Dublettenbereinigung nach dem Record Linkage Algorithmus

Dipl.-Wirtschaftsinformatiker (FH) Sönke Cordts,
Prof. Dr. Burkhard Müller

Wirtschaftsinformatik
Fachhochschule Westküste, BWL
Fritz-Thiedemann-Ring 20
25746 Heide
cordts@fh-westkueste.de
mueller@fh-westkueste.de

Abstract: Unter Dublettenbereinigung versteht man das Entfernen mehrfach gespeicherter Datensätze, die auf dasselbe Objekt verweisen. Der bekannteste Algorithmus hierzu ist der Record Linkage Algorithmus nach Fellegi und Sunter. Hierbei wird ein Gesamtgewicht auf Grundlage eines Vergleichs einzelner Attribute von zwei Datensätzen errechnet. Neben dem exakten Vergleich von Attributen sind vor allem Algorithmen notwendig, die orthographische oder typographische Fehler berücksichtigen.

1 Grundlagen zur Dublettenbereinigung

1.1 Problemstellung

Bei der Dublettenbereinigung geht es um das Entfernen mehrfach gespeicherter Datensätze, die auf dasselbe Objekt verweisen. Folgende Beispieldaten sollen das Problem verdeutlichen:

Satznr	Name	Strasse	Plz	Ort	Geschlecht	Alter
1	Hilke Friedrichs	Kabelstr. 3	25746	Heide	w	38
2	Hillie Friedrichs	Gabelstr. 3	25764	Haide	w	37
3	H. Franzen	Kabelstr. 3	25746	Heite	m	40

Tabelle 1: Beispieldaten zur Erläuterung der Problemstellung der Dublettenbereinigung

Die drei Datensätze verweisen alle auf dieselbe reale Person, jedoch werden durch orthographische Fehler („Kabelstr.“ - „Gabelstr.“, „Haide“ - „Heide“), inkorrekte Angaben (Geschlecht und Alter) usw. die Datensätze nicht als identisch angesehen. Um solche Unterschiede in den Werten zu ermitteln, sind Algorithmen notwendig, die über eine Maßzahl die Ähnlichkeit zwischen zwei Werten errechnen. Für numerische Attribute kann z.B. die Differenz oder die euklidische Distanz zwischen zwei Werten ermittelt werden.

Für nominale Daten sind Algorithmen notwendig, die ein Ähnlichkeits- oder Distanzmaß für zwei Zeichenketten berechnen bzw. die phonetische Übereinstimmung zwischen zwei Zeichenketten ermitteln. Einfache Ansätze verwenden hierbei Wildcards oder Teilzeichenketten. Sinnvoller ist jedoch die Berechnung eines Proximätsmaßes über Approximate String Algorithmen oder die Ermittlung phonetischer Codes. [RA00] In das Ähnlichkeitsmaß sollten auch die Häufigkeiten von Werten über eine Gewichtung einfließen. Zum Beispiel sollte ein Nachname, der sehr selten im Kundendatenbestand vorkommt, eine höhere Gewichtung bekommen als ein häufig vorkommender. Hat man für vorher festgelegte Attribute die Ähnlichkeitsmaße für zwei Datensätze errechnet, so kann für diese mit Hilfe der Gewichtungen ein Gesamtgewicht ermittelt werden. Je höher das Gesamtgewicht, desto ähnlicher sind sich die zwei Datensätze. Bevor auf den bekanntesten Algorithmus zur Dublettenbereinigung – den Record Linkage Algorithmus nach Fellegi und Sunter – eingegangen wird, sollen zunächst die Grundlagen zur Berechnung von Proximitätsmaßen zwischen zwei Zeichenketten erläutert werden.

1.2 String Matching Algorithmen

Die einfachste Möglichkeit, Ungenauigkeiten zwischen zwei Zeichenketten zu berücksichtigen, besteht darin, nur eine Teilzeichenkette zu vergleichen oder Wildcards [SC01] zu verwenden. Stimmt die Teilzeichenkette überein, so wird davon ausgegangen, dass die beiden Zeichenketten ähnlich sind. Exact String und Wildcard Matching sind sinnvoll, wenn bestimmte Zeichenfolgen stabil bleiben. Da dies eher die Ausnahme ist, sind Algorithmen notwendig, die vor allem orthographische und typographische Fehler berücksichtigen. Diese bezeichnet man als Approximate String Matching [NA98] Algorithmen. Hierbei geht es um die Berechnung der Ähnlichkeit zwischen zwei Zeichenketten, indem man Proximitätsmaße berechnet, die die Distanz oder die Ähnlichkeit zwischen zwei Zeichenketten widerspiegeln. Das bekannteste Distanzmaß zweier Zeichenketten ist die Levenshtein Distanz nach dem gleichnamigen russischen Mathematiker aus dem Jahr 1966. [BI03b] Levenshtein berechnet die geringste Anzahl an Zeichenoperationen (Einfügen, Löschen, Austausch), die notwendig sind, um eine Zeichenkette in die andere zu transformieren. Für solche Distanzmaße gilt, je größer das Maß, umso verschiedener die beiden Zeichenketten.

Ähnlichkeitsmaße sind dagegen umso größer, je mehr sich zwei Zeichenketten ähneln. Das bekannteste Ähnlichkeitsmaß ist das Jaro-Maß, das die Anzahl und die Reihenfolge übereinstimmender Zeichen berücksichtigt, wobei diese sich nicht an der gleichen Position befinden müssen. Das Jaro-Maß berechnet die Ähnlichkeit zwischen zwei Zeichenketten ausschließlich zeichenweise. Diese Vorgehensweise ist insbesondere für kurze Zeichenketten bzw. einzelne Wörter sinnvoll. Für längere Zeichenketten, die aus mehreren Wörtern bestehen, erreicht man bessere Ergebnisse durch token-basierte Algorithmen, die die Zeichenkette vor der Berechnung des Ähnlichkeitsmaßes in einzelne Wörter aufteilen. Die einfachste token-basierte Maßzahl ist das Jaccard-Maß. [BI03b] Die token-basierten Maßzahlen haben allerdings den Nachteil, nicht zeichenweise zu überprüfen. Deshalb gibt es neben zeichen- und token-basierten auch hybride Maßzahlen (z.B. Monge-Elkan), die eine Kombination aus beiden darstellen. [CO03]

Unterschiedlich geschriebene Wörter erzeugen aufgrund gleicher phonetischer Aussprache gleiche Schallwellen. So klingen z.B. die Nachnamen Meyer, Maier oder Meier in der deutschen Sprache identisch. Phonetische Algorithmen erzeugen deshalb identische Zeichenfolgen für gleichklingende Wörter. Da die Aussprache von Wörtern sprachabhängig ist, gelten phonetische Algorithmen i.d.R. nur für eine bestimmte Landessprache und eine bestimmte Domäne. Sie sind also domänenabhängig. [PF94] Der bekannteste phonetische Algorithmus von M. Odell und R. Russell, der 1900 in den USA entwickelte Soundex-Algorithmus, ersetzt mehrere Buchstaben, die aufgrund der Lautbildung ähnlich klingen, z.B. m und n, durch eine gemeinsame Zahl und kürzt diesen Code auf 4 Zeichen. Wendet man den Algorithmus auf die Nachnamen Meyer, Maier und Meier an, so erhält man als Soundex-Code die Zeichenfolge „M600“.

2 Record Linkage nach Fellegi und Sunter

1962 führte H. Newcombe Entscheidungsregeln zum Ableiten von übereinstimmenden (Matches) und nicht-übereinstimmenden Datensätzen (Unmatches) ein, um mehrfach gespeicherte Datensätze zu erkennen. Die formalen mathematischen Grundlagen, die auf der Wahrscheinlichkeitstheorie basieren, wurden hierzu 1969 von I. Fellegi und A. Sunter beschrieben. [FS69] An folgenden Beispieldaten sollen die Grundlagen des Record Linkage erläutert werden.

Satznr	Vorname	Nachname	Strasse	Plz	Ort	Geschlecht
1	Hans	Friedrichs	Kabelstr. 3	25746	Heide	m
2	Hans	Friedrichs	Gabelstr. 3	25746	Heide	m
3	Greta	Haffner	Fritzweg 2	22761	Hamburg	w
4	Marie	Haller	Hohlgang 2	22529	Hamburg	w
5	Karl	Friedrichs	Fritzstr. 1	22761	Hamburg	m

Tabelle 2: Beispieldaten zur Veranschaulichung des Record Linkage Verfahrens

Um zu überprüfen, ob ein Datensatz im Datenbestand mit einem beliebigen anderen übereinstimmt, müsste normalerweise jeder Datensatz mit jedem anderen verglichen werden. Für das obige Beispiel wären 10 unterschiedliche Vergleiche notwendig [ELoJ]. Bei nur 1.000 Datensätzen sind es etwa 500.000 Vergleiche und bei nur 10.000 Datensätzen bereits 50 Mio. Da der Berechnungsaufwand bei großen Datenbeständen somit unpraktikabel ist, wurde das Konzept des Blocking eingeführt, um die Anzahl der Vergleiche zu reduzieren. Dabei werden ein oder mehrere Attribute als Blockingvariable ausgewählt. Datensätze mit gleichem Wert in der Blockingvariable werden zusammengefasst. Vergleiche werden dann nur noch innerhalb der zu diesem Block gehörenden Datensätze vorgenommen. Verwendet man im obigen Beispiel das Attribut Geschlecht zum Blocking, so entstehen zwei Blöcke mit je 2 und 3 Datensätzen. Die Anzahl der Vergleiche reduziert sich auf 4.

Um nun die Ähnlichkeit zwischen 2 Datensätzen zu berechnen, werden, wie oben beschrieben, verschiedene Vergleichsalgorithmen für jedes zu überprüfende Attribut verwendet. Im einfachsten Fall werden die beiden zu vergleichenden Werte eines Attributs auf exakte Gleichheit getestet. Nach Newcombe geht nun jedes zu überprüfende Attribut mit einem Gewicht in das Gesamtgewicht ein, das bestimmt, ob die Datensätze zur Gruppe der Matches oder Unmatches gehören. Die Gewichte basieren auf Häufigkeitsverhältnissen, die idealerweise anhand eines vorhandenen Datenbestandes ermittelt werden, für den die Matches und Unmatches bereits manuell klassifiziert wurden. Um die einzelnen Gewichte zu einem Gesamtgewicht zu addieren, werden sie über den Logarithmus zur Basis 2 transformiert.

$$G_{Agree} = \log_2(M / U)$$

mit

G_{Agree}	Gewicht bei Übereinstimmung
M	relative Häufigkeit der Übereinstimmung in der Gruppe Match
U	relative Häufigkeit der Übereinstimmung in der Gruppe Unmatch

Angenommen für das obige Beispiel hat sich bei einem vergleichbaren repräsentativen Datenbestand ergeben, dass der Nachname in 93% der Gruppe Matches übereinstimmte und nur in 2% der Gruppe Unmatches. Beim Geschlecht waren es 95% bei der Gruppe Matches und 50% bei der Gruppe Unmatches. Auf Basis dieser Häufigkeitsverhältnisse berechnen sich die Gewichte für eine Übereinstimmung (G_{Agree}) wie folgt [OA04]:

$$G_{Agree}(\text{Nachname}) = \log_2(0{,}93 / 0{,}02) = 5{,}54$$
$$G_{Agree}(\text{Geschlecht}) = \log_2(0{,}95 / 0{,}50) = 0{,}93$$

Analog berechnen sich die Gewichte für eine Nichtübereinstimmung ($G_{Disagree}$): [GI01]

$$G_{Disagree} = \log_2(1 - M / 1 - U)$$

mit

$G_{Disagree}$	Gewicht bei Nichtübereinstimmung

Als Gewichte bei Nichtübereinstimmung der beiden Attribute Nachname und Geschlecht ergeben sich damit also:

$$G_{Disagree}(\text{Nachname}) = \log_2((1–0{,}93) / (1–0{,}02)) = -3{,}81$$
$$G_{Disagree}(\text{Geschlecht}) = \log_2((1–0{,}95) / (1–0{,}50)) = -3{,}32$$

Für das Beispiel sollen die folgenden Gewichte gelten:

	Vorname	**Nachname**	**Plz**
G_{Agree}	4,50	5,54	2,50
$G_{Disagree}$	-2,00	-3,81	-1,00

Tabelle 3: Gewichte für Vorname, Nachname, Plz

Im vorliegenden Beispiel soll eine Überprüfung der jeweiligen Datensatzpaare für die Attribute Vorname, Nachname und Plz erfolgen. Als Blockingvariable wird das Geschlecht verwendet. Die Attribute Ort und Strasse werden ignoriert. Die drei zu überprüfenden Attribute werden auf exakte Gleichheit getestet. Exemplarisch ergeben sich bei einem Vergleich zwischen den Datensatzpaaren mit den Satznummern 1 und 2 und den Satznummern 3 und 4 folgende Gesamtgewichte:

G(Satznummer=1, Satznummer=2) = 4,50 + 5,54 + 2,50 = 12,54
G(Satznummer=3, Satznummer=4) = -2,00 - 3,81 -1,00 = -6,81

Das Gesamtgewicht kann als Wahrscheinlichkeit der Übereinstimmung zwischen zwei Datensätzen interpretiert werden. Demnach ist das Datensatzpaar 1 und 2 wahrscheinlich identisch, und das Datensatzpaar 3 und 4 wahrscheinlich nicht identisch. Nachdem für die Datensätze der Blöcke die Gesamtgewichte ermittelt wurden, muss entschieden werden, ob das jeweilige Datensatzpaar zur Gruppe Matches oder Unmatches gehört. Datensatzpaare, für die eine Entscheidung zu einer der beiden Gruppen nicht eindeutig möglich ist, werden einer weiteren Gruppe Possibles zugeordnet Zur Entscheidung müssen zwei Schwellwerte, „upper threshold“ und „lower threshold“, festgelegt werden, über die eine Zuordnung erfolgt. Für das konstruierte Beispiel soll als „upper threshold“ der Wert 9,0 und als „lower threshold“ der Wert 1,0 festgelegt werden. Damit ergeben sich für alle zu vergleichenden Datensatzpaare folgende Zuordnungen zu den Gruppen:

Datensatzpaar	Gesamtgewicht	Gruppe
1 und 2	12,54	Matches
1 und 5	2,54	Possibles
2 und 5	2,54	Possibles
3 und 4	-6,81	Unmatches

Tabelle 4: Gesamtgewichte und Klassifizierungen der Datensaatzpaarvergleiche

Literaturverzeichnis

[BI03b] Bilenko, M.: Learnable Similarity Functions and Their Applications to Record Linkage and Clustering – Doctoral Dissertation Proposal, University of Texas, Austin, 2003

[CO03] Cohen, W. u.a.: A Comparison of String Distance Metrics for Name-Matching Tasks, America Assocation for Artificial Intelligence, Menlo Park, 2003

[ELoJ] Elfeky, M. u.a.: Towards Quality Assurance of Government Databases: A Record Linkage Web Service, Purdue University, West Lafayette, o.J.

[FS69] Fellegi, I.; Sunter, A.: A Theory for Record Linkage, Journal of the American Statistical Assocation Vol. 64 No. 328, American Statistical Assocation, o.A., 1969

[GI01] Gill, L.: Methods for Automatic Record Matching and Linkage and their Use in National Statistics, National Statistics Methodological Series No. 25, Oxford, 2001

[NA98] Navarro, G.: Approximate Text Searching, University of Chile, Chile, 1998

[OA04] o.A.: LinkageWiz User Manual – Version 3.5, LinkageWiz Software, Payneham, 2004

[PF94] Pfeifer, U. u.a.: Searching Proper Names in Databases, University of Dortmund, Dortmund, 1994

[RA00] Rahm, E.; Hai Do, H.: Data Cleaning: Problems and Current Approaches, Data Engineering IEEE Computer Society, Vol. 3 No. 4, o.A., 2000

[SC01] Schöning, U.: Algorithmik, Spektrum Akademischer Verlag, Heidelberg, 2001