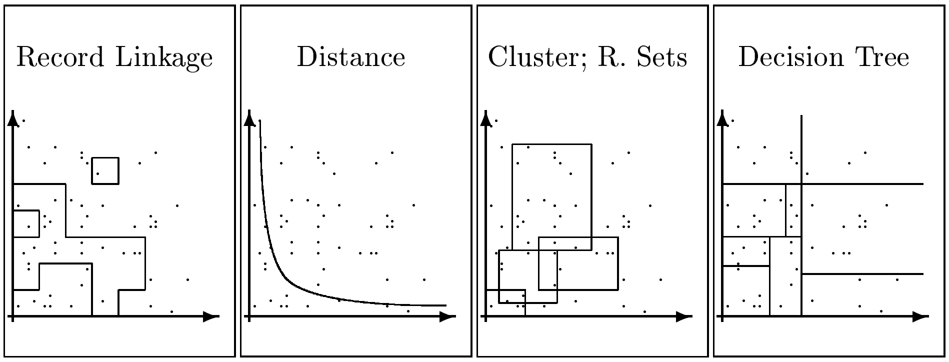


Abbildung 4: Verschiedene Klassifikationsmethoden. Die Zerlegung des Vergleichsraumes $R \subset \mathbb{R}^k$. Die Punkte bezeichnen Vergleichswerte für Datensatzpaare

3 Mögliche Ansätze für die Klassifikation

Man könnte sagen: Das Problem der *Datenintegration* ist gelöst, da es auf eine bekannte Problemklasse zurückgeführt werden konnte: Es bleibt ein Klassifikationsproblem zu lösen. Dieser Problemklasse wollen wir aber doch etwas Aufmerksamkeit schenken, da für den Erfolg eine sinnvolle Wahl des Verfahrens entscheidend ist. Der Phantasie sind keine Grenzen gesetzt, welches Klassifikationsverfahren eingesetzt werden könnte, vier Möglichkeiten sind in der Abbildung 4 gezeigt.

Für den Fall kategorieller Werte der Vergleichsfunktion sind Regel-erzeugende Verfahren wie Entscheidungsbäume und Rough Sets denkbar (siehe Abschnitt 3.3 und 3.4). Möglich ist aber auch der Einsatz wissensbasierter Systeme.

Sofern eine Verteilungsfunktion (für kategorielle Werte die der Multinomialverteilung) geschätzt werden kann, so kommen statistische Verfahren ins Spiel, wie der monotone Likelihood-Quotienten-Test im klassischen Record-Linkage-Ansatz (Abschnitt 3.2).

Für kontinuierliche Werte — wie bei Verwendung einer Metrik als Vergleichsfunktion — bietet sich der *k-Nearest-Neighbour-Ansatz* an, bei dem eine neue Beobachtung derjenigen Klasse zugeordnet wird, die die meisten der k ähnlichsten Beobachtungen enthält, d.h. die nächsten im Sinne der Metrik, siehe Schürmann (1996, S.233ff).

In den nächsten Abschnitten wird kurz angerissen, wie sich drei Verfahren für die Klassifikation einsetzen lassen. Zuvor wird noch angegeben, was wir unter einer Entscheidungsregel verstehen.

3.1 Entscheidungsregeln

Aus einer Beispielstichprobe (von Datensatzpaaren) lassen sich Regeln ableiten. Dazu berechnet man die Vergleichswerte und notiert für jedes Datensatzpaar, ob sich die Einträge auf dasselbe Objekt beziehen oder nicht. Die Vergleichswerte finden sich unten in der <Bedingung> wieder, der Objektbezug in der <Folgerung>.

Wir betrachten für eine allgemeine Regel eine Beispielmenge mit positiven und negativen Beispielen. Eine Regel läßt sich wie folgt formulieren:

if <Bedingung> **then** <Folgerung>
    with confidence conf%
    and support supp%.

wobei die Anteile conf und supp folgendermaßen berechnet werden:

Wir haben N_0 Fälle beobachtet, bei denen <Folgerung>= **TRUE** gilt sowie N_1 Fälle mit <Folgerung>= **FALSE**. Unter den N_0 positiven Beispielen sind q_0 Fälle zu finden, für die <Bedingung>= **TRUE** gilt. Bei den N_1 negativen Beispielen finden sich q_1 Fälle, für die ebenfalls <Bedingung>= **TRUE** gilt. dann können wir definieren:

$$\texttt{conf} := \frac{q_0}{q_0 + q_1}, \quad \texttt{supp} := \frac{q_0}{N_0} \tag{5}$$

- *Confidence* läßt sich am besten mit "Glaubwürdigkeit" übersetzen. conf $= 100\%$ bedeutet, es gibt kein Gegenbeispiel für diese Regel in der Stichprobe.
- Der *Support* gibt an, wieviele Fälle es gibt, die mit dieser Regel erfaßt werden. supp $= 50\%$ heißt, daß die Hälfte aller Fälle durch diese Regel bestimmt wird.

Eine gemäß der Definition 2.7 gebildete Vergleichsfunktion f hat in vielen Fällen endlich viele diskrete Werte (kategorieller oder ordinaler Natur), so daß sich für eine Stichprobe eine Entscheidungstabelle erzeugen läßt. Aus dieser Tabelle lassen sich viele (mehr oder weniger gute) Regeln ableiten.

Die Herleitung von Regeln ist für großen Stichproben sehr rechenintensiv, wenn man den Support nicht zu stark beschneidet (Etwa: nur Regeln mit Support größer 5%). Dieses kann aber dazu führen, daß gerade die interessanten Fälle, die unterhalb der 5%-Grenze liegen, unter den Tisch fallen.

Das führt auf zwei untrennbare zentrale Fragen:

1. Welche Regeln sollen generiert werden, wie soll eine Regel verfeinert werden (d.h. wie läßt sie sich verbessern)?
2. Wie läßt sich messen, ob eine Regel besonders gut geeignet ist, die Zusammenhänge in den Daten zu beschreiben?

3.2 Der klassische Record-Linkage-Ansatz

Der von Howard Newcombe, Ivan Fellegi und Alan Sunter in den 50ern und 60ern entwickelte Ansatz des *Record Linkage* wird für die Zusammenführung großer Datenbestände seit über 20 Jahren angewandt.[5] Hauptsächlich wird diese Art der Datenintegration (Der Begriff *Record Linkage* kommt aus diesem Umfeld) für statistische Zwecke benützt, aber wie schon der Titel der ersten Arbeit von H. Newcombe aussagt, werden auch medizinische Datensätze zusammengeführt und zwar einerseits personenbezogene Daten als auch krankheitsbezogene (z.B. für statistische Untersuchungen über Krankheitsursachen). Die Vielfalt der Anwendungsgebiete und den aktuellen Stand der Dinge findet man in den Proceedings der einschlägigen Konferenzen in Arlington, Virginia (Kilss and Alvey (1985), Alvey and Jamerson (1997)).

Der klassische Record-Linkage Ansatz ist Wahrscheinlichkeits-basierend, auf endlich vielen Vergleichsfällen werden die Parameter der Multinomial-Verteilung geschätzt.

Im Einzelnen sieht das Vorgehen so aus:

Zuerst wird eine Vergleichsfunktion (γ genannt) direkt auf den Datenquellen A und B definiert, die endlich viele Vergleichswerte hat.

Auf zwei Beispiel-Stichproben, je eine für zusammengehörige Paare (*M, Matches*) und unpaarige Fälle (*U, Unmatches*) wird die Verteilungsfunktion der Multinomial-Verteilung geschätzt. Für alle auftretende Werte γ erhalten wir zwei bedingte Wahrscheinlichkeiten:[6]

$$P(\gamma \mid M) \text{ und } P(\gamma \mid U). \tag{6}$$

Aus diesen Werten wird ein Likelihood-Quotient gebildet:

$$\lambda(\gamma) := \frac{P(\gamma \mid M)}{P(\gamma \mid U)}. \tag{7}$$

Offensichtlich gibt dieser Quotient an, wie gut ein Datensatzpaar $(a, b) \in A \times B$ mit Vergleichswert γ zueinander paßt:

L^+ $(\lambda(\gamma) > 1)$: Es ist wahrscheinlicher,
daß sie zusammengehören als der umgekehrte Fall

$L^\pm$ $(\lambda(\gamma) \approx 1)$: Die Wahrscheinlichkeit ist für beide Fälle genauso hoch

[5] die grundlegende Arbeit (Standardreferenz) ist Fellegi and Sunter (1969), die erste Veröffentlichung zum Thema ist Newcombe, Kennedy, Axford, and James (1959). In Neiling (1998) wird dieser Ansatz auch dargestellt.

[6] Es handelt sich um die bedingte Wahrscheinlichkeit für einen Vergleichswert γ, wissend das ein Paar aus M bzw. U ist.

L^- $(\lambda(\gamma) < 1)$: Es ist wahrscheinlicher, daß sie nicht zusammengehören

Aus diesen drei Fällen werden drei mögliche Entscheidungen abgeleitet: Entweder als *Link* (L^+) oder als *Nonlink* (L^-) registriert oder, für den unendscheidbaren mittleren Fall als *possible Link* ($L^{\pm}$) klassifiziert.[7]

Diese Entscheidungsfunktion ist unverzerrt, sofern

(i) die Verteilungsfunktion stimmt und

(ii) die Stichproben repräsentativ sind

Das Problem (ii) stellt sich generell, ist hier aber von besonderer Relevanz.

Zu (i) läßt sich feststellen, daß nur bei bedingter Unabhängigkeit der Verteilungsfunktions-Komponenten auf den Daten untereinander (z.B. darf aus dem Fehlen des Alters in einem Datensatz nicht die ethnische Gruppenzugehörigkeit folgen), da die Verteilungsfunktion i.d.R. über die Randverteilungen der einzelnen Komponenten geschätzt wird. Aus diesem Grund stellt sich die Definition der Vergleichsfunktion als besonders wichtig für die Treffsicherheit der Entscheidung heraus.

Dieses Problem wurde erkannt und infolgedessen wird in den letzten Jahren verstärkt versucht, dem Rechnung zu tragen, indem bei der Schätzung der Verteilungsfunktion Interaktionen zwischen den Komponenten bis zur 3. Stufe mit berücksichtigt werden.[8] Hierbei wird ein spezieller Expectation-Maximisation-Algorithmus (EM) angewandt, um die Paramenter der Verteilungsfunktion zu bestimmen, siehe dazu die Arbeiten Winkler (1993), Winkler (1995), Meng and Rubin (1993), Liu and Rubin (1994).

3.3 Entscheidungsbäume

Entscheidungsbaum-Verfahren (wie Quinlan's Verfahren[9] C4.5 oder CART: *Classification and Regression Tree*) eignen sich sehr gut, um aus einer Beispielsammlung (wie etwa eine Informationstabelle) Regeln abzuleiten.

Die Idee bei dem Verfahren C4.5 ist hierbei, die beobachteten Fälle dergestalt zu partitionieren, daß die "Unordnung" minimal wird. Die Unordnung einer Menge von Beobachtungen wird beispielsweise mit der *Entropie* gemessen, die für eine Zufallsvariable $X(\omega)$ mit Werten x wie folgt definiert

[7] Diese Vorgehensweise entspricht einem zweiseitigen monotonen Likelihood-Quotienten-Test, für statistische Testtheorie siehe z.B. Mood, Graybill, and Boes (1974).

[8] Interaktionen im Sinne loglinearer Modelle, siehe dazu beispielsweise Whittaker (1990) oder Bishop, Fienberg, and Holland (1975)

[9] siehe Quinlan (1993); oft wird C4.5 unter dem Namen des Vorgängers ID3 genannt

ist:

$$E = -\sum_{\text{all } x} \big(p_x \cdot \log_2 p_x\big), \tag{8}$$

mit der Konvention $0 \log_2 0 = 0$ und der abkürzenden Schreibweise $p_x := P\big(X(\omega) = x\big)$.

Die Entropie ist ein Maß für die erwartete Ungenauigkeit für Realisierungen der Zufallsvariable.

Im einfachsten Fall ist die betrachtete Zufallsvariable einfach die Ziehung einer Beobachtung aus einer Menge von Beobachtungen, und wird als gleichverteilt angesehen (d.h. die Wahrscheinlichkeit der Ziehung jeder einzelnen Beobachtung ist genauso groß). Ist eine Wahrscheinlichkeitsfunktion für die einzelnen Beobachtungsfälle (Vergleichsfälle) bekannt, so kann man diese bei der Berechnung der Entropie in der Formel (8) explizit berücksichtigen.

Das Verfahren funktioniert nach folgendem Prinzip:

1. Finde die Eigenschaft und eine Zerlegung ihrer Werte, bei der die Verminderung der Entropie durch eine Aufteilung der beobachteten Fälle am größten ist (Partitionierung der Domain).

2. Teile die Menge der Beobachtungen dort und erhalte so mehrere Knoten (z.B. zwei), die jeweils einen Teil der Beobachtungen enthalten

3. Fahre für jeden Knoten für die darin enthaltenen Beobachtungen mit Schritt 1 fort, sofern sich die Entropie dieses Knotens noch vermindern läßt.

Für die Fallstudie im Kapitel 4 wird das Verfahren exemplarisch angewandt.

3.4 Rough Sets — ungehobelte Mengen?

Die *Rough Set Theory* ist ein noch recht junges Forschungsgebiet der Mathematik und Künstlichen Intelligenz. Sie wurde von Zdzisław Pawlak in den 80er Jahren eingeführt.

Mittlerweile gibt es — neben der polnischen Gruppe um den Schöpfer Z. Pawlak selbst — einige Forscher in Japan, Kanada und den USA, die sich sowohl grundlagentheoretisch als auch praxisorientiert betätigen. Als kompakte und gut verständliche Einführung sei Munakata (1998, S. 140–182) empfohlen, als Standardreferenz sei auf die Arbeiten Pawlak (1982) und Pawlak (1991) verwiesen.

3.4.1 Was zeichnet die Rough Set Theory aus?

Es sei hervorgehoben, daß sie sehr gut geeignet ist, um Beziehungen (*relationships*) in Daten aufzudecken, die durch Relationen beschreibbar sind. Relationen sind dabei im Sinne der Mengenlehre und Logik zu verstehen, d.h. wir können jede Beziehung darstellen, die in der Potenzmenge (Der Menge aller Teilmengen) der beobachteten Datensätze enthalten ist.

Wichtige Charakteristika der Rough Set Theory sind:

- Sie kommt ohne zusätzlichen Annahmen über eine Wahrscheinlichkeitsverteilung aus.
- Die Relationen werden direkt aus den Daten ermittelt.
- Mit den Mengenoperationen Vereinigung, Durchschnitt und Komplement läßt sich jeder Zusammenhang stringent formulieren (Relationen-Algebra).
- Jede Relation auf einer Menge induziert eine Partitionierung (Bildung von Äquivalenzklassen)
- Das Konzept der oberen und unteren Approximation einer Menge eignet sich vorzüglich zur Beschreibung von Abhängigkeiten zwischen Eigenschaften.
- Mit der Berechnung von *Reducts* läßt sich die Auswahl von Eigenschaften für die Regelerzeugung steuern (Feature Selection)

Nun könnte man jetzt fragen, warum überhaupt jemand andere Verfahren einsetzt, wenn Rough Sets so schöne Eigenschaften haben. Es ist offensichtlich, daß — sofern eine Wahrscheinlichkeitsverteilung angegeben werden kann und sei dies auch nur näherungsweise — anderen Verfahren (Entscheidungsbaumverfahren, statistischen Verfahren) der Vorzug gegeben werden muß.

Für Entscheidungsprobleme auf kontinuierlichen Mengen (wie wir sie bei Verwendung eines Abstandsmaßes erhalten) sind Rough Sets — aufgrund der dann nötigen recht grobkörnigen Diskretisierung — ebenfalls nicht besonders gut geeignet; dieses ist übrigens eine Eigenart, die sie mit dem probabilistischen Record Linkage gemein haben.

Nichtsdestotrotz, für die Partitionierung eines nicht zu großen Zustandsraumes mit endlich vielen Werten läßt sich dieser Ansatz gut verwenden.

Mit dem Handwerkszeug der Rough Set Theory läßt sich aus einer Tabelle mit Beispielen ermitteln, welche Eigenschaften relevant sind, um eine Zielvariable zu erklären.

Für den praktischen Einsatz läßt sich die in Poznan entwickelte Software *ROSE: The Rough Sets Data Explorer* verwenden.[10]

4 Die Fallstudie: Ein Bibliothekskatalog

Wir betrachten einen Bibliothekskatalog, der über das Internet verfügbar sei, den Online-Katalog *GBVdirekt* des Gemeinsamen Bibliotheksverbundes GBV (`http://www.gbv.de`). Nach einer Suchanfrage wird dem Benutzer die Trefferliste angezeigt, deren detaillierte Einträge über einen Hyperlink erreichbar sind. In dieser Online-Datenbank sind die Bestandsdaten vieler Bibliotheken Deutschlands zusammengefaßt. Durch unterschiedliche Erfassung der Bücher in den verschiedenen Bibliotheken kommt es zu Mehrfachnennunngen von Büchern, die in mehreren Bibliotheken vorhanden sind. Teilweise sind die Verweise eines Buches auf verschiedene Bibliotheken schon zusammengefaßt: Ein Beispiel für vollzogene Datenintegration.[11]

Dieses Beispiel soll das Prinzip des Record Linkage verdeutlichen. Die Duplikate lassen sich weitestgehend identifizieren, so daß sich die Qualität der Online-Datenbank verbessern ließe.

Wir wollen die Online-Datenbank auf Mehrfachnennungen untersuchen. Dazu betrachten wir das Universum U der Bücher, wo wir eine Ausgabe (Auflage) eines Buches als ein Objekt auffassen. Die Bestandsdaten eines Buches in der Datenbank A sollen in *einem* Datensatz einer Zieldatenbank C zusammengefaßt werden.

Für dieses Beispiel wurde die Suchanfrage Titelstichwörter=*"database systems introduction"*) gegen die GBV-Datenbank gestartet, die 30 Treffer ergab. Dabei treten einige Mehrfachnennungen auf. Insbesondere finden sich Duplikate für die verschiedenen Ausgaben des Buches *C.J. Date: An Introduction to Database Systems, Reading Mass: Addison-Wesley* in der Trefferliste, insgesamt 22 Einträge in der Tabelle 2. Ein Datensatz wird dabei als HTML-Dokument mit der in der Tabelle 5 wiedergegebenen Struktur angezeigt.

Es sei bemerkt, daß dieselbe Suchanfrage (*"database systems introduction"*) gegen die Online-Datenbank des Deutschen Bibliotheksinstitutes in Berlin (`http://www.dbilink.de`) 187 Treffer ergab, wovon sich 171 (*sic!*) auf die verschiedenen Ausgaben des Werkes von C.J. Date bezogen. Hier zeigt sich der Bedarf nach einer effektiven Integration der Verweise auf ein Buch in einen Datensatz sehr deutlich.

[10] Siehe Predki et al. (1998); unter `http://www-idss.cs.put.poznan.pl/rose/index.html` ist eine Lite-Version kostenlos zu erhalten.

[11] Das Beispiel ist erstmals in Neiling (1998) erwähnt und wird weiter ausgebaut.

Titel:	An introduction to database systems / C. J. Date
Verfasser:	C. J. Date
Ausgabe:	6. ed.
Erschienen:	Reading, Mass : Addison-Wesley, 1995
Umfang:	XXIII, 839 S. : graph. Darst. ; 24 cm
Serie:	Addison-Wesley systems programming series
Anmerkung:	Früh. Aufl. erschienen zweibändig
	Literaturangaben
ISBN:	0-201-54329-X
Schlagwoerter:	*Datenbanksystem
Sachgruppen:	54.64 Datenbanken

Abbildung 5: Die Detailansicht eines Datensatzes bei *GBVdirekt.* Auf der linken Seite stehen die Attributnamen der Datenquelle — die Spaltennamen der Online-Datenbank A (A.Titel, A.Verfasser ...)

Step 1: Preprocessing

Für den Vergleich der Datensätze ermitteln wir gemeinsame Eigenschaften der Objekte, über die Daten gespeichert sind. Da wir Einträge innerhalb der Datenbank miteinander vergleichen, sind das sicherlich alle Attribute dieser Datenbank sowie die aus diesen ableitbaren Attribute.

In diesem Beispiel werden wir die Datensätze auf $k = 5$ abgeleiteten Eigenschaften $e_1 =$ ISBN, $e_2 =$ Name (Nachname des Autors), $e_3 =$ Title, $e_4 =$ Year (Erscheinungsjahr) und $e_5 =$ Pages (Seitenzahl) untersuchen.[12] Es sei bemerkt, daß die Eigenschaft *ISBN* konstruktionsgemäß ein eindeutiger Identifizierer ist, aber durchaus fehlerhaft enthalten sein kann oder möglicherweise gänzlich fehlt.

Die Konvertierungsfunktionen ergeben sich, wie man sich leicht an der Gestalt eines detaillierten Datensatzes klar macht (siehe Abbildung 5), wie

[12] Für den Vergleich empfiehlt sich die Auswahl geeigneter Eigenschaften, die Verwendung aller ableitbaren Eigenschaften ist nicht immer nötig, manchmal sogar kontraproduktiv. Dieses Problem wird in der Literatur *Feature Selection* genannt.

new search (Title words) database introduction systems

There are 30 hits

1	An introduction to database systems / Date, C. J. 6. ed., [Nachdr.] Addison-Wesley 1997
2	An introduction to database systems / Date, C. J. 6. ed., repr. with corr., August 1995 Addison-Wesley 1995
3	An introduction to database systems / Date, C. J. 6. ed., 3. print Addison-Wesley 1995
4	An introduction to database systems / Date, C. J. 6. ed., reprinted with corr Addison-Wesley 1995
5	An introduction to database systems / Date, C. J. 6th ed., repr. with corr., march 1995 Reading, Mass [u.a.]: Addison-Wesley Pub. Co 1995
6	An introduction to database systems / Date, C. J. 6th ed., repr. with corr., Nov. 1994 Reading, Mass [u.a.]: Addison-Wesley Pub. Co 1995
7	An introduction to database systems / Date, C. J. 6. ed Addison-Wesley 1995
8	An introduction to database systems / Date, Chris J. 6. ed Addison-Wesley 1995
9	An introduction to spatial database systems / Gueting, R. 1994
10	An introduction to spatial database systems / Güting, Ralf Hartmut Fernuniv., Fachbereich Informatik 1994
11	Introduction to database and knowledge-base systems / Krishna, S. World Scientific 1992
12	Semantic database systems : a functional introduction / Prabhu, C. S. R. Sangam 1992
13	An introduction to database systems / Desai, Bipin C. West Publ. Co. 1992
14	Expert database systems : a gentle introduction / Beynon-Davies, Paul McGraw-Hill 1991
15	An Introduction to Database Systems Vol. 1 / Date, C.J Addison Wesley 1990
16	An Introduction to database systems / Desai, Bipin C. West Publ. Co. 1990
17	An introduction to database systems / Date, C. J. 5th ed Addison-Wesley 1990
18	Vol. 1: An Introduction to Database systems 4. ed Addison-Wesley 1986
19	An Introduction to Database systems / Date, C. J. Addison-Wesley 1983
20	An introduction to database systems / Date, C. J. 3. ed Addison-Wesley 1981
21	An Introduction to Database systems / Date, C.J. Addison-Wesley Publ. 1981
22	An Introduction to Database systems / Date, C. J. 2. Ed Addison-Wesley 1981
23	Vvedenie v sistemy baz dannych / Dejt, K. Nauka 1980
24	An introduction to database systems / Date, Christopher John 2. ed Addison-Wesley 1979
25	An introduction to database systems / Date, C. J. 2. Aufl Addison-Wesley Pub. Co 1977
26	An introduction to database systems / Date, C. 2nd ed. Addison-Wesley 1977
27	An Introduction to Database Systems Second Edition / Date, C. J. Addison-Wesley 1977
28	An introduction to database systems / Date, C. J. Addison-Wesley 1975
29	An introduction to database systems / Date, C. J. Addison-Wesley Pub. Co 1975
30	An introduction to database systems / Date, Chris J. Addison-Wesley

Tabelle 1: Die Trefferliste der Suchanfrage "Titelstichwörter =*introduction database systems*" von *GBVdirekt* (`http://www.gbv.de`), für die Detailansicht eines Datensatzes siehe Abbildung 5

ID	ISBN	Title	Name	Year	Pages
1	0-201-54329-X	An introduction to database systems	Date	1997	839
2	0-201-54329-X	An introduction to database systems	Date	1995	839
3	0-201-54329-X	An introduction to database systems	Date	1995	839
4	0-201-54329-X	An introduction to database systems	Date	1995	839
5	0-201-54329-X	An introduction to database systems	Date	1995	839
6	0-201-54329-X	An introduction to database systems	Date	1995	839
7	0-201-54329-X	An introduction to database systems	Date	1995	839
8	0-201-82458-2	An introduction to database systems	Date	1995	839
9		An introduction to spatial database systems	Gueting	1994	32
10		An introduction to spatial database systems	Güting	1994	13
11	981-020619-4	Introduction to database and knowledge-base systems	Krishna	1992	328
12	0-86311-346-X	Semantic database systems : a functional introduction	Prabhu	1992	157
13	0-314-66771-7	An introduction to database systems	Desai	1992	820
14	0-07-707240-5	Expert database systems : a gentle introduction	Beynon-Davies	1991	186
15	0-201-51381-1	An Introduction to Database Systems Vol. 1	Date	1990	
16	0-314-66771-7	An Introduction to database systems	Desai	1990	820
17	0-201-14439-5	An introduction to database systems	Date	1990	574
18	0-201-19215-2	Vol. 1: An Introduction to Database Systems	Date	1986	
19	0-201-14474-3	An Introduction to Database Systems	Date	1983	
20	0-201-14439-5	An Introduction to Database Systems	Date	1981	
21	0-201-14439-5	An introduction to database systems	Date	1981	574
22	0-201-14439-5	An Introduction to Database Systems	Date	1981	574
23		Vvedenie v sistemy baz dannych	Dejt	1980	463
24	0-201-01530-7	An introduction to database systems	Date	1979	536
25	0-201-14456-5	An introduction to database systems	Date	1977	536
26	0-201-14456-5	An introduction to database systems	Date	1977	536
27	0-201-14456-5	An Introduction to Database Systems Second Edition	Date	1977	
28		An introduction to database systems	Date	1975	
29	0-201-14452-2	An introduction to database systems	Date	1975	366
30		An introduction to database systems	Date		

Tabelle 2: Die mittels der Abbildungen $h_1, \ldots, h_5$ in die fünf gemeinsamen Eigenschaften transformierte Trefferliste (vgl. Tabelle 1)

folgt:

$$
\begin{aligned}
h_1 &:= Identity(\text{ISBN}),\\
h_2 &:= \texttt{Extract}\ lastName\ \texttt{From}\ \text{A.Verfasser}),\\
h_3 &:= \texttt{Extract}\ title\ \texttt{From}\ \text{A.Titel}),\\
h_4 &:= \texttt{Extract}\ year\ \texttt{From}\ \text{A.Erschienen})\quad \text{und}\\
h_5 &:= \texttt{Extract}\ pages\ \text{A.Umfang}).
\end{aligned}
$$

Für die Definition einer Vergleichsfunktion sollte man sich die Datensatzpaare einer Stichprobe genau ansehen: Fehlende Werte (*missing values*) explizit als Vergleichswert vorzusehen ist manchmal sinnvoll, bei der ISBN bietet es sich an, da sie durchaus fehlen kann. Für Vergleiche von Zeichenketten gibt es vielerlei Möglichkeiten, ein "Ähnlichkeitsmaß" zu definieren, wir werden bei dem Titel die übereinstimmenden Worte zählen. Die einfachste Form finden wir sonst: gleich/verschieden.

Die Vergleichsfunktion $f = (f_1, f_2, f_3, f_4, f_5)^{\mathrm{T}}$ könnte folgendermaßen definiert werden: en

$$
\begin{aligned}
f_1(ISBN_1, ISBN_2) &:= \begin{cases} 0 & : \quad ISBN_1 = ISBN_2 \\ 1 & : \quad ISBN_1 \text{ and } ISBN_2 \text{ are missing} \\ 2 & : \quad \text{otherwise} \end{cases}\\
f_2(name_1, name_2) &:= \begin{cases} 0 & : \quad name_1 = name_2 \\ 1 & : \quad \text{otherwise} \end{cases}\\
f_3(title_1, title_2) &:= \begin{cases} 0 & : \quad title_1 = title_2 \\ 1 & : \quad title_1 \neq title_2, \quad \text{but} \\ & \quad\;\; PortionSameWords(title_1, title_2) \geq 70\% \\ 2 & : \quad \text{otherwise} \end{cases}\\
f_4(year_1, year_2) &:= \begin{cases} 0 & : \quad year_1 = year_2 \\ 1 & : \quad \text{otherwise} \end{cases}\\
f_5(pages_1, pages_2) &:= \begin{cases} 0 & : \quad pages_1 = pages_2 \\ 1 & : \quad \text{otherwise} \end{cases}
\end{aligned}
$$

Bei der Prüfung auf Gleichheit zweier Strings in f_2 wird Groß- und Kleinschreibung nicht unterschieden und Umlaute sowie Sonderzeichen werden umgewandelt (d.h. es gilt ä=ae, ö=oe, ü=ue und ß=ss).

Die Funktion $PortionSameWords$ soll die Ähnlichkeit zweier Texte (hier des Titels) mittels der Übereinstimmung der enthaltenen Wörter messen. $PortionSameWords(text_1, text_2)$ ist definiert als der Quotient aus der Anzahl der in beiden Texten auftretenden Wörter und der kleineren Anzahl der in einem der Texte $text_i$ insgesamt enthaltenden Wörter (als Worttrennzeichen werden Leerzeichen benützt). Hierbei sollte man Artikel u.ä. nicht berücksichtigen. Da wir keine Wortliste verwenden wollen, zählen wir nur Wörter mit mindestens vier Buchstaben.

Der fünfdimensionale Vergleichsraum $R \subset \mathbb{R}^5$ bildet sich aus den Werten der Vergleichsfunktion $f = (f_1, f_2, f_3, f_4, f_5)^{\mathrm{T}}$:

$$R = \{0, 1, 2\} \times \{0, 1\} \times \{0, 1, 2\} \times \{0, 1\} \times \{0, 1\} \tag{9}$$

und hat also $72 = 3 \cdot 2 \cdot 3 \cdot 2 \cdot 2$ unterschiedliche Vergleichszustände. Der Vergleichsvektor $(0, 0, 0, 0, 0)^{\mathrm{T}}$ repräsentiert hier dann all jene Datensatzpaare, bei denen die fünf gemeinsamen Eigenschaften sämtlichst übereinstimmen.[13]

Step 2: Sampling

Nach dieser Vorarbeit können wir die Vergleichsvektoren auf einer Lernstichprobe ausrechnen. Zusätzlich notieren wir zu jedem Datensatzpaar[14], ob es sich auf dasselbe Buch (und Ausgabe) bezieht oder ob nicht.

Die Werte für die — zugegebenermaßen willkürlich — gezogene Stichprobe sind in der Tabelle 3 zu finden.Es handelt sich hier um eine Tabelle, die die Vergleichswerte der abgeleiteten Attribute (Title, Name ...) eines Datensatzpaares links und den (bekannten) Objekt-Bezug in der rechten Spalte als Klassifikationsattribut enthält.

Es sei bemerkt, daß für aussagekräftige Regeln eine repräsentative Stichprobe nötig ist, die aus den Ergebnissen mehrerer Suchanfragen gezogen wird. Es ist offensichtlich, daß bei 72 möglichen Vergleichsfällen eine Stichprobe von 24 Paaren zu klein ist.

Step 3: Learning/Classification

Anhand der ermittelten Werte kann man dann eine Partitionierung der Paare vornehmen, hierbei sind statistische Verfahren für die Diskrimination wie der Likelihood-Quotienten-Test (siehe Abschnitt 3.2), Entscheidungsbaumverfahren (C4.5 oder CART, siehe Abschnitt 3.3) oder der in dieser Arbeit nur angedachte auf *Rough Sets* basierende Ansatz denkbar (siehe Abschnitt 3.4).

Die folgende Regelableitung soll das Prinzip des überwachten Lernens (*supervised learning*) verdeutlichen.

[13] Insgesamt gibt es für unsere Trefferliste 17 Datensatzpaare mit dem Vergleichsvektor $\mathbf{0} = (0, 0, 0, 0, 0)^{\mathrm{T}}$, die sich alle auf dasselbe Buch beziehen, woran man schon ersehen kann, daß die oben konstruierten Transformationsabbildungen $h_1, \ldots, h_5$ recht gut gewählt sind. Sofern es Datensätze mit Vergleichswert $\mathbf{0}$ in der Stichprobe gibt, die sich auf verschiedene Bücher beziehen, sollte man tunlichst die gewählten gemeinsamen Eigenschaften sowie die Transformationsabbildungen überdenken.

[14] Da wir in unserem Beispiel eine Tabelle mit sich selbst abgleichen, werden nur die Datensätze $a_i, b_j \in A$ miteinander verglichen, für die $1 \leq i < j \leq 30$ gilt.

unit No.	pair (ID's) i	j	comparison values for the properties ISBN	Title	Name	Year	Pages	same book?
			a sample of the uncritical cases of matched pairs					
u_1	3	4	0	0	0	0	0	Yes
u_2	4	5	0	0	0	0	0	Yes
u_3	5	6	0	0	0	0	0	Yes
u_4	6	7	0	0	0	0	0	Yes
u_5	21	22	0	0	0	0	0	Yes
u_6	25	26	0	0	0	0	0	Yes
			a sample of the critical cases of matched pairs					
u_7	2	8	2	0	0	0	0	Yes
u_8	9	10	1	0	0	0	1	Yes
u_9	20	21	0	0	0	0	1	Yes
u_{10}	25	27	0	1	0	0	1	Yes
u_{11}	26	27	0	1	0	0	1	Yes
u_{12}	28	29	2	0	0	0	1	Yes
			a sample of the critical cases of unmatched pairs					
u_{13}	1	2	0	0	0	1	0	No
u_{14}	13	16	2	0	1	0	1	No
u_{15}	15	17	2	1	0	1	0	No
u_{16}	16	17	0	0	0	1	0	No
u_{17}	24	25	2	0	0	1	0	No
u_{18}	28	30	1	0	0	1	1	No
			a sample of the uncritical cases of unmatched pairs					
u_{19}	5	9	2	1	1	1	1	No
u_{20}	10	23	1	2	1	1	1	No
u_{21}	11	12	2	1	1	0	1	No
u_{22}	14	27	2	2	1	1	1	No
u_{23}	18	19	2	1	1	1	1	No
u_{24}	26	28	2	0	0	1	1	No

Tabelle 3: Die Tabelle mit den Vergleichswerten einer Stichprobe der Datensatzpaare der Tabelle 2. Der bekannte Objekt-Bezug (same book? Yes/No) ist für die Datensatzpaare in der rechten Spalte angegeben. Die Einträge " 0 0 1 0 0 " bezeichnen hier den Wert eines Datensatzpaares (a, b) unter der Vergleichsfunktion f (siehe Seite 19) Die Stichprobe wurde hier nicht zufällig gezogen, sondern speziell verlesen (daher die vier Gruppen mit je sechs Paaren)!

Durch scharfes Hinsehen lassen sich direkt aus der Tabelle 3 Regeln für die Entscheidung ablesen, ob sich zwei Datensätze auf ein Buch beziehen oder ob nicht. Es sind 9 Datensatzpaare (a_i, b_j) in der Stichprobe enthalten, bei denen sowohl die ISBN als auch das Erscheinungsjahr übereinstimmen (d.h. $f_1\big(h_1(a_i), h_1(b_j)\big) = 0$ und $f_4\big(h_4(a_i), h_4(b_j)\big) = 0$). Für alle 9 Paare hat die Variable *same book* den Wert *Yes*, weshalb wir folgende Regel angeben können:[15]

if `both ISBN and Year agree`
  then `entries refers to the same book`
  with confidence `100%`
  and support `75%.`

Diese Regel überrascht zwar nicht gerade, macht aber immerhin deutlich, wie Regeln ermittelt werden können. Um aussagekräftige Regeln zu erhalten, wird man nicht umhin kommen, die Stichprobengröße zu erhöhen, so daß für alle relevanten Vergleichswerte einige Fälle (Paare) auftreten.[16]

Eine andere aus der Tabelle ablesbare Regel ist:

if `both Pages and Year agree`
  then `entries refers to the same book`
  with confidence `100%`
  and support `58%.`

Das Entscheidungsbaumverfahren C4.5 verwenden

Wir wollen das auf im Abschnitt 3.3 eingeführte Verfahren anwenden. In unserem Fall hat die Zufallsvariable X, *same book?* die zwei Werte $x = Yes$ und $x = No$, so daß wir die Entropie für eine Menge von Beobachtungen mit (8) auf Seite 13 nach der Formel

$$E = -\big(p_0 \cdot \log_2 p_0 \ + \ p_1 \cdot \log_2 p_1\big), \tag{10}$$

berechnen können, wobei $p_0 := P\big(X(\omega) = Yes\big)$ und $p_1 := P\big(X(\omega) = No\big)$ bezeichnen.

Die 24 Beobachtungen der Stichprobe in Tabelle 3) teilen sich je hälftig auf die beiden Werte *same/not same book*, weshalb die Wahrscheinlichkeiten sich unter der Gleichverteilungsannahme wie folgt berechnen:

$$p_0 = \frac{12}{24} = \frac{1}{2}, \quad p_1 = \frac{12}{24} = \frac{1}{2}.$$

[15] Für die Notation der Regeln sei auf den Abschnitt 3.1 verwiesen.

[16] Es liegt in der Natur der Dinge, daß es Vergleichswerte geben kann, die überhaupt nicht vorkommen. Der Wert $(0, 2, 1, 0, 0)^{\mathrm{T}}$ sei dafür ein Beispiel: die Werte der Eigenschaften ISBN, Year und Pages sind identisch, während Author und Title jeweils verschieden sind.

Damit erhalten wir für die Entropie $E(0)$ der gesamten Stichprobe

$$\begin{aligned} E(0) &= -\bigl(p_0 \cdot \log_2 p_0 + p_1 \cdot \log_2 p_1\bigr) = -\Bigl(\frac{1}{2} \cdot \log_2 \frac{1}{2} + \frac{1}{2} \cdot \log_2 \frac{1}{2}\Bigr) \\ &= -\Bigl(-\frac{1}{2} + \bigl(-\frac{1}{2}\bigr)\Bigr) = 1.00. \end{aligned}$$

Nun suchen wir nach einem Attribut, bei dem eine binäre Partitionierung der Domain die größte Verringerung der Entropie erzielt. Generell läßt sich sagen, daß die Verminderung groß ist, wenn die Variation der Variable *same book* in jeder Partition gering ausfällt. Wenn innerhalb einer Partition die Variable für alle Paare den gleichen Wert hat, also keine Unordnung vorhanden ist, ist die Entropy gleich Null.

Wenn wir die Tabelle 3 genauer unter die Lupe nehmen, ist die Partitionierung für das Attribut `Year` recht aussichtsreich, da für `Year`=1 10 negative Beispiele vorhanden sind, alle 12 positiven Beispiele und die restlichen 2 negativen haben den Vergleichswert `Year`=0. Damit ergibt sich

$$\begin{aligned} E\bigl(\texttt{Year}, (\{0\},\{1\})\bigr) &= -\Bigl[\frac{14}{24}\Bigl(\frac{12}{14}\log_2\frac{12}{14} + \frac{2}{14}\log_2\frac{2}{14}\Bigr) + \underbrace{\frac{10}{24}\Bigl(\frac{10}{10}\log_2\frac{10}{10}\Bigr)}_{=0}\Bigr] \\ &= -0.58\bigl(-0.19 + (-0.40)\bigr) = 0.35. \end{aligned}$$

Also

$$E\bigl(0\bigr) - E\bigl(\texttt{Year}, (\{0\},\{1\})\bigr) = 0.65.$$

Die Zerlegung bei `Year` liefert die größte Verringerung der Entropie, weshalb diese ausgewählt wird. Weiter können wir aus der Tabelle 3 ablesen, daß `Name` das einzige Attribut ist, bei dem die positiven Fälle von den negativen getrennt werden: Die beiden negativen Fälle u_{14} und u_{21} haben den Wert 1, während alle positiven Fälle u_1 bis u_{12} den Wert 0 haben (für `Name`). Damit ist die Verminderung der Entropy maximal, sie ist für die gesamte Partitionierung gleich Null. Wir erhalten damit den Entscheidungsbaum in Abbildung 6. Aus dem Entscheidungsbaum lassen sich folgende vier Regeln ablesen:

if `Year = 1` **then** `not same book` **with confidence** `100%` **and support** `83%.`	**if** `(Year = 0 ∧ Name = 0)` **then** `same book` **with confidence** `100%` **and support** `100%.`
if `Year = 0` **then** `same book` **with confidence** `86%` **and support** `100%.`	**if** `(Year = 0 ∧ Name = 1)` **then** `not same book` **with confidence** `100%` **and support** `17%.`

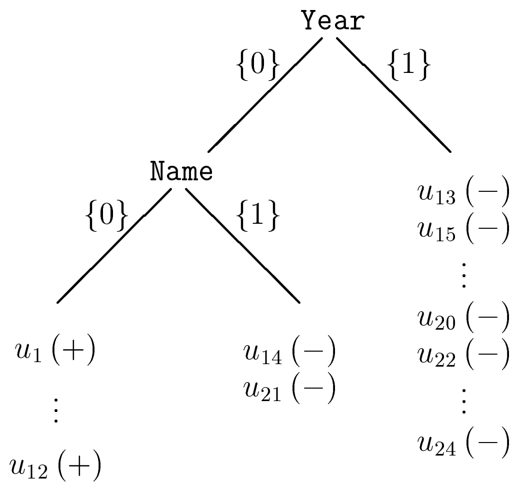


Abbildung 6: Der Entscheidungsbaum für die Stichprobe aus Tabelle 3 (u_i bezeichnet die Nummer des Paares in der Tabelle; (+/−) bezeichnet den Objektbezug *same/not same book*)

Steps 4 + 5: Validation & Integration

Die letzten beiden Schritte des Datenintegrations-Prozesses, die Validierung und die eigentliche Integration der Datenquellen, bilden dann den Abschluß.

Für die Validierung prüfe man beispielsweise die gewonnenen Regeln auf Treffgenauigkeit mit einer neuen Test-Stichprobe.

Schließlich fassen wir alle Datensätze der Datenbank zusammen, die sich auf eine Ausgabe eines Buches beziehen (z.B. mit Hyperlinks). Dieses kann auch als Bereinigung der Trefferliste um Duplikate in Folge einer erneuten Suchanfrage geschehen — die Klassifikationsregeln sind ja bekannt.

5 Zusammenfassung und Ausblick

Wir haben gesehen, wie sich der Ansatz der Objekt-Identifikation für das Erkennen semantoisch äquivalenter Daten-Objekte einsetzen läßt. Von Bedeutung ist dabei die Ermittlung der korrespondierenden Attribute zweier Datenquellen, deren Werte miteinander verglichen werden können. Es wurde ausgeführt, daß anhand dieser Vergleichswerte entschieden werden kann, ob zwei Datensätze sich auf dasselbe Objekt beziehen oder ob nicht. Verschiedene Statistische und Data Mining Methoden können dafür eingesetzt werden. Die Fallstudie verdeutlicht die praktische Umsetzung dieses Ansatzes.

In der Zukunft ist zu prüfen, welche Methoden sich für die Klassifikation der Vergleichswerte eignen. Die Generierung geeigneter Test-Datensätze ist ein geignetes Hilfsmittel, um die Fehlerraten der Verfahren miteinander vergleichen zu können.

Weitere Forschung wird sich den Eigenschaften und Charakteristika der ableitbaren Attribute und der Konstruktion von Vergleichsfunktionen widmen. Läßt sich dieser Teil des Datenintegrations-Prozesses — zumindest teilweise — automatisieren? Technische Aspekte wie die Speicherung von Zwischenergebnissen und die Repräsentation der Konversions- und Vergleichsabbildungen müssen auch geklärt werden. Nicht zuletzt spielt für viele Anwendungen der Zeitfaktor eine große Rolle, es ist in der Regel undenkbar, jeden Datensatz der einen mit jedem Datensatz der anderen Quelle zu vergleichen. Läßt sich die Komplexität dieser Prozedur verringern, ohne hohe Fehlerraten in Kauf zu nehmen? Für Echtzeit-Anwendungen wie Anfragebearbeitungen stellt sich zudem die Frage, wie innerhalb kürzester Zeit die passenden Einträge gefunden werden. Dazu werden Mechanismen nötig sein, die die (vorher ermittelten) Regeln über Zusammenhänge effektiv umsetzen.

Wichtig ist auch die praktische Umsetzung mit "echten Daten", geplant ist die Modellierung des Prozesses für einen registerbasierten Zensus in Deutschland. Die Integration von Internet-Quellen scheint auch zukünftig große Bedeutung zu erlangen — die Verwendung von XML, der eXtended Markup Language des W3-Konsortiums, ermöglicht die Verknüpfung von *strukturierten* Dokumenten im Internet. Die Erkennung der Objekt-Identität stellt ein Problem dar, das auch in diesem Zusammenhang gelöst werden muß.

Literatur

Alvey, W. and B. Jamerson (Eds.) (1997). *Record Linkage Techniques — 1997. Proceedings of an International Workshop and Exposition. March 20-21, 1997 in Arlington, Virginia*, Washington, DC. Federal Committee on Statistical Methodology, Office of Management and Budget; c. f. http://www.census.gov/srd/www/reclink/reclink.html.

Bishop, Y. M. M., S. E. Fienberg, and P. W. Holland (1975). *Discrete Multivariate Analysis. Theory and Practice*. Cambridge: MIT Press.

Bornhövd, C. and A. P. Buchmann (1999). A prototype for metadata-based integration of internet sources. In *Advanced Information Systems Engineering (CAiSE '99)*, pp. 439–445. Springer-Verlag.

Fellegi, I. P. and A. B. Sunter (1969). A theory of record linkage. *Journal of the American Statistical Association 64*, 1183–1210. Reprinted in Kilss and Alvey (1985, 51–78).

Huang, T. and S. Russell (1998). Object identification: a bayesian analysis with application to traffic surveillance. *Artificial Intelligence 103*, 77–93.

Huck, G., P. Fankhauser, K. Aberer, and E. Neuhold (1998). Jedi: Extracting and synthesizing information from the web. In *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems (CoopIS'98), NY, August 1998*, pp. 32–43. download at http://xml.darmstadt.gmd.de/.

Kashyap, V. and A. Sheth (1996). Semantic and schematic similarities between database objects: a context-based approach. *The VLDB Journal 5*, 276–304.

Kilss, B. and W. Alvey (Eds.) (1985). *Record Linkage Techniques — 1985. Proceedings of the Workshop on Exakt Matching Methodologies in Arlington, Virginia May 9–10, 1985*, Internal Revenue Service Publication, Washington, DC. Department of the Treasury, Statistics of Income Division; download at http://www.bts.gov/fcsm/methodology (a 29 MByte pdf-file).

Liu, C. and D. B. Rubin (1994). The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *Biometrika 81*(4), 633–48.

Meng, X.-L. and D. B. Rubin (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika 80*(2), 267–78.

Mood, A. M., F. A. Graybill, and D. C. Boes (1974). *Introduction to the Theory of Statistics*. McGraw-Hill series in probability and statistics. Tokyo: McGraw-Hill.

Müller, T. (1997). Identifikation semantisch äquivalenter Datenbankobjekte in Multidatenbanksystemen. diploma thesis, TU Magdeburg. in german; download as ps-file at http://wwwiti.cs.uni-magdeburg.de/iti_db/diplomarbeiten.html.

Munakata, T. (1998). *Fundamentals of the New Artificial Intelligence.* New York: Springer-Verlag.

Neiling, M. (1998). Data Fusion with Record Linkage. In *3. Workshop "Föderierte Datenbanken" Magdeburg 1998.* only included in the online-proceedings at http://wwwiti.cs.uni-magdeburg.de/fdb98/online-proc/.

Newcombe, H. B., J. M. Kennedy, S. J. Axford, and A. P. James (1959). Automatic linkage of vital records. *Science 130*, 954–959. Reprinted in Kilss and Alvey (1985, 7–12).

Pawlak, Z. (1982). Rough sets. *Int. J. Computer and Information Science 11*, 341–356.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects and Reasoning about Data.* Dordrecht, Netherlands: Kluwer Academic.

Predki, B., R. Slowinnski, J. Stefanowski, R. Susmaga, and S. Wilk (1998). Rose — software implementation of the rough set theory. In *Rough Sets and Current Trends in Computing*, pp. 605–608. Springer-Verlag.

Quinlan, J. R. (1993). *Q4.5: Programs for Machine Learning.* Morgan Kaufmann.

Schürmann, J. (1996). *Pattern Classification.* New York: John Wiley & Sons.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Analysis.* Chichester; New York et al.: John Wiley & Sons.

Winkler, W. E. (1993). Improved decision rules in the Fellegi-Sunter model of record linkage. The Research Report Series RR-93/12, U.S. Bureau of the Census. http://www.census.gov/srd/www/byyear.html.

Winkler, W. E. (1995). Matching and record linkage. In B. G. Cox (Ed.), *Business Survey Methods*, pp. 355–384. New York: J. Wiley. Reprinted in Alvey and Jamerson (1997, 374–403).