Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik

Entity-Resolution für Zitationen mit Probabilistischen Relationalen Modellen

Diplomarbeit im Studiengang Diplom-Informatik

Eingereicht von: Gunar Maiwald

Gutachter:
Prof. Dr. Tobias Scheffer, Institut für Informatik
Prof. Dr. Hans-Dieter Burkhard, Institut für Informatik

Betreuer:
Dipl.-Inf. Ulf Brefeld

Berlin, den 1.Juni 2007

Zusammenfassung

In dieser Diplomarbeit wird untersucht, wie auf der Basis von Literaturreferenzen ein Zitationsgraph durch ein automatisches Verfahren aufgebaut werden kann. Zur Lösung des Problems werden Probabilistische Relationale Modelle herangezogen. Eine problemspezifische Erweiterung des Modells ermöglicht es, dass bestehende Unsicherheiten im Zitationsgraphen mit Hilfe eines Inferenzverfahrens aufgelöst werden können. Zur Evaluierung des Verfahren werden Experimente auf dem Cora-Datensatz durchgeführt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Bei der Erstellung einer wissenschaftlichen Arbeit zu einem bestimmten Forschungsthema stellt die Suche nach themenrelevanter Literatur einen wichtigen Teil dar. Dabei stellt die Literatur, die von anderen Forschern zum selben Thema verwendet und zitiert wurde oftmals einen guten Einstieg dar, da diese von anderen Autoren als relevant für ihre Arbeit erachtet wurde. Der Verweis einer wissenschaftlichen Arbeit auf eine andere Publikation wird dabei als Zitation bezeichnet. Auf der Basis mehrerer Zitationen lässt sich eine Verweisstruktur, ein sogenannter Zitationsgraph erzeugen. In diesem werden Publikationen durch Knoten repräsentiert und Zitationen als Kanten zwischen den Knoten. Die Kanten sind dabei gerichtet und zeigen von der zitierenden auf die zitierte Publikation. Ein Beispiel für einen Zitationsgraph ist in Abbildung 1.1 dargestellt.

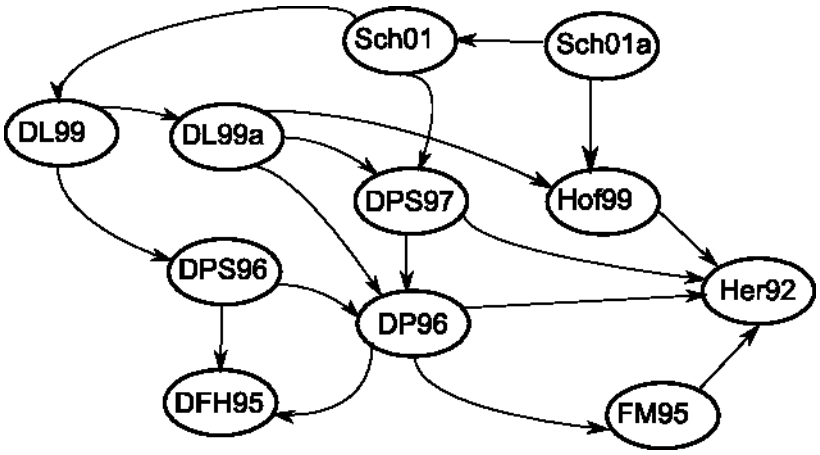

Abbildung 1.1: Beispiel eines Zitationsgraphen auf elf Publikationen

Auf Basis eines Zitationsgraphen lassen sich viele Untersuchungen durchführen. So kann z.B. das Publikationsverhalten von Autoren abgelesen sowie Communities sich häufig gegenseitig zitierender Forscher gefunden werden. Ebenso kann mit Hilfe eines Zitations-

graphen relevante Literatur ausfindig gemacht werden. Garfield [8] ging sogar so weit zu behaupten, man könne damit mögliche Nobelpreisträger vorhersagen.

Die Generierung eines solchen Zitationsgraphen ist für einen kleinen Korpus von Artikeln problemlos von Hand zu bewerkstelligen. Die Zahl wissenschafticher Publikationen, die Monat für Monat herausgegeben werden, ist jedoch unüberschaubar groß. So wächst beispielsweise PubMed, eine medizinische Artikeldatenbank, jährlich um 600.000(!) Artikel. Eine solch große Menge kann effektiv nur von einem automatischen Verfahren verarbeitet werden. Für ein solches Verfahren müssen die Artikel in einem maschinenlesbaren Textformat vorliegen. Das Verfahren extrahiert zunächst die Metadaten eines jeden Artikels - es handelt sich dabei um die bibliographischen Angaben - sowie jede einzelne Literaturreferenz. Anschließend werden die Referenzen analysiert und deren Metadaten extrahiert. Den letzten Schritt bildet die Gruppierung von Referenzen und Artikeln auf Basis ihrer Metadaten In dieser Arbeit wird der letzte Schritt genauer untersucht. Die Aufgabe Publikationen und die auf sie verweisenden Zitationen korrekt zu gruppieren, ist eine Instanz der Entity Resolution.

Ein Problem, dem sich ein Verfahren gegenübersieht ist die Vielzahl an Zitationsstilen. Selbst wenn für jeden dieser Stile ein Schema gelernt werden kann, so muss beim Auftreten eines neuen Zitationsstils das Verfahren erweitert werden. Ein weiteres Problem liegt in der Fehlerhaftigkeit maschineller Texterkennung, dieses kann durch kein deterministisches Verfahren gut gelöst werden. Hinzu kommen fehlerhafte oder unvollständige Angaben der Autoren. Daher stellt ein probabilistischer Ansatz eine interessante Alternative dar.

Ausgehend von einem Korpus, welcher Artikel und Referenzen zwischen den Artikeln enthält, soll untersucht werden, ob sich auf Basis probabilistischer Analyse dieses Korpus ein Zitationsgraph generieren läßt. Ziel der Arbeit ist es, ein Modell zu entwickeln, welches zur Generierung eines Zitationsgraphen verwendet werden kann. Darüberhinaus soll ein auf diesem Modell operierendes Inferenzverfahren entwickelt werden, daß den Zitationsgraphen erzeugt.

Im Anschluß an die Einleitung wird in Kapitel 2 das Problem der Entity Resolution genauer beschrieben und es wird in Kapitel 3 ein Überblick über verwandte Arbeiten gegeben. Kapitel 4 führt zum einen das Probabilistische Relationale Modell ein, das in dieser Arbeit zur Lösung des Problems herangezogen wird. Es werden MCMC-Algorithmen vorgestellt, die auf dem beschriebenen Modell operieren und Inferenzen auf Basis gelernter

Daten ermöglichen. In Kapitel 5 wird das Problem der Entity Resolution für Zitationen analysiert und ein Modell sowie ein Inferenzverfahren vorgestellt. Im Anschluß daran werden in Kapitel 6 Experimente zur Evaluierung des Vefahrens beschrieben und die Ergebnisse vorgestellt. Den Abschluß der Arbeit bildet die Zusammenfassung in Kapitel 7, die darüberhinaus einen Ausblick auf mögliche Erweiterungen von Modell und Verfahren gibt.

2 Problemstellung

In diesem Kapitel wird das Problem der Entity-Resolution für Zitationen formalisiert. Zunächst wird anhand eines Beispiels gezeigt, wie sich die Schwierigkeit des Problems äussert. Anschließend wird der eingangs erwähnte Zitationsgraph definiert. Es wird gezeigt, dass die Generierung eines solchen Graphen als Entity-Resolution angesehen werden kann. Es werden mehrere Kriterien vorgestellt, um die Qualität eines konstruierten Zitationsgraphen zu messen.

Das Problem der Entity-Resolution für Zitationen soll anhand eines Beispiels näher gebracht werden. In Abbildung 2.1 erkennt man, daß ein und dieselbe Publikation auf sehr unterschieldiche Art und Weise zitiert werden kann. Die aufgeführten Zitationen beschreiben alle denselben Artikel unterscheiden sich dabei jedoch in der Art Autorennamen, Band- und Heftangaben darzustellen sowie in der Vollständigkeit der Angaben.

```
Davenport, T., DeLong, D., and Beers, M. "Successful knowledge management
projects," Sloan Management Review (39:2)

Davenport, T., DeLong, D., & Beers, M. (1998) Successful knowledge management
projects. Sloan Management Review, 39(2), 43-57.

1. Davenport, T., DeLong, D. and Beers, M. 1998. Successful knowledge
management projects. Sloan Management Review, 39 (2). 43-57.

[1] T. Davenport, D. DeLong, and M. Beers, "Successful knowledge management
projects," Sloan Management Review, vol. 39, no. 2, pp. 43-57, 1998.
```

Abbildung 2.1: Vier unterscheidliche Literaturreferenzen auf dieselbe Publikation

Der wichtigste Grund für unterschiedliche Literaturangaben zu ein und derselben Publikation ist liegt in der Vielzahl gültiger Zitationsstile, mit Hilfe derer Autoren die Quellenangaben erstellen. Im angloamerikanischen Sprachraum sind die Zitationsstile von American Psychological Association (APA) [1], Modern Language Association (MLA)[12] und

American Medical Association (AMA)[14] sowie Chicago Style[29], Harvard Style[28] und Turabian Style[34] weit verbreitet. Im deutssprachigen Raum werden Zitationsregeln u.a. durch die DIN 1505-2 [35] vorgegeben. Auf internationaler Ebene gibt es die ISO 690 [6]. Das National Information Standards Organization (NISO) hat ebenfalls Richtlinien zum Referenzieren für eine Vielzahl von Materialien erstellt [30]. Betrachtet man sich die handelsüblichen Computerprogramme zur automatischen Erstellung von Literaturverzeichnissen, so erhält man ein ähnliches Bild. Die Software Endnote bietet nach Angaben des Herstellers ISI ResearchSoft[1] in der Version X (Dezember 2006) weit mehr als 2.300 unterschiedliche Templates für Literaturreferenzen zu insgesamt 42 verschiedenen Referenztypen an.

- *Buch*: Monographien ; mehrbändige Werke mit eigenen Stücktiteln ; Forschungsberichte
- *Artikel*: Aufsätze in Zeitschriften, Zeitungen
- *Kapitel*: Kapitel in Monographien ; Beiträge in Handbüchern, Sammelwerken ; Vorträge auf Tagungen u.ä.
- *Paper*: Schriften von Kongressen, Tagungen , Symposien und ähnlichen Zusammenkünften
- *Abschlussarbeit*: Dissertationen ; Diplomarbeiten
- *Technischer Report*: Hochschulschriften ; unveröffentliche wissenschaftliche Arbeiten

Abbildung 2.2: Die sechs unterschiedlichen Publikationstypen (nach K.F. Lorenzen [18])

Neben der Vielzahl an Zitationsstilen existiert ebenfalls eine weite Bandbreite an zitierfähigen Publikationsformen, von denen nahezu jede ihr eigenes Zitationsschema besitzt. Mit der für LaTEX entwickelten BibTEX-Erweiterung zur automatischen Erstellung eines Quellenverzeichnis' lassen sich allein 14 unterschiedliche Publikationsformen referenzieren, die sich jedoch in einigen Punkten überchneiden. Nach DIN 1505-2 unterscheidet K.F. Lorenzen [18] sechs relevante zitierfähige Publikationstypen (vgl. Abbildung

[1] http://www.endnote.com/enXinfo.asp

2.2). Für jeden dieser Publikationstypen existieren obligatorische Metadateb, die, wenn sie dem Autor vorliegen, bei der Referenzierung in einem Literaturverzeichnis angegeben werden müssen [30]. Bei den Metadaten handelt es sich um die bibliographische Angaben, wie Autor, Titel, Jahr, anhand derer eine Publikation eindeutig referenziert werden kann. Sie bilden die Grundvoraussetzung für die Konstruktion eines Zittaionsgraphen. Einen Überblick über die obligatorischen Metadaten der sechs oben genannten Publikationstypen ist in Abbildung 2.3 zu sehen.

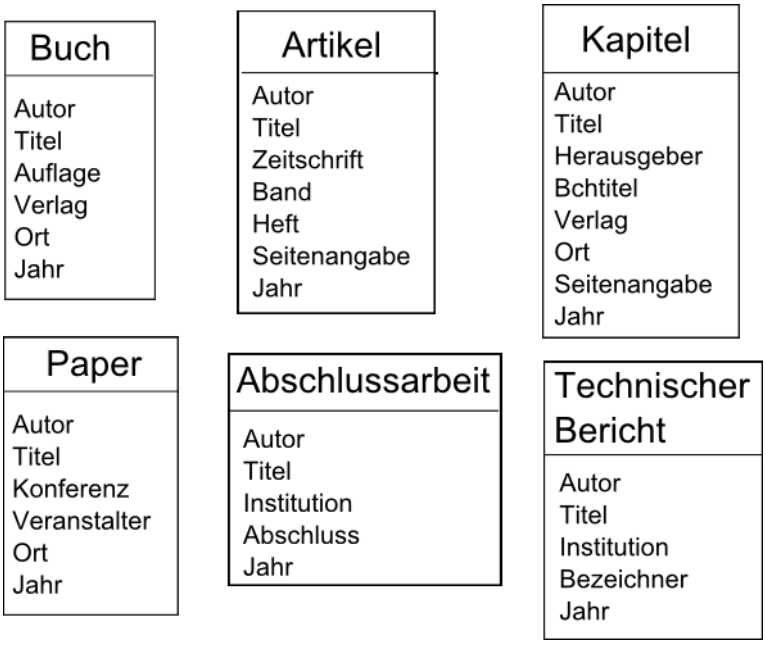

Abbildung 2.3: Die sechs am häufigsten zitierten Publikationsformen und deren Metadaten bei einer Quellenangabe.

Bei einem Zitationsgraphen handelt es sich um eine Verweisstruktur, die auf der Basis von Publikationen und deren Metadaten Zitationen zwischen Publikationen abbildet. Die nachfolgende Definition formalisiert den Begriff des Zitationsgraphen.

Definition 1 Ein Zitationsgraph $G(P, Z)$ besteht aus einer Menge von m Publikationen, $P = p_1, \ldots, p_m$, den Knoten des Graphen, und aus einer Menge von n Zitationen $Z = z_1, \ldots, z_n$, den gerichteten Kanten im Graph, wobei für jede Zitation $z_i = (p_i, p_{i'})$ gilt: $p_i, p_{i'} \in P$.

Ein korrekter Zitationsgraph erfüllt dabei zwei Bedingungen. Eine Publikation zitiert sich

niemals selbst:

$$\forall z_i = (p_i, p_{i'}) : p_i \neq p_{i'}, \tag{2.1}$$

sowie eine eine Publikation zitiert eine andere Publikation maximal einmal:

$$\forall z_i = (p_i, p_{i'}) : \neg\exists z_j = (p_j, p_{j'}) : p_i = p_j \wedge p_{i'} = p_{j'}. \tag{2.2}$$

Gegenseitige Zitationen der Publikationen untereinander sowie Zyklen innerhalb des Graphen sind dagegen legitim.

Ein Knoten im Zitationsgraphen repräsentiert dabei eine Menge von Datenpunkten, genau diejenigen, die dieselbe Publikation beschreiben. Publikationen können daher auch als Äquivalenzklassen von Datenpunkten interpretiert werden. Zwei Datenpunkte gelten dabei als äquivalent, wenn sie dieselbe Publikation beschreiben. Dann sind sie Elemente derselben Äquivalenzklasse. Ein Zitationsgraph kann daher auch als Äquivalenzrelation auf der Menge D, der Menge aller Datenpunkte, angesehen werden.

Die Generierung eines Zitationsgraphen ist eine Instanz der Entity-Resolution. Im Gegensatz zur klassischen Entity-Resolution, bei der die paarweise Äquivalenz von Datenpunkten betrachtet wird, geht es bei der Generierung eines Zitationsgraphen um die n-fache Äquivalenz von Datenpunkten. Das Problem der Entity-Resolution sei folgendermassen definiert:

Definition 1 Gegeben sei eine Menge K von Äquivalenzklassen, $K = m_1, \ldots, m_n$, und eine Menge D von Datenpunkten, $D = d_1, \ldots, d_n$. Eine Äquivalenzrelation $\omega \in \Omega$, wobei Ω die Menge aller Äquivalenzrealtionen über D ist, ordnet jedem Datenpunkt $d_i, i = 1, \ldots, n$ eine Äquivalenzklasse aus K zu. Das Problem der Entity-Resolution besteht darin, die optimale Äquivalenzrelation $\omega' \in \Omega$ zu finden.

Die Menge möglicher Äquivalenzrelationen Ω verhält sich dabei exponentiell zur Anzahl der Datenpunkte. Dadurch ist es praktisch nicht möglich für eine grosse Menge von Datenpunkten alle in Frage kommenden Äquivalenzrelationen zu betrachten. Ein vielversprechender Ansatz ist, nur solche Äquivalenzrelationen zu betrachten, die mit hoher Wahrscheinlichkeit die optimale Äquivalenzrelation sind. Dieser Aspekt findet sich in der Auswahl des Verfahrens wieder, mit dem die optimale Äquivalenzrelation gesucht wird.

Ein wichtiger Bestandteil des Suchverfahrens ist es, Äquivalenzrelationen zu bewerten. Zur Bestimmung der Qualität einer Äquivalenzrelation ω werden Genauigkeit (Precision) und

Vollständigkeit (Recall), die klassischen Bewertungskriterien des Information Retrievals, herangezogen. Diese beruhen auf dem paarweisen Vergleich zweier Datenpunkte. Sei ω' die optimale Äquivalenzrelationen. Dann ergeben sich die folgenden vier Indikatoren:

- TP = Anzahl der Datenpunktpaare, die sich sowohl unter ω als auch unter ω' in derselben Äquivalenzklasse befinden
- TN = Anzahl der Datenpunktpaare, die sich sowohl unter ω als auch unter ω' in verschiedenen Äquivalenzklasse befinden
- FP = Anzahl der Datenpunktpaare, die sich unter ω in derselben Äquivalenzklasse aber unter ω' in verschiedenen Äquivalenzklassen befinden
- FN = Anzahl der Datenpunktpaare, die sich unter ω' in derselben Äquivalenzklasse aber unter ω in verschiedenen Äquivalenzklassen befinden

Daraus ergeben sich die folgenden Gleichungen:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

sowie

$$Recall = \frac{TP}{TP + FN}. \tag{2.4}$$

Um sowohl die Anzahl der falsch positiv als auch die der falsch negativ gruppierten Datenpunktpaare in einer Gleichung zu berücksichtigen, wird des weiteren der Jacard-Index [2] herangezogen:

$$J(\omega, \omega') = \frac{TP}{TP + FN + FN}. \tag{2.5}$$

Obwohl es sich um standardisierte Bewertungskriterien handelt, sind sie nicht frei von Kritik. Kritikpunkt an obigen Maßen ist, daß diese lediglich den Grad der Übereinstimmungen einer Äquivalenzrelation ω mit der optimalen Äquivalenzrelation ω' bestimmen, nicht jedoch wie die auftretenden Differenzen entstanden sein könnten. Ein alternativer Ansatz basiert auf der Kardinalitä der einzelnen Äquivalenzklassen und wird von Larsen et al. [16] und Heckerman et al. [22] aufgegriffen. Meila [23] verfolgt die Idee, zu jeder Äquivalenzklasse unter ω die ähnlichste Äquivalenzklasse unter ω' zu finden.

3 Verwandte Arbeiten

An dieser Stelle werden wissenschaftliche Arbeiten vorgestellt, die sich allgemein mit dem Thema *Entity Resolution* bzw. im speziellen mit dem Matchen von Zitationen befassen. Bei *Entity Resolution* handelt es sich um eine Problemstellung, die in vielen Gebieten der Informationsintegration aufgeworfen wird. Dabei geht es darum, verschiedene Instanzen unterschiedlicher Quellen die dasselbe Objekt der realen Welt, eine sogenannte *Entität* beschreiben, zu gruppieren. Synonym zum Begriff Entity Resolution werden unter anderem die Begriffe *Record Linkage*, *Deduplication*, *Objectidentification* und *Identity Uncertainty* gebraucht.

Der Begriff Rcord Linkage wurde zum ersten Mal von H.L. Dunn [4] im Rahmen einer Untersuchung von Patientenakten eingeführt, mit dem Ziel Informationen über ein und dieselbe Person aus unterschiedlichen Quellen zusammenzuführen um einen Informationsgewinn zu erzielen. Später wurde diese Idee von Newcombe et al. [26] sowie Fellegi et al. [5] aufgegriffen und formalisiert.
Das Prinzip ist simpel. Aus beiden Quellen werden Records paarweise bezüglich einer bestimmten Menge von Eigenschaften verglichen. Zunächst werden zwei Wahrscheinlichkeiten gebildet, eine dafür daß beide Records dieselbe Entität beschreiben und einer dafür, daß die unterschiedliche Entitäten ebschreiben. Ist der Quotient oberhalb eines oberen Schwellwerts, geht man davon aus, daß beide dieselbe Entität beschreiben, liegt unterhalb einer unteren Grenze, kann man sicher sein, daß es unterschiedliche Entitäten sind. Liegt er dazwischen, so herrscht eine gewisse Unsicherheit. Diese kann beispielsweise durch manuelle Überprüfungd er Records aufgelöst werden

Entity Resolution für Zitationen wird in der Literatur auch als *Citation Matching* bezeichnet. Lawrence et al. [17] beschreiben einen mehrstufigen Algorithmus, mit dem Ziel Literaturreferenzen, die auf denselben Artikel verweisen, zu gruppieren. Der erste Schritt besteht in der Normalisierung der Zitationen: Kleinschreibung, Entfernung von Tags und

einiger Sonderzeichen sowie der Ausschreibung häufig vorkommender Abküerzungen sowie dem Entfernen von belanglosen Wörtern. Das Matchen der Zitationen wurde einerseits mit unterscheidlichen Wort- und Phrase-Matching-Algorithmen als auch auf der Ähnlichkeit wichtiger Subfields wie Autor und Titel durchgeführt.

Ein zweistufiges Verfahren zum Clustering bibliographischer Daten wird von McCallu et al. [21] vorgestellt. Das Hauptaugenmerk dieser Arbeit befasst sich mit der Problemstellung wie auf der Basis groser Datenmengen ein effizientes Clustering von Zitationen realisiert werden kann. Zunächst werden durch ein grobes Clustering Untermengen von Referenzen, sogenannte *Canopies*, erzeugt. Die Besonderheit an dieser Herangehensweise ist die, daß sich die Teilmengen überschneiden können, eine Referenz kann also mehreren Canopies zugeordnet. Im Anschluß daran werden klassische Clusteringverfahren (hier: *Greedy Agglomerative Clustering* bzw. *k-Means*) angewendet.

Sarawagi et al. [33] untersuchen ein Verfahren, welches auf einem Teil des arXiv-Datensatz [1] einen Zitationsgraphen generiert. Die ca. 35.000 Artikel des Korpus liegen im Latex-Format vor. Dies hat den Vorteil, daß wichtige Metadaten der Artikel ebenso wie der Zitationen durch LaTeX-spezifische Tags markiert sind, wodurch der nachfolgende Schritt der Informationsextraktion stark vereinfacht wird. Der zweite Schritt besteht darin, die extrahierten Metadaten von unrelevanten Bestandteilen zu bereinigen. Anschliessend wurde für jede im Korpus enthaltene Zitation eine Menge von 15 Kandidaten-Artikel generiert, welche besonders relevante Woerter der Citation wie Nachnamen der Autoren, signifikante Titelwörter enthaelt. Auf dieser Liste werden anschliessend verschiedene Filterverfahren angewendet. Die dadurch erhaltene Restmenge von Artikeln wird abschliessend mittels TF-IDF auf Ähnlichkeit mit der Zitation überprüft. Falls der auf diese Weise bestbewerteste Artikel signifikant besser als alle anderen Artikel abschneidet, wird angenommen, dass dieser durch die Zitation referenziert wird.

Ein wahrscheinlickheitrstheoretischer Ansatz wird von Pasula et al. [31] aufeggriffen. Es basiert auf einem Probabilistischen Relationalen Modell. Dieser Ansatz ist sehr vielversprechend, da er sich nicht auf die Ähnlichkeit der Metadaten beschränkt, sondern diese als eigenständige Entitäten betrachtet, die in Relationen zueinander stehen können. Zwei Zitationen referenzieren dieselbe Publikation, wenn sich die Objekte in ihren Eigenschaften ähneln. Als Inferenzverfahren wird ein MCMC-Algorithmus verwendet, welcher auf

[1] http://arxiv.org/

Basis geschätzter Wahrscheinlichkeiten gleiche Referenzen erkennt. Schwachstelle dieses Ansatzes ist die Vielzahl an Entitäten die sich dadurch ergibt, wenn viele Zitationen aufgelöst werden.

Der Science Citation Index (SCI) ist eine Zitationsdatenbank, die vom Thomson Institute for Scientific Information betrieben wird. Er basiert auf der Idee von E. Garfield [], Zitationen zur bibliometrischen Analyse zu verwenden. Seit 1961 werden jährlich Zitationen aus anfangs 562, heute über 3.700 Fachzeitschriften (Stand: Mai 2007) gesammelt und ausgewertet. Der SCI Expanded ist eine noch umfangreichere Datenbank die seit 1945 Zitationen aus 5.900 Fachzeitschriften (Stand: Mai 2007) auswertet. Er ist Teil des Web Of Science [2] und bildet die Grundlage für den jährlich veröffentlichten Impact Factor im Rahmen des Journal Citation Reports.

CiteSeer [3] ist eine Suchmaschine für frei zugängliche, wissenschaftliche Publikationen. Sie wurde am NEC Research Institute entwickelt und wird heute von der Pennsylvania State University betrieben. CiteSeer stellt die praktische Umsetzung des von Lawrence et al. [17] beschriebenen Verfahren dar. Die Datenbank umfasst nach eigenen Angaben mehr als 720.000 Artikel überwiegend aus dem Bereich der Informatik und Informationstechnologie. Die Literaturreferenzen wurden aufgelöst, so dass zu jeder Publikation die Anzahl der Zitationen und deren Herkunf angezeigt wird. Diese Informationen fliessen in das Ranking ein, so daß die Ergebnisse einer Suchanfrage nach deren Zitationshäufigkeit sortiert sind.

SMEALSearch [4] wird ebenfalls von der Pennsylvania State University betrieben. Es basiert auf dem gleichen Verfahren wie CiteSeer. Das Hauptaugenmerk richtet sich auf wissenschaftliche Publikationen aus dem Bereich Wirtschaft. Darüberhinaus haben die Nutzer die Möglichkeit, Artikel eigenständig in die Datenbank einzufügen. Auch bei SMEALSearch werden Zitationen automatisch analysiert und spiegeln sich im Ranking wieder.

Seit 2004 bietet Google mit GoogleScholar [5] eine Artikeldatenbank im Internet an. Diese entstand aus der Zusammenarbeit mit mehreren Fachverlagen. Neben frei verfügbaren Artikeln sind auch kostenpflichtige Artikel in der Datenbank enthalten. Auch GoogleScholar sortiert die Ergebnisliste einer Anfrage nach der Häufigkeit von Zitationen auf

[2] http://scientific.thomson.com/products/wos/
[3] http://citeseer.ist.psu.edu/
[4] http://smealsearch2.psu.edu/index.html/
[5] http://scholar.google.de/

eine Publikation.

Im Bereich des Bibliothekswesens bildet Deduplication ein zentrales Problem. Bei der föderierten Anfrage an mehrere OPACs werden die Ergebnissmengen durch Match und Mergeverfahren nachbearbeitet mit dem Ziel gleiche Treffer aus unterschiedlichen Quellen zu erkennen, ein Mehr an Informationen zu generieren und den Bestand ein und desselben Buches in allen OPACs anzuzeigen. Ein Beispiel hierfür bildet die Suchmaschine des Kooperativen Biblitheksverbundes Berlin Brandenburg (KOBV) [6].

[6]http://digibib.kobv.de

4 Theoretische Grundlagen

4.1 Das Probabilistische Relationale Modell

Probabilistische Relationale Modelle (PRMs) wurde von Friedman et al. [7] eingeführt, die sich dabei auf die grundlegenden Arbeiten von Poole [32], Ngo et al. [27] sowie Koller et al. [15] berufen. Es kann als eine Erweiterung des Relationalen Modells (RM) angesehen werden. Im Gegensatz zum RM, welches Instanzen, deren Eigenschaften und die Relationen der Instanzen untereinander als absolut betrachtet, können mit dem PRM auch unsichere Informationen, wie bei der Entity-Resolution, modelliert werden. Dazu ist es notwendig, eine geeignete Abhängigkeitsstruktur zu lernen und die Parameter dieser Struktur zu schätzen.

4.1.1 Das Relationale Modell

Das RM stammt aus dem Bereich der Datenbanktheorie und wurde von Codd [3] entwickelt. Es dient dazu, Entitäten und deren Relationen zueinander zu modellieren. Grundlage des RM bilden Tabellen, in denen Entitäten als Tupel und deren Eigenschaften als Attribute abgespeichert werden. Weitere Tabellen dienen dazu, Relationen zwischen den Entitäten abzuspeichern.

Die nachfolgende Definition des RMs ist angelehnt an Friedman et al. [7]:

Definition 1 Das Schema eines RMs besteht aus folgenden Komponenten:

- einer Menge verschiedener Klassen, $\mathcal{X} = \{X_1, \ldots, X_n\}$,
- zu jeder Klasse $X_i \in \mathcal{X}$ eine Menge von Attributen $\mathcal{A}(X_i)$, wobei das Attribut A der Klasse X_i mit $X_i.A$ und dessen Wertebereich mit $WB(X_i.A)$ bezeichnet wird sowie

- eine Menge verschiedener Relationen, $\mathcal{R} = \{R_1, \ldots, R_m\}$, die Beziehungen zwischen den Entitäten der verschiedenen Klassen bechreiben.

Eine Instanz $\mathcal{I}$ definiert für jede Klasse X_i die Menge der Entitäten sowie für jede Entität $x \in \mathcal{I}(X_i)$ und jedes Attribut $X_i.A \in \mathcal{A}(X_i)$ eine Belegung $x.A$. Dieselbe Instanz $\mathcal{I}$ bestimmt für jede Relation $R_j(X_{j_1}, \ldots, X_{j_k}) \in \mathcal{R}$ die Entitäten $\langle x_1, \ldots, x_k \rangle \in \mathcal{I}(X_{j_1}) \times \cdots \times \mathcal{I}(X_{j_k})$, die R_j erfüllen.

Sei $R(X_1, \ldots, X_k)$ eine k-stellige Relation, mit $k \geq 2$, so kann diese auf ihr $i - tes$ und $j - tes$ Argument projeziert werden. Daraus ergibt sich die binäre Relation $\rho(X_i, X_j)$, die als Schlüssel bezeichnet wird. Für eine beliebige Entität $x \in \mathcal{I}(X_i)$ definiert $x.\rho$ diejenigen Entitäten $x' \in \mathcal{I}X_j)$ für die $\rho(X_i, X_j)$ erfüllt ist. Unter der Schlüsselkette $\tau = \rho_1 \cdot \ldots \cdot \rho_m$ versteht man die Konkatenation mehrerer zu einander passender Schlüssel, wobei für alle $l = 1, \ldots, m-1$ gilt: $\rho_l(X_i, X_j)$ und $\rho_{l+1}(X_j, X_k)$.

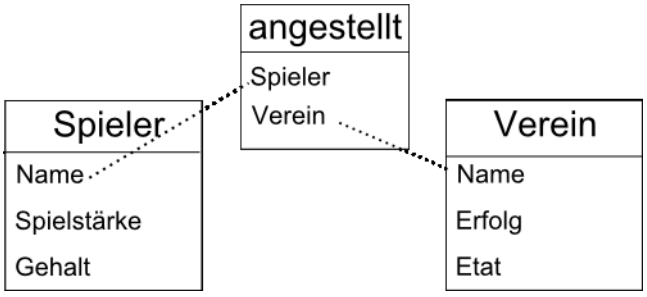

Abbildung 4.1: Schema eines RMs mit zwei Klassen und einer Relationen. Die

In Abbildung 4.1 ist das Schema eines RMs dargestellt. Es besteht aus der Klasse *Spieler*, mit den Attributen *Name*, *Spielstärke* und *Gehalt*, der Klasse *Verein*, mit den Attributen *Name*, *Erfolg* und *Etat*, und der Relation *angestellt* besteht. Aus der Relation *angestellt* lassen sich die Schlüssel $\rho_1(Spieler, Verein)$ und $\rho_2(Verein, Spieler)$ ableiten. Neben den trivialen Schlüsselketten $\tau_1 = \rho_1$ und $\tau_2 = \rho_2$ lässt sich die Schlüsselkette $\tau_3 = \rho_1 \cdot \rho_2$ konstruieren, die zur Ermittlung aller Mitspieler eines Spielers dienen kann.

Abbildung 4.2 zeigt eine Instanz $\mathcal{I}$ des Schemas aus Abbildung 4.1. Die Instanz $\mathcal{I}$ definiert dabei sowohl alle Entitäten der Klasse *Spieler*, wobei $\mathcal{I}(Spieler) = \{S_1, S_2, S_3\}$ sowie deren Attribute, als auch die Entiäten der Klasse $Verein$, wobei $\mathcal{I}(Verein) = \{V_1, V_2\}$, und deren Attribute. Zusätzlich bestimmt die Instanz $\mathcal{I}$ für die Relation *angestellt* die Tupel $angestellt(S_1, V_1)$, $angestellt(S_2, V_1)$ und $angestellt(S_3, V_2)$. Der aus der Relati-

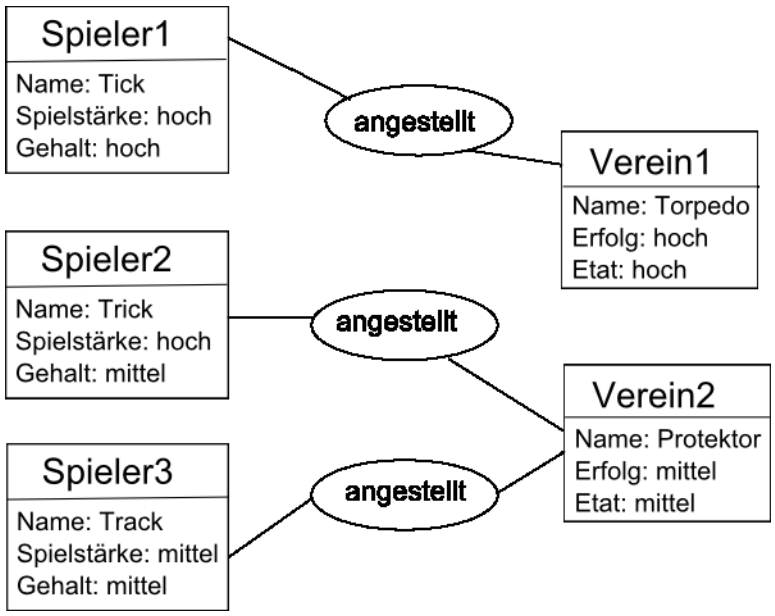

Abbildung 4.2: Instanz eines RM-Schemas mit zwei Klassen und einer Relation.

on *angestellt* resultierende Schlüssel $\rho_1(Spieler, Verein)$, der jedem Spieler einen Verein zuordnet, hat die Belegungen $S_1.\rho_1 = V_1$, $S_2.\rho_1 = V_2$ und $S_3.\rho_1 = V_2$. Der Schlüssel $\rho_2(Verein, Spieler)$, der jedem Team eine Menge von Spielern zuordnet, hat die Belegungen $V_1.\rho_2 = \{S_1, S_2\}$ und $V_2.\rho_2 = \{S_3\}$.

4.1.2 Modellbeschreibung

Beim PRM wird die Menge der Attribute in zwei disjunkte Teilmengen unterteilt. *Feste* Attribute sind im Gegensatz zu den *probabilistischen* Attributen für alle Entitäten determiniert. Auf dieser Unterscheidung der Attribute basiert der Begriff der Skelettstruktur σ, eine partielle Spezifizierung der Instanz $\mathcal{I}$, die für alle Entitäten die Belegung der festen Attribute bestimmt und die probabilistischen Attribute unspezifiziert lässt. Die Skelettstruktur σ definiert ebenfalls die Relationen zwischen den einzelnen Entitäten.

Das PRM beschreibt die Wahrscheinlichkeitsverteilung für potentielle Instanzen $\mathcal{I}$, zu einer gegebenen Skelettstruktur σ und ordnet dabei jeder möglichen Instanz I eine Wahrscheinlichkeit $P(\mathcal{I} \mid \sigma)$ zu. Die Menge aller Verteilungen dieser Wahrscheinlichkeiten wird durch das PRM beschrieben. Das PRM weist dabei zwei Bestandteile auf: eine Abhängig-

keitsstruktur S sowie deren Parameter θ_S. Die Abhängigkeitsstruktur S ordnet jedem probabilistischen Attribut $X.A$ eine *Elternmenge* von festen Attributen $E(X.A)$ zu. Intuitiv handelt es sich dabei um diejenigen Attribute, die $X.A$ direkt beeinflussen. Die Parameter θ_S beschreiben eine Wahrscheinlichkeitsverteilung.

Es werden zwei Arten von Eltern unterschieden. Das probabilistische Attribut $X.A$ kann einerseits von einem festen Attribut $X.B$ derselben Klasse beeinflusst werden. Andererseits kann $X.A$ auch durch ein festes Attribut $Y.\tau.B$ einer anderen Klasse Y, die durch die Schlüsselkette τ mit X verbunden ist, beeinflusst werden. Für den Fall, dass es sich bei $Y.\tau.B$ um eine Menge handelt, wird für die Darstellung der Abhängigkeit eine Aggregatfunktion verwendet und die Abhängigkeit wird bezüglich eines aggregierten Wertes berechnet. Beispiele für Aggregatfunktionen sind Minimum, Maximum, Arithmetisches Mittel, häufigster auftretender Wert oder Kardinalität der Menge.

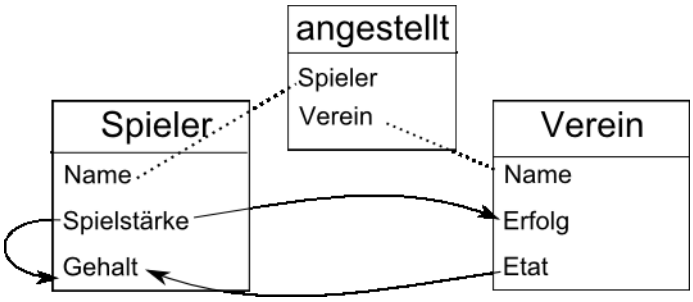

Abbildung 4.3: Abhängigkeiten innerhalb eines RM-Schemas

In Abbildung 4.3 sind die Abhängikeiten für das begleitende Beispiel aufgezeigt. Die Klasse *Spieler* besitzt die beiden festen Attribute *Name, Spielstärke* und das probabilistisches Attribut *Gehalt* mit der Elternmenge $\{Spieler.Spielstarke, Verein.Etat\}$. Diese Abhängigkeit deutet daraufhin, daß das Gehalt eines Spielers sowohl von dessen Spielstärke als auch vom Etat seines Vereins abhängig ist. Für die Klasse Verein existieren die festen Attribut *Name, Etat* und das probabilistische Attribut *Erfolg* mit der Elternmenge $\{Spieler.Spielstarke\}$. Diese Abhängigkeit besagt, daß der Erfolg eines Vereins von den Spielstärken seiner angestellten Spieler abhängig ist.

Für eine Entität x ist die Belegung eines Attributs $X.A$ abhängig von der Belegung der Eltermenge dieses Attributs, $E(X.A)$, und wird durch die bedingte Wahrscheinlichkeit $P(X.A \mid E(X.A))$ spezifiziert. Diese besagt, wie wahrscheinlich die Ausprägung eines Wertes des Attributs $X.A$ für eine gegebenen Belegung seiner Elternmenge $E(X.A)$ ist.

Diese wird durch die bedingte Wahrscheinlichkeitsverteilung $\theta_{X.A|E(X.A)}$ beschrieben. Die Wahrscheinlichkietsverteilung $\theta_{X.A|E(X.A)}$ definiert also für ein Attribut $X.A$ bei gegebener Belegung der Elternmenge $E(X.A)$ die Wahrscheinlichkeitsverteilung über dem gesamten Wertebereich des Attributs $WB(X.A)$. Bei θ_s handelt es sich also um die Menge aller bedingten die Wahrscheinlichkeiten für alle Klassen $X_1, \ldots, X_k$, deren Attribute $X.A$ und den dazugehörigen Elternmengen $E(X.A)$.

Anhand der Abhängigkeitstrukutur S und deren Parameter θ_S kann damit für eine Instanz I zu einer gegebenen Skelettstruktur σ die Wahrscheinlichkeit berechnet werden. Dabei wird zunächst die Menge aller Klassen und deren Attributmengen ermittelt. Für alle Klassen X und für alle durch die Skelettstruktur σ spezifizierten Instanzen x werden die Wahrschenlichkeiten für die Belegungen der probabilistischen Attribute $X.A$ aufmultipliziert:

$$P(\mathcal{I} \mid \sigma, S, \theta_S) = \prod_{X \in \mathcal{X}} \prod_{A \in \mathcal{A}(X)} \prod_{x \in \mathcal{I}\sigma(X)} P(x.A \mid E(X.A)). \tag{4.1}$$

Zentraler Bestandteil der Gleichung 4.1 ist die Wahrscheinlichkeitsverteilung $P(x.A \mid E(X.A))$. Wie diese bestimmt werden kann, wird im Abschnitt 4.1.3 erläutert. Abschnitt 4.1.4 setzt sich mit der Frage auseinander, wie Abhängigkeiten zwischen Attributen ermittelt werden können.

4.1.3 Bestimmung der Modellparameter

Die Bestimmung der Modellparameter bei PRM hat starke Ähnlichkeit zu der von Bayes'schen Netzen [13]. Die Aufgabe besteht darin, zu einer gegeben Abhängigkeitsstruktur S und einer Instanz $\mathcal{I}$ die Strukturparameter θ_S zu schätzen. Bei θ_S handelt es sich um die Menge aller bedingten Wahrscheinlichkeiten $\theta_{X.A|E(X.A)}$. Diese beschreiben eine Wahrscheinlichkeitsverteilung der Belegung eines probabilistischen Attributs $X.A$ zu gegebener Belegung der Elternmenge $E(X.A)$.

Ein gängiges Verfahren zur Bestimmung der Modellparameter ist die *Maximum-Likelihood-Parameterschätzung*, kurz *ML-Schätzung*, die auf einer Trainingsmenge durchgeführt wird. Zentraler Aspekt der ML-Schätzung für PRM ist die Likelihood $L(\theta_S \mid \mathcal{I}, \sigma, \mathcal{S})$. Sie beschreibt wie gut die geschätzten Parameter θ_S zu einem gegebenen Modell, bestehend aus einer Abhängigkeitsstruktur S, einer Skelettstruktur σ und einer Instanz

$\mathcal{I}$ passen:

$$L(\theta_S \mid \mathcal{I}, \sigma, \mathcal{S}) = P(\mathcal{I} \mid \sigma, \mathcal{S}, \theta_S). \tag{4.2}$$

Das Ziel der ML-Schätzung ist es, diejenigen Parameter θ_S zu bestimmen, die die Likelihood $L(\theta_S \mid \mathcal{I}, \sigma, \mathcal{S})$ maximieren. Um den für die Schätzung benötigten Rechenaufwand zu reduzieren, wird häufig die log-Likelihood l berechnet:

$$l(\theta_S \mid \mathcal{I}, \sigma, \mathcal{S}) = \log P(\mathcal{I} \mid \sigma, \mathcal{S}, \theta_S). \tag{4.3}$$

Für die bedingten Wahrscheinlichkeiten des PRM kann die ML-Schätzung durch Abzählen realisiert werden. Dabei stellt $C_{X.A}[v, u]$ die Häufigkeit für das gemeinsame Auftreten einer Belegung des Attributs $X.A = v$ und dessen Elternmenge $E(X.A) = u$ innerhalb der Trainingsmenge. Anhand dieser Häufigkeiten werden die bedingten Wahrscheinlichkeiten wie folgt bestimmt:

$$P(X.A = v \mid E(X.A) = u) = \frac{C_{X.A}[v, u]}{\sum_{v'} C_{X.A}[v', u]}. \tag{4.4}$$

Kritikpunkt am ML-Schätzer ist der zu starke Einfluß der Trainingsdaten auf die bedingten Wahrscheinlichkeiten. Eine Möglichkeit dies einzuschränken ist es, eine Glättung der Parameter durchzuführen. Bei einer Glättung wird ein sogenannter *Prior* α zu den in der Trainingsmenge aufgetretenen Häufigkeiten hinzuaddiert. Daraus ergibt sich für die bedingten Wahrscheinlichkeiten die folgende Gleichung:

$$P(X.A = v \mid E(X.A) = u) = \frac{C_{X.A}[v, u] + \alpha}{\sum_{v'} C_{X.A}[v', u] + \alpha}. \tag{4.5}$$

4.1.4 Lernen der Abhängigkeitsstruktur

Ist ein ausreichen großes Expertenwissen über das Schema eines PRM vorhanden, können die Abhängigkeiten zwischen den Attributen von Hand erstellt werden. Anders sieht es aus, wenn dieses Wissen nicht vorhanden ist. Dann ist es notwendig, die Abhängigkeiten auf einer Trainingsmenge zu lernen. Für PRMs ist das Lernen einer Abhängigkeitsstruktur ähnlich zu dem von Bayes'schen Netzen [13]: innerhalb eines definierten Suchraumes soll mittels einer Bewertungsfunktion und eines Suchalgorithmus die höchst bewertete Strukur gefunden werden.

Für das Lernen einer Abhänigkeitsstruktur sind die folgenden drei Teilprobleme zu lösen:

- Einschränkung des Suchraums
- Bewertung einer gegebenen Struktur
- Suche nach der am höchsten bewerten Struktur

Diese Teilprobleme können unabhängig voneinander betrachtet werden.

Einschränkung des Suchraums
Nicht alle Abhängigkeitsstrukturen sind für ein PRM geeignet. Nur solche Strukturen, aus denen ein konsistentes Wahrscheinlichkeitsmodell folgt, sollen dabei berücksichtigt werden. Dies bedeutet, daß kein Attribut innerhalb dieses Modells direkt oder indirekt von sich selbst abhängig ist. Um den Suchraum auf konsistente Strukturen einzuschränken, kann man sich der äquivalenten Darstellung einer Abhängigkeitsstruktur durch einen Graphen bedienen (vgl. [7, 11]). In diesem Graphen werden Attribute durch Knoten dargestellt und Abhängigkeiten durch gerichetete Kanten zwischen den Knoten. Handelt es sich dabei um einen zyklenfreien Graph, so ist gewährleistet, daß kein Attribut in dieser Struktur direkt oder indirekt von sich selbst abhängig ist. Die Azyklität eines Graphen lässt sich algorithmisch feststellen (vgl. [7, 11]).

Bewertung von Strukturen
Der auf dem Suchraum operierende Suchalgorithmus benötigt eine Funktion, mit der Strukturen evaluiert werden können. Auch hier ist eine Anlehnung an Bayes'sche Netze sinnvoll. Dabei wird die bedingte Wahrscheinlichkeit einer Abhängigkeitsstruktur S zu einer gegebenen Instanz $\mathcal{I}$ und einer definierten Skelettstruktur σ ermittelt. Durch Anwendung des Satz' von Bayes ergibt sich die folgende Gleichung:

$$P(S \mid \mathcal{I}, \sigma) = P(S \mid \sigma)P(\mathcal{I} \mid S, \sigma). \tag{4.6}$$

Unter der Annahme, daß die Wahrscheinlichkeit einer Struktur S unabhängig von der Skelettstruktur σ ist, lässt sich Gleichung 4.1.4 noch vereinfachen:

$$P(S \mid \mathcal{I}, \sigma) = P(S)P(\mathcal{I} \mid S, \sigma). \tag{4.7}$$

Wie in Gleichung 4.7 leicht zu erkennen ist, berechnet sich die Wahrscheinlichkeit einer Struktur S, gegeben eine Instanz $\mathcal{I}$ und eine Skelettstruktur σ aus zwei Faktoren: deren A-priori-Wahrscheinlichkeit von S und der bedingten Wahrscheinlichkeit von I, gegeben

S und σ. Vorschläge zur Bestimmung beider Wahrscheinlichkeiten finden sich u.a. bei Friedman et al. [7]

Suche nach Strukturen
Die Lösung für das dritte Teilproblem bedingt der Lösung der beiden vorherigen. Ziel ist es, einen Suchalgorithmus zu entwerfen, der in einem definierten Suchraum hoch bewertete Strukturen findet. Da das Suchproblem für Bayes'sche Netze NP-hart ist (vgl. [7]) und PRMs komplexer als Bays'sche Netze sind, ist ein Algorithmus, welcher die optimale Struktur findet nicht effektiv anwendbar. Deshalb muss bei der Suche nach Strukturen auf Heuristiken zurückgegriffen werden, wie dem Greedy-Hillclimbing-Algorithmus (vgl. [7]). Ein solcher Algorithmus wird in Kapitel 5 vorgestellt.

4.2 MCMC-Algorithmen

Mit dem in Abschnitt 4.1 eingeführten PRM lassen sich unsichere Informationen über Entitäten modellieren. Es bedarf daher eines Verfahrens, daß diese Unsicherheiten auflöst. Bei MCMC-Algorithmen handelt es sich um Verfahren, die auf unsicheren Zustandsräumen operieren und Inferenzen ableiten können. Sie lassen sich auch auf grossen Zustandsräumen, wie dem der Entity-Resolution, anwenden. Dies ist eine weitere Motivation, diese genauer zu untersuchen. In diesem Kapitel werden zunächst Markov-Ketten eingeführt, da diese die Grundlage für MCMC-Algorithmen bilden. Anschließend werden MCMC-Algorithmen im Allgemeinen sowie Gibbs-Sampling und der Metropolis-Hastings-Algorithmus im Speziellen vorgestellt.

4.2.1 Markov-Ketten

Markov-Ketten sind eine spezielle Klasse stochastischer Prozesse. Sie gehen zurück auf Andrej Andrejewitsch Markov, einen russischen Mathematiker, der diese Anfang des 20.Jahrhunderts erstmalig beschreibt [19]. Markov-Ketten ermöglichen es, Wahrscheinlichkeiten für das Eintreten zukünftiger Ereignisse anzugeben.

Die nachfolgende Definition ist die formale Beschreibung einer Markov Kette:

Definition 4.2.1 Sei $S = \{s_1, s_2, \ldots, s_k\}$ ein endlicher Zustandsraum und P eine $(k \times k)$-

Matrix mit den Elementen $\{P_{i,j} : i, j \in \{1, \ldots, k\}\}$. Ein Zufallsprozess $X = (X^{(0)}, X^{(1)}, \ldots)$ wird **Markov Kette** genannt, falls für alle $n > 0$, alle $i, j \in \{1, \ldots, k\}$ und alle $i_0, \ldots, i_{n-1} \in \{1, \ldots, k\}$ gilt:

$$\begin{aligned} & P(X^{(n)} = s_j \mid X^{(0)} = s_{i_0}, X^{(1)} = s_{i_1}, \ldots, X^{(n-1)} = s_i) \\ = \;& P(X^{(n)} = s_j \mid X^{(n-1)} = s_i) \\ = \;& P_{i,j}. \end{aligned}$$

Die Elemente $P_{i,j}$ der Transitionsmatrix P werden *Transitionswahrscheinlichkeiten* genannt und besagen, wie wahrscheinlich der Übergang von einem Zustand i in einen Zustand j ist. Für diese Transitionswahrscheinlichkeiten gilt:

$$P_{i,j} \geq 0, \text{für alle } i, j \in \{1, \ldots, k\} \tag{4.8}$$

sowie

$$\sum_{i=1}^{k} P_{i,j} = 1, \text{für alle } i \in \{1, \ldots, k\}\,. \tag{4.9}$$

Eine Markov Kette wird neben Transitionswahrscheinlichkeiten auch durch *Startwahrscheinlichkeiten* beschrieben. Diese entsprechen einer Wahrscheinlichkeitsverteilung über allen Zustände für den Zeitpunkt 0 und werden als Vektor dargestellt:

$$\mu^{(0)} = \begin{pmatrix} \mu_1^{(0)} \\ \mu_2^{(0)} \\ \vdots \\ \mu_k^{(0)} \end{pmatrix} = \begin{pmatrix} P(X^{(0)} = s_1) \\ P(X^{(0)} = s_2) \\ \vdots \\ P(X^{(0)} = s_k) \end{pmatrix}, \tag{4.10}$$

wobei folgendes gilt:

$$\sum_{i=1}^{k} \mu_i^{(0)} = 1. \tag{4.11}$$

Die nachfolgende Definition beschreibt den Begriff der stationären Verteilung, der für den weiteren Verlauf der Arbeit von besonderer Bedeutung ist:

Definition 4.2.2 Der Vektor $\pi = (\,\pi_1 \quad \ldots \quad \pi_k\,)$ heisst **stationäre Verteilung** einer Markov Kette, wenn folgendes gilt:

1. $\pi_i \geq 0$ für alle $i \in \{1, \ldots, k\}$ und $\sum_{i=1}^{k} \pi_i = 1$, und
2. $\sum_{i=1}^{k} \pi_i P_{i,j} = \pi_j$ für alle $j \in \{1, \ldots, k\}$.

4.2.2 Einführung in MCMC-Algorithmen

Monte Carlo-Methoden wurden erstmals in den 1940er Jahren zur Simulation komplexer Zufallsprozesse angewendet. Sie gehören heute zu den wichtigsten numerischen Verfahren und lassen sich auf viele naturwissenschaftliche, technische und medizinische Probleme anwenden. Die Namensgebung ist eine Anspielung auf den für Glücksspiele bekannten Ort Monte Carlo, da die Grundlage des Verfahrens Zufallszahlen sind, wie sie auch an einem Roulette-Tisch erzeugt werden können. Bei MCMC-Algorithmen handelt es sich um eine Unterklasse der Monte Carlo-Methoden. MCMC-Algorithmen generieren Stichproben aus einer grossen Zustandsmenge, die einer bestimmten aber unbekannten Verteilung entsprechen. Mit Hilfe einer oder mehrerer Markov-Ketten wird versucht, die Zustandsverteilung zu approximieren.

Die Menge der Äquivalenzrelationen bei der Entity-Resolution bilden ebenfalls einen großen Zustandsraum. Eine exakte Inferenz ist daher nicht möglich. Mit Hilfe von MCMC-Algorithmen kann aber eine Wahrscheinlichkeitsverteilung aller Zustände approximiert werden, aus der dann der wahrscheinlichste Zustand ermittelt wird, von dem man annimmt, dass er die optimale Äquivalenzrelation repräsentiert. Dies soll in diesem Kapitel genauer untersucht werden. Im folgenden werde zwei MCMC-Algorithmen vorgestellt.

4.2.3 Der Gibbs Sampler

Der Gibbs-Sampler generiert eine Folge von Stichproben zwei oder mehrerer Zufallsvariablen, die auf einer gemeinsamen, jedoch unbekannten Wahrscheinlichkeitsverteilung basieren. Ziel des Gibbs-Samplers ist es aus der Stichprobenmenge die ihr zugrundeliegende Verteilung zu approximieren. Der Gibbs-Sampler wurde von Geman et al. [10] entwickelt und ist nach J.W. Gibbs benannt, der ein ähnliches Verfahren im Bereich der statistischen Physik entworfen hatte.

Die nachfolgende Charakterisierung des Gibbs-Samplers ist angelehnt an [25]. Seien:

- $\mathcal{X} = \{X_1, \ldots, X_n\}$ eine Menge von Zufallsvariablen, die eine bestimmte zu modellierende Situation charakterisieren,
- $x_i \in WB(X_i)$ eine Belegung der Zufallsvariablen X_i,

- $x = (x_1, \ldots, x_n) \in WB(X_1) \times \ldots \times WB(X_n)$ die Belegung eines n-Tupels von Zufallsvariablen,
- $\mathcal{X}^{(t)} = \left\{X_1^{(t)}, \ldots, X_n^{(t)}\right\}$ diee Menge der Zufallsvariablen zum Zeitpunkt t,
- $x^{(t)} = (x_1^{(t)}, \ldots, x_n^{(t)}) \in WB(X_1) \times \ldots \times WB(X_n)$ eine Belegung der Zufallsvariablen zum Zeitpunkt t.

Wie in Abschnitt 4.2.1 beschrieben, ist eine Markov-Kette durch eine Initialverteilung für $\mathcal{X}^{(0)}$ und eine Transitionsmatrix P definiert. Die sequentielle Anwendung von P entspricht dabei der Simulation einer Markov-Kette $\mathcal{X} = (\mathcal{X}^{(0)}, \mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots)$. Der Übergang von $\mathcal{X}^{(t-1)}$ nach $\mathcal{X}^{(t)}$ geschieht beim Gibbs-Sampler derart, dass iterativ jede Zufallsvariable in Abhängigkeit aller anderen Zufallsvariablen bestimmt wird:

- Wähle $x_1^{(t)} \in WB(X_1)$, gegeben $(x_2^{(t-1)}, \ldots, x_n^{(t-1)})$
- Wähle $x_2^{(t)} \in WB(X_2)$, gegeben $(x_1^{(t)}, x_3^{(t-1)}, \ldots, x_n^{(t-1)})$

  ...
- Wähle $x_k^{(t)} \in WB(X_k)$, gegeben $x_1^{(t)}, \ldots, x_{k-1}^{(t)}, x_{k+1}^{(t-1)}, \ldots, x_n^{(t-1)}$

  ...
- Wähle $x_n^{(t)} \in WB(X_n)$, gegeben $x_1^{(t)}, \ldots, x_{n-1}^{(t)}$

Die Belegung der Variablen x_k ist dabei abhängig von der Belegung aller anderen Variablen und wird auf Basis der Transitionsmatrix P generiert.

4.2.4 Der Metropolis Algorithmus

Der Metropolis Algorithmus hat viele Parallelen zum Gibbs-Sampler, er ist jedoch allgemeiner verwendbar. Er wurde zum ersten Mal von Metroplos et al. [24] im Rahmen einer Problemanalyse auf dem Gebiet der statistischen Physik vorgestellt wurde und von ... erweitert. Die nachfolgende Definition wurde von R.M. Neal [25] aufgestellt. Seien:

- $\mathcal{X} = \{X_1, \ldots, X_n\}$ eine Menge von Zufallsvariablen,
- $S_k(x, x_k^*)$ ein Wahrscheinlichkeitsverteilung für Zustandsübergänge und
- $A(x, x^*)$ eine Akzeptanzmatrix, die die Wahrscheinlichkeit enthält, daß ein Zu-

standsübergang akzeptiert wird.

Die folgenden zwei Schritte werden iterativ ausgeführt:

1. Wähle einen beliebigen Zustand x^*, in welchem alle Komponenten mit Ausnahme von x_k identisch mit dem Zustand x sind, während x_k^* gemäss der Verteilung $S_k(x, x_k^*)$ ausgewählt wird.

2. Der Zustand x^* wird mit der Akzeptanzwahrscheinlichkeit $A(x, x^*)$ akzeptiert; andernfalls wird im Zustand x verblieben. Die Akzeptanz kann mittels einer Zufallsvariablen u gesteuert werden; sei diese gleichverteilt auf $[0, 1]$, wobei über den Nachfolgezustand x' wie folgt entschieden wird:

$$x' = \begin{cases} x^* & \text{wenn } u < A(x, x^*), \\ x & \text{sonst.} \end{cases} \tag{4.12}$$

Beim Metropolis-Algorithmus erfolgt hingegen eine zielgerichtete Änderung der Belegung einer Variablen $X_i.A$ von x zu x'. Dies geschieht allerdings nicht zwingend, sondern wird mittels einer Akzeptanzwahrscheinlichkeit reguliert:

$$P(X_i.A = x' \mid X_i.A = x) = \min\left(1, \frac{P(X_i.A = x' \mid E(X_i.A))}{P(X_i.A = x \mid E(X_i.A))}\right). \tag{4.13}$$

Der Metropolis-Algorithmus hat gegenüber dem Gibbs-Sampler den Vorteil, dass nicht alle potentiellen Belegungen der Variablen X_i bei der Berechnung berücksichtigt werden müssen

4.2.5 Konvergenz von MCMC-Algorithmen

Ein zentrales Problem bei der Simulation von Markovketten ist es denjenigen Zeitpunkt zu ermitteln, an dem die Markovkette ihren stationären Zustand π erreicht hat. Man spricht hierbei von Konvergenz. Tierney [] zeigt, daß es im allgemeinen nicht möglich ist, die Konvergenz *einer einzelnen* Markovkette zu messen. Ein alternativer Ansatz, basierend auf der parallelen Simulation mehrer Markovketten, wird von Gelman [9] näher untersucht. Die Grundidee besteht darin, mehrere Markovketten mit unterschiedlichem Startzuständen über einen längeren Zeitraum zu iterieren. Sind die Zustandsverteilungen aller Ketten einander ähnlich, nimmt man an, die Ketten sind konvergiert.

Zur Konvergenzmessung wird ein Skalar verwendet, der den Zustand, in dem sich die Markov-Kette befindet repräsentiert.

Als Basis dient dabei ein Skalar ψ, welches für eine Markovkette $\mathcal{X}$ zu jedem Zeitpunkt ermittelt wird. Dieses wird durch die Funktion f realisiert, die den Zustand der Markovkette zum Zeitpunkt k auf das Intervall $[0, 1]$ abbildet:

$$f : \mathcal{X}^{(k)} \rightarrow [0, 1] \tag{4.14}$$

Es werden m parallel Markovketten $\mathcal{X}_1 \ldots \mathcal{X}_m$ der Länge n simuliert. Dann stellt ψ_{ij} , mit $i = 1, \ldots, m$ und $j = 1, \ldots, n$, das Skalar der Markovkette i zum Zeitpunkt j, kurz $\mathcal{X}_i^{(j)}$, dar.

Für die Konvergenzberechnung werden zwei Größen herangezogen. Die Intraketten-Varianz W misst die durchschnittliche Varianzen der Zustandsverteilungen über allen Ketten und wird wie folgt berechnet:

$$W = \frac{1}{m}\sum_{i=1}^{m} s_i^2, \; wobei \; s_i^2 = \frac{1}{n-1}\sum_{j=1}^{n}(\psi_{ij} - \overline{\psi}_{i.})^2 \; und \; \overline{\psi}_{i.} = \frac{1}{n}\sum_{j=1}^{n}\psi_{ij}. \tag{4.15}$$

Die Interketten-Varianz B misst das Verhältnis der durchschnittliche Varianzen jeder einzelnen Kette zur Varianz aller Ketten:

$$B = \frac{n}{m-1}\sum_{i=1}^{m}(\overline{\psi}_{i.} - \overline{\psi}_{..}), \; wobei \; \overline{\psi}_{..} = \frac{1}{m}\sum_{i=1}^{m}\overline{\psi}_{i.}. \tag{4.16}$$

Aus beiden Varianzen werden zwei Schätzer gemittelt. Der konservative Schätzer $\widehat{var}(\psi)$ berechnet sich wie folgt:

$$\widehat{var}(\psi) = \frac{n-1}{n}W + \frac{1}{n}B. \tag{4.17}$$

Der ïrealistischeS̈chätzer $\widehat{R}$ berechnet sich folgendermassen:

$$\widehat{R} = \frac{\widehat{var}(\psi)}{W}. \tag{4.18}$$

Unterschreitet der konservative Schätzer $\widehat{R}$ einen bestimmten Schwellwert, so geht mann von einer Konvergenz der Markov-Ketten aus.

5 Das PRM zur Lösung der Entity-Resolution

In diesem Kapitel wird die eingangs beschriebene Problemstellung aufgegriffen. Es wird ein PRM vorgestellt, mit welchem das Problem der Entity-Resolution gelöst werden kann. Es wird gezeigt, wie für dieses Modell eine existierenede Abhängigkeitsstruktur gelernt und deren Parameter geschätzt werden können. Anschließend wird ein MCMC-Algorithmus vorgestellt, mit dem Unsicherheiten durch Inferenzen aufgelöst und ein Zitationsgraph generiert werden kann.

5.1 Erweiterung des klassischen PRM

Betrachtet man sich die in Kapitel 2 beschriebene Problemstellung der Entity-Resolution, so ist erkennbar, daß mehrere Datenpunkte dieselbe Entität repräsentieren können. Dies sprengt den Rahmen des von Friedman et al. [7] beschriebenen PRM, da darin keine Modellierug äquivalenter Datenpunke möglich ist. Daher ist eine Erweiterung dieses Modells notwendig. Zentrales Element des erweiterten Modells ist eine Äquivalenzrelation. Angelehnt an die in Kapitel 2 beschriebene Problemstellung der Entity-Resolution, dient die Äquivalenzrelation dazu, Datenpunkte, die dieselbe Entität beschreiben zu gruppieren. Ein weiterer Unterschied zum dem von Friedman et al. beschriebenen Modell ist die Unterscheidung von *festen* und *probabilistischen* Klassen.

Für das erweiterte Modell müssen dabei zwei Aspekte berücksichtigt werden. Zum einen ist es notwendig, die in Abbildung 2.2 dargestellte Vielfalt an Publikationsformen zu modellieren. Zum anderen soll das Modell die Basis für einen Zitationsgraphen bilden. Eine Möglichkeit der Modellierung ist es, jeder Publikationsform eine eigene Klasse zuzuordnen,

in der die obligatorischen Metadaten durch Attributmengen repräsentiert werden. Dieser Ansatz birgt allerdings Redundanzen, da sich die Attributmengen in vielen Elementen (Autor, Titel, Ort, Jahr, Seitenangaben) überschneiden würden. Aus diesem Grund wird eine Modellierung bevorzugt, bei der eine einzelne Klasse alle Publikationstypen und deren Metadaten repräsentiert.

Um Redundanzen durch nicht belegte Attribute zu vermeiden, werden in diesem Modell nicht alle Metadaten berücksichtigt. Zum einen werden solche Metadaten modelliert, die für alle Publikationsformen vorherrschen (*Autor*, *Titel*, *Jahr*), zum anderen Metadaten die für einen Publikationstyp unabdingbar sind (der Buchtitel eines Kapitels, die *Zeitschrift* eines Artikels und die Konferenz eines Papers). Da ein Paper immer in einem Buch mit dem Titel der Konferenz publiziert wird, werden Konferenz und Buchtitel zum Attribut *Buch* zusammengefasst.

Der in Kapitel 2 definierte Zitationsgraph besteht aus einer Menge von Publikationen und Zitationen. Publikationen werden im Modell durch die Klasse *Publikation* repräsentiert und Zitationen durch die Relation ψ. Die Grundlage einer Zitation bilden die Metadaten der referenzierenden und der referenzierten Publikation. Beide Informationen werden als eigenständige Entitäten der Klasse *Datenpunkt* repräsentiert und die Beziehung beider Datenpunkte zueinander durch die Relation ϕ. Die unsichere Information, ob ein Datenpunkt eine bestimmte Publikation beschreibt, wird durch die χ dargestellt. Das Modell besteht also aus den folgenden Komponenten:

- Die feste Klasse D beschreibt einen Datenpunkt, der eine Literaturreferenz repräsentiert
- Die probabilistische Klasse P beschreibt Publikationen, welche durch einen oder mehrere Datenpunkte repräsentiert werden.
- Die Attributmengen sind für beide Klassen identisch: *Autor*, *Titel*, *Jahr*, *Bucht* und *Zeitschrift*; im Falle der Klasse D sind sie fest, für die Klasse P sind die probabilistisch.
- Die feste Relation $\phi(d_i, d_j)$ mit $d_i, d_j \in D$ beschreibt die Beziehung zwischen zwei Datenpunkten. Dabei wird die z
- Die probabilistische Relation $\chi(d, p)$, mit $d \in D$ und $p \in P$, beschreibt die Zu-

gehörigkeit eines Datenpunktes zu einer Publikation

- Die probabilistische Relation $\psi(p_i, p_j)$ beschreibt die aus ϕ und χ resultierenden Kanten des Zitationsgraphen, die Zitationen.

Die Skelettstruktur σ beschreibt alle Entitäten der Klasse D sowie deren Attributbelegungen. Ebenso spezifiziert σ alle Tupel der Relation ϕ. Unbestimmt bleiben allerdings die Anzahl der Publikationen und damit einhergehend die Tupel der Relationen χ und ψ. Eine Instanz I des Schemas bstimmt die Entitäten der Klasse P, deren Attributbelegungen sowie Tupel der Relationen χ und ψ. Ausgehend von den beschriebenen Relationen lassen sich verschiedene Schlüssel und Schlüsselketten definieren:

- Der Schlüssel $\rho_1(D, D)$ beschreibt den Verweis einer Quellpublikation auf eine Zielpublikation,
- Der Schlüssel $\rho_2(D, P)$ modelliert die Zugehörigkeit eines Datenpunktes zu einer Publikation,
- die Schlüsselkette $\tau = \rho_1\rho_2$ modelliert die entfernte Beziehung eines Datenpunktes zu einer Publikation

Die Tupel der Relation ψ resultieren aus den Tupeln der Relationenen ϕ und χ, wobei Folgendes gilt:

$$\forall\, p_1, p_2 : (p_1, p_2) \in \psi \;\Leftrightarrow\; \exists\, d_1, d_2 : (d_1, p_1) \in \phi \;\wedge (d_2, p_2) \in \phi \;\wedge (p_1, p_2) \in \psi. \qquad (5.1)$$

Eine Äquivalenzrelation ω wird dabei durch die probabilistische Relation χ realisert und bezeichnet alle Datenpunkte, die dieselbe Publikation repräsentieren. Auf dem Raum aller Äquivalenzrelationen kann durch ein Inferenzverfahren, daß in Abschnitt 5.4 beschrieben wird, die optimale Äquivalenzrelation approximativ ermittelt werden.

5.2 Schätzen der Modellparameter

Zur Bestimmung der Parameter θ_S des PRM wird ein ML-Schätzer mit Prior verwendet. Da es sich hierbei um eine Multinomialverteilung handelt, wird ein konjugierter Prior zur Multinomialverteilung, der *Dirichlet-Prior*, verwendet. Dieser ist durch die Hyperparameter α charakterisiert, wobei für alle Paare aus Attributwert $X.A = v$ und Belegung

der Elternmenge $E(X.A) = u$ ein $\alpha[v, u]$ existiert. Die Bestimmung von θ_S wird in dieser Arbeit durch Gleichung 5.2 realisert:

$$P(X.A = v \mid E(X.A) = u) = \frac{C_{X.A}\left[v, u\right] + \alpha\left[v, u\right]}{\sum_{v'} C_{X.A}\left[v', u\right] + \alpha\left[v', u\right]}. \tag{5.2}$$

Bei der Auswahl der Hyperparameter wurden zwei Alternativen in Erwägung gezogen. Die Hyperparameter $\alpha_1[v, u]$ wurden für alle $X.A = v$ und $E(X.A) = u$ auf den Wert 1 gesetzt. Dies hat zur Folge, daß ungesehene Daten durch das Modell repräsentiert werden. Alternativ zum konstanten Wert wurde ein zweiter Prior α_2 verwendet, der durch ein Ähnlichkeitsmass charakterisiert ist:

$$\alpha_2\left[v, u\right] = 0.1* \sim \left[v, u\right], \tag{5.3}$$

wobei $\sim [[v, u]$ eine Zahl aus dem Intervall $[0, 1]$ ist, die bei starker Ähnlichkeit von v und u den Wert 1 annimmt und den Wert 0, wenn u und v keine Ähnlichkeit aufweisen.

5.3 Lernen der Abhängigkeitsstruktur

Das Lernen der Abhängigkeitsstruktur basiert auf den drei in Abschnitt 4.1.4 beschriebenen Schritten:

- Einschränkung des Suchraumes
- Bewertung von Strukturen
- Finden eines Suchalgorithmus

Einschränkung des Suchraumes
Wie in Abschnitt 4.1.2 beschrieben besteht ein PRM neben einem Schema aus zwei Bestandteilen, einer Abhängigkeitsstruktur S und deren Parameter θ_S. Bei der Struktur S handelt es sich um die jeweiligen Elternmengen der probabilistischen Attribute. Im obigen PRM sind das die Attribute der Klasse P. Die Definition eines PRM lässt als Elternattribute nur feste Attribute zu. Daher kommen Attribute der Klasse P nicht in Frage. Als potentielle Elternattribute kommen daher nur die festen Attribute der Klasse D in Frage. Damit ist gewährleistet, daß kein Attribut der Klasse P direkt oder indirket von einem anderen Attribut der Klasse P abhängig ist. Da Abhängigkeiten nur für probabilistische

Attribute in Frage kommen, werden Abhängigkeitszyklen vermieden, was eine notwendige Voraussetzung für ein konsistentes Modell ist.

Bewertung von Strukturen

Prinzipiell lassen sich zwei Arten von Abhängigkeiten unterscheiden. *Direkte* Abhängigkeiten beschreiben den unmittelbaren Einfluss eines Datenpunktes auf eine Publikation, der er zugeordnet ist. Diese Zuordnung wird durch die Relation χ bzw. den Schlüssel ρ_2 realisiert. In ihr spiegelt sich der (triviale) Sachverhalt wieder, daß die probabilistischen Attribute der Publikation, dieselben Wert annehmen können wie die äquivalenten Attribut der Datenpunkte, z.B. $D.Titel \rightarrow P.Titel$. Aber auch Abhängigkeiten zwischen unterschiedlichen Attributen sind denkbar, so z.B. $D.Titel \rightarrow P.Jahr$ oder $D.Autor \rightarrow P.Jahr$. *Indirekte* Abhängigkeiten beschreiben hingegen den Einfluß entfernter Datenpunkte auf Publikationen. Dahinter verbirgt sich die Intuition, daß aus dem Wissen über die Metadaten einer Publikation, Wissen über referenzierte Publikationen abgeleitet werden kann. Dieses wird durch die Relationen ϕ und ψ bzw. den Schlüssel ρ_1 realisiert.

Die Bewertung einer Struktur beruht auf Gleichung 4.7, die zum besseren Verständnis an dieser Stelle noch einmal aufgeführt ist.

$$P(S \mid \mathcal{I}, \sigma) = P(S)P(\mathcal{I} \mid S, \sigma). \tag{5.4}$$

Beide Faktoren der Gleichung können dabei separat betrachtet werden. Der Faktor $P(S)$ beschreibt dabei die Wahrscheinlichkeit einer Struktur. Der in dieser Arbeit angewandte Ansatz berachtet die Strukturen dabei als gleichwahrschienlich. Dadurch reduziert sich die Bewertung einer Struktur S auf die Likelihood der Instanz I zu dieser Struktur und der definierten Skelettstruktur σ.

Komplizierter stellt sich die Berechnung des zweiten Faktors der Gleichung dar. Dieser berechnet sich aus der Randverteilung von $\mathcal{I}$ über θ_S, sowie den Parametern der Struktur S:

$$P(\mathcal{I} \mid S, \sigma) = \int P(\mathcal{I} \mid S, \sigma, \theta_S)P(S, \theta_S)d\theta_S. \tag{5.5}$$

Die Verwendung eines Dirichlet-Priors für die Parameterschätzung (vgl. Gleichung 5.2) führt dazu, daß das komplexe Integral in ein Produkt aus einfach zu lösenden Integralen aufgelöst werden kann. Die Dirichlet-Modell-Gleichung ist das zentrale Element der durch Auflösung der Integralgleichung erhaltenen Produktgleichung. Für jedes probabilistische Attribute der Klasse P wird die Elternmenge ermittelt und mitsamt aller geschätzten

Modellparameter in die Dirichlet-Modell-Gleichung (DM) eingesetzt. Anschließend werden die Ergebnisse aller Attribute miteinander multipliziert:

$$P(\mathcal{I} \mid S, \sigma) = \prod_{\mathcal{A}(P)} \prod_{E(P.A)} DM(\{C_{P.A}[v,u]\}, \{\alpha_{P.A}[v,u]\}), \tag{5.6}$$

Die Dirichlet-Modell-Gleichung hat dabei die folgende Form:

$$DM(\{C[v]\}, \{\alpha[v]\}) = \frac{\Gamma(\sum_v \alpha[v])}{\Gamma(\sum_v (\alpha[v] + C[v]))} \prod_v \frac{\Gamma\alpha[v] + C[v]}{\Gamma(\alpha[v])}, \tag{5.7}$$

wobei

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \tag{5.8}$$

die *Gammafunktion* ist, die für natürliche Zahlen wie folgt definiert ist: $\Gamma(x) = (x-1)!$. Diese Gleichung wird bei der Vorstellung des Suchalgorithmus erneut aufgegriffen.

Suchalgorithmus

Trotz der Einschränkung des Suchraumes auf legale Strukturen ist dieser immer noch zu groß, um mittels eines deterministischen Verfahrens die bestbewertetste Struktur zu ermitteln. Werden für das zu Anfang dieses Kapitels eingeführte PRM nur direkte Abhängigkeiten betrachtet, so ist die Größe des Suchraumes äquivalent zur Potenzmenge der festen Attribute der Klasse D und umfaßt 2^5 Elemente. Werden darüberhinaus auch indirekte Abhängigkeiten betrachtet, so vergrössert sich dieser sogar auf 2^{10} Elemente.

Aus diesem Grunde ist es notwendig für die Suche nach einer hochbewerteten Struktur eine Heuristik anzuwenden. Zu den bekanntesten heuistischen Suchverfahren zählen Greedy-Hill-Algorithmen, die auf folgendem Prinzip beruhen. Ausgehend von einem Startzustand wird die Menge aller von diesem Zustand aus erreichbaren Zustände ermittelt und jeder dieser Zustände bewertet. Besitzt der bestbewerteste Zustand dieser Menge eine höhere Bewertung als der aktuelle Zustand, zu Beginn ist dies der Startzustand, so wid dieser Zustand zum aktuellen Zustand. Anschliessend werden wieder alle erreichbaren Zustände ermittelt, bewertet und mit dem aktuellen Zustand verglichen. Der Algorithmus bricht ab, wenn sich in der ermittelten Menge kein Zustand mit einer höheren Bewertung als die des aktuellen Zustandes befindet. Abschließend wird der aktuelle Zustand als Ergebnis ausgegeben.

Der wichtigste Kritikpunkt am Greedy-Hillclimbing-Algorithmus ist der, daß er häufig nur ein lokales Optimum und selten das globale Optimum findet. Die Wahrscheinlichkeit,

daß der Algorithmus das globale Optimum findet steigt allerdings, wenn er mehrmals mit unterschiedlichen Startzuständen durchgeführt wird. Nachfolgend wird ein Greedy-Hillclimbing-Algorithmus vorgestellt, der auf dem eingeschränkten Suchraum operiert. Aus Gleichung 5.6 folgt, daß die Elternmenge eines Attributs unabhängig von denen der anderen ermittelt werden kann. Diese Eigenschaft wird im Folgenden bei der Formulierung des Suchalgorithmus berücksichtigt.

Die Arbeitsweise des Algorithmus' ist die folgende: Zunächst wird ein probabilistisches Attribut A der Klasse P gewählt, fü welches $E(P.A)$ noch nicht definiert ist. Zunächst wird für alle Attribute, die über den Schlüssel ρ_2 mit $P.A$ verbunden sind, mit Hilfe der DM-Gleichung 5.7 ein Wert ermittelt. Das Attribut mit dem höchsten Wert wird in die anfangs leere Eltermenge aufgenommen. Anschließend wird überprüft, ob sich dieser Wert durch Hinzunahme weiterer Attribute vergrössert. Ist dies der Fall, wird die Eltermenge um dieses Attribut erweitert. Kommt es in einem Iterationschritt zu keiner Verbesserung, werden in einer zweiten Schleife alle Attribute, die über die Schlüsselkette τ mit $P.A$ verbunden sind, überprüft.

5.4 Auflösung der Unsicherheiten

Das zu Beginn dieses Kapitels beschriebene Modell ist durch mehrere Unsicherheiten gekennzeichnet. Dabei handelt es sich zum einen um die Belegung der probabilistischen Attribute der Publikationen, zum anderen um die Zuordnung von Datenpunkten zu Publikationen. Zunächst wird beschrieben, wie der in Abschnitt 4.2.3 eingeführte Gibbs-Samplers dazu genutzt werden kann, die Belegungen der probabilistischen Publikations-Attribute zu bestimmen. Anschließsend wird gezeigt, wie durch Anwendung eines MH-Algorithmus die Zuordnungen der Datenpunkte zu Publikationen hergestellt werden.

5.4.1 Wertzuweisung der probabilistischen Attribute

Die Wertzuweisung der Attribute kann durch einen Gibbs-Sampler realisiert werden. Dieser wählt zufällig eine Entität $p \in P$ und bestimmt alle Entitäten der Klasse D, die in direkter Relation zu p stehen. Anschließend wird für jedes Attribut von p zufällig eine in Relation zu p stehende Entität $d \in D$ gewählt und dessen Belegung dieses Attributs

Algorithm 1 Greedy-Hillclimbing-Algorithmus zur Bestimmung der Abhängigkeitsstruktur

for all $P.A$ in $\mathcal{A}(P)$ **do**
  $E(A) \leftarrow \oslash$
  $MaxValue \leftarrow 0$
  $MaxA \leftarrow null$
  $flag \leftarrow true$
  $slot \leftarrow \rho_2$
  while $flag = true$ **do**
    $E(A) \leftarrow E(A) \cup MaxA$
    for all $D.A$ in $\mathcal{A}(D)$ **do**
      $flag \leftarrow false$
      $Value \leftarrow P(P.A \leftarrow E(A) \cup D.A)$
      if $Value > MaxV$ **then**
        $MaxValue \leftarrow Value$
        $MaxA \leftarrow D.A$
        $flag \leftarrow true$
      end if
    end for
    if $slot = \rho_2$ **then**
      $flag \leftarrow true$
      $slot \leftarrow \tau$
    end if
  end while
end for

an p übergeben. Ist dieses Attribut für d nicht belegt wird eine andere Entität zufällig bestimmt. Dieses Vorgehen führt dazu, daß häufig vorkommende Attributwerte mit einer grossen Wahrscheinlichkeit ausgewählt werden und nichtbelegte Attribute für p vermieden werden.

Algorithm 2 Gibssampling zur Bestimmung der probabilistischen Attribute einer Entität $p \in P$

```
for all P.A ∈ A(P) do
  p.A ← null
  for all P.A ∈ A(P) do
    Liste ← ⊘
    for all d ∈ ρ*_2.p do
      if e = d.E(P.A) ≠ null then
        Liste ← Liste ∪ e
      end if
    end for
    if Liste ≠ ⊘ then
      p.A ← rand(Liste)
    end if
  end for
end for
```

5.4.2 Bestimmung der probabilistischen Relationen

Die zweite Unsicherheit des PRMs ist die Relation χ, die jedem Datenpunkt eine Publikation zuordnet. Diese Unsicherheit wird für das beschriebene Modell durch einen MH-Algorithmus aufgelöst. Dieser ist angelehnt an den in Abschnitt 4.6 beschriebenen Algorithmus. Dabei wird zunächst zufällig eine von drei Zustandsänderungen gewählt:

- Publikationsfusion
- Publikationsteilung
- Publikationswechsel

Ein Zustand des MH-Algorithmus' entspricht dabei einer Instanz I des Schemas. Eine Zustandsänderung erzeugt also eine neue Instanz. Für eine gewählte Zustandsänderung wird die Likelihood des aktuellen und des neuen Zustandes ermittelt. Die Likelihood eines Zustandes berechnet sich aus der Summe der Likelihood aller Publikationen. Die Likelihood einer Publikation basiert wiederum auf der Likelihood ihrer Attributwerte in Bezug auf die Belegung der Elternattribute aller ihr zugeordneten Datenpunkte und basiert auf Gleichung 5.2. Daraus ergibt sich die folgende Gleichung für die Berechnung der Likelihood einer Instanz I, die durch diesen Zustand repräsentiert wird:

$$l(I) = \sum_{p \in P} \sum_{d : d.\rho_2 = p} \sum_{A.P} P(p.A \mid E(d.A)) \tag{5.9}$$

Da die Likelihood einer Publikation unabhängig von allen anderen Publikationen ist, müssen bei einer Zustandsänderung nur die davon betroffenden Publikationen neu berechnet werden. Liegt der Quotient von neuer und alter Likelihood über einem Schwellwert, der einer Zufallszahl aus einer [0,1]-Gleichverteilung entspricht, wird die Zustandsänderung akzeptiert und der Schritt ausgeführt, andernfalls verworfen.

Publikationsfusion
Bei einer Publikationsfusion werden zufällig zwei Publikationen $p_i, p_j \in P$ gewählt. Der neue Zustand ergibt sich durch die Änderung der Relation χ für alle Datenpunkte, die in Relation zu p_i stehen. Diese stehen nun in Relation zu p_j.

Publikationssteilung
Bei einer Publikationssteilung wird zufällig eine Publikation $p_i \in P$ gewählt. Anschließend wird eine neue Publikation p_j generiert. Für alle Datenpunkte, die in Relation zu p_i stehen wird zufällig entschieden, ob diese in Relation zu p_j gehen.

Publikationsswechsel
Bei einem Publikationswechsel werden zufällig zwei Publikationen $p_i, p_j \in P$ gewählt. Für jeden Datenpunkt aus p_i wird zufällig entschieden, ob dieser im neuen Zustand in Relation zu p_j steht.

Die *zufällige Auswahl* der Publikationen und deren Datenpunkte kann mit der Suche einer Nadel im Heuhaufen verglichen werden. Insbesondere bei einer grosen Anzahl von Publikationen und Datenpunkten ist die Wahrscheinlichkeit sehr gering, den optimalen Zustand zu erreichen. Deswegen werden an dieser Stelle zwei Herangehensweisen vorgestellt, die durch eine zielgerichtete Auswahl charakterisiert sind.

Bei der *strategischen Auswahl* werden für alle jeder der drei Zustandsänderungen ein Wert berechnet, der die Wahrscheinlichkeit angibt, daß die Zustandsänderung in Richtung des optimalen Zustandes erfolgt. Im Zentrum steht dabei die Idee, die Wahrscheinlichkeit für die Zuordnung eines Datenpunktes zu einer Publikation. Dies entspricht der der Likelihood eines Datenpunktes zu einer Publikation:

$$l(d \in D \mid p \in P) = \sum_{A.P} P(p.A \mid E(d.A)) \tag{5.10}$$

Bei der Publikationsfusion werden nun die beiden Publikationen bestimmt, deren Datenpunkte durchschnittlich die grösste Likelihood zu einer gemeinsamen Publikation aufweisen. Bei der Publikationsteilung wird die Publikation bestimmt, dessen Datenpunkte durchschnittlich die geringste Likelihood zu ihrer Publikation aufweisen. Bei einem Publikationswechsel wird zur Publikation, deren Datenpunkte die geringste Likelihood zu ihr aufweisen, die Publikation bestimmt, mit deren Datenpunkten sie die grösste Likelihood aufweist.

Neben der zielgerichteten Auswahl der Publikationen für jede de drei Zustandsänderungen ist es sinnvoll die vielversprechendste Zustandsänderung zu bestimmen. Dies wird erreicht, indem zu jeder der drei Zustandsänderungen der daraus resultierende neue Zustand berechnet wird. Ausgewählt wird nun der Schritt, bei dem der neue Zustand die höchste Likelihood aufweist.

Die strategische Auswahl der Zustände birgt die Gefahr, daß sich der Algorithmus in einem suboptimalen Zustandsraum verfängt. Dies kann eintreten, wenn es zu einem sich immer wiederholenden Wechselspiel zwischen Publikationsfusion und Publikationsteilung kommt. Dies kann durch die *zufällig-strategische Auswahl* verhindert werden. Ähnlich der strategischen Auswahl werden für alle drei Zustandsänderungen die Publikationen zielgerichtet ausgewählt und die Likelihood der neuen Zustände berechnet. Die Wahrscheinlichkeit für die Auswahl einer Zustandsänderung ergibt sich aus dem Verhältnis seiner Likelihood zur Summe aller Likelihood-Werte.

5.4.3 Simulation mehrerer Markov-Ketten

Die der Simulation einer Markov-Kette gestaltet sich so, daß diese über einen längeren Zeitraum simuliert wird und zu jedem Iterationszeitpunkt der Zustand der Markov-Kette

gespeichert wird. Nach einer festen Anzahl von Schritten geht man davon aus, daß die Menge der abgespeicherten Zustände die tatsächliche Zustandsverteilung gut approximiert.

Die Simulation mehrere Markov-Ketten unterscheidet sich in zwei Punkten von der Simulation einer Markov-Kette. Zum einen ist es möglich ein Konvergenzkriterium anzuwenden, wodurch die Anzahl der Iterationsschritte nvon vornherein bekannt ist. Die Basis dieses Kriterium bildet die in Abschnitt 4.2.5 beschriebene Konvergenzmessung von Markov-Ketten. Diese benötigt zur Konvergenzmessung einen Skalar, der den Zustand einer Markov-Kette repräsentiert. Für das PRM wurde dazu ein Skalar ψ gewählt, dass auf der durchschnittlichen Likelihood aller Datenpunkte zu ihren Clustern basiert. Es berechnet sich wie folgt:

$$\sum_{d} d \in D \mid p = d.\rho_2 \tag{5.11}$$

Der zweite Unterschied ist die unterschiedliche Handhabung des Zustandsraumes. Einerseits kann dieser als gemeinsamer Zustandsraum aller Ketten betrachtet werden. Andererseits besteht die Möglichkeit einen Zustandsraum zu verwenden, der durch eine Funktion γ über der Menge aller Markov-Ketten zu einem bestimmten Zeipunkt charakterisiert ist. Diese sei wie folgt definiert:

$$\gamma(d_i, d_j) = \begin{cases} 1 & \text{wenn für alle Markov-Ketten } d_i \text{ und } d_j \text{ in derselben Äquivalenzklasse} \\ 0 & \text{sonst.} \end{cases} \tag{5.12}$$

5.4.4 Gesamter Algorithmus

Der Inferenz-Algorithmus zur Auflösung der Unsicherheiten kann als ein Wechselspiel aus Gibbs-Sampling und MH-Algorithmus angesehen werden. Während der Gibbs-Sampler Publikationen auf Basis der ihm zugeordneten Datenpunkte bewertet, reguliert der MH-Algorithmus die Relationen zwischen Datenpunkten und Publikationen. Dies führt wiederum zu einer neuen Belegung einzelner probabilistischer Attribute.

In jedem Iterationsschritt wird zufällig entschieden, ob eine Neubewertung der Entitäten durch den Gibbssampler erfolgt. Im Anschluss daran wird eine der drei beschriebenen Zustandsänderungen bewertet und verworfen oder akzeptiert. Abschließend wird in jeder

Iteration die Abbruchbedingung überprüft und gegebenenfalls der Algorithmus beendet und ein vollständige Instanz zurückgegeben. Bei der Simulation mehrerer Markov-Ketten ist das Abbruchkriterium die Konvergenz der Markov-Ketten. Bei der Simulation einer einzelnen Markov-Kette erfolgt der Abbruch nach einer festen Anzahl von Iterationen.

Algorithm 3 MCMC-Algorithmus zur Auflösung der Unsicherheiten

$finished \leftarrow false$
while $notfinished$ **do**
  for all Markov-Ketten $K_1, dots, K_n$ **do**
    if rand(1) ¿ 0.5 **then**
      for all $p \in P$ **do**
        Bestimme alle Attribute von p mit Gibbs-Sampler
      end for
    end if
    $\{ll_{fusion}, p_i, p_j\} \leftarrow PreprocessingPublikatonsfusion$
    $\{ll_{teilung}, p_k\} \leftarrow PreprocessingPublikatonsteilung$
    $\{ll_{wechsel}, p_l, p_m\} \leftarrow PreprocessingPublikatonswechsel$
    if strat = zufällig **then**
      Führe zufällig eine Zustandsänderung durch
    end if
    if strat = zufällig **then**
      Bestimme maximales ll und führe Zustandsänderung aus
    end if
    if strat = zufällig **then**
      Wähle eine Zustandsänderung auf Basis von ll_{fusion}, $ll_{teilung}$ und $ll_{wechsel}$ durch
    end if
  end for
  finished ← Prüfe Konvergenz für Markov-Ketten $K_1, \ldots, K_n$
end while

5.4.5 Generierung des Zitationsgraphen

Um einen Zitationsgraphen zu konstruieren ist es sinnvoll, die Menge der Datenpunkte in zwei disjunkte Mengen aufzuteilen: die Menge der referenzierenden Datenpunkte

D_1 und die der referenzierten Datenpunkte D_2 . Auf D_1 wird der im vorigen Abschnitt beschriebene MCMC-Algorithmus angewendet und solange iteriert, bis Konvergenz der Ketten angenommen werden kann. Anschließend wird für alle Datenpunkte aus D_2 diejenige Publiaktion p' ermittelt, die Gleichung $l(d \mid p \in P)$ maximiert. Liegt $l(d \mid p')$ oberhalb eines Schwellwertes, so wird der Datenpunkt in die Äquivalenzklasse eingefügt. Andernfalls bildet er eine eigene Äquivalenzklasse.

6 Experimente und Evaluierung

Ziel der Experimente ist es zu untersuchen, ob das in Abschnitt 5.1 vorgestellte PRM und der in Abschnitt 5.4.4 vorgestellte Algorithmus dazu geeignet sind, einen Zitationsgraphen generieren. Dazu werden zunächst Struktur und Parameter des Modells experimentell bestimmt. Anschließend werden verschiedene Strategien des MH-Algorithmus untersucht. Weitere Experimente zeigen das Verhalten des Algorithmus bei der Simulation mehrerer Markov-Ketten. Abschließend wird der Algorithmus in Bezug auf den Aufbau eines Zitationsgraphen untersucht und evaluiert. Die damit erzielten Ergebnisse werden mit denen eines Baseline-Verfahrens verglichen, welches einen Zitationsgraphen durch einen Zeichenkettenvergleich aufbaut. Die dafür notwendigen Experimente wurden auf dem Cora-Datensatz durchgeführt, der zuvor durch ein Preprocessing aufbereitet wurde.

6.1 Datenaufbereitung

Zur Durchführung der Experimente wurde der Cora-Datensatz von Andrew McCallum [20] verwendet. Ausschlaggebend für die Verwendung des Datensatzes waren die grosse Datenmenge und die freie Verfügbarkeit der Daten. Der Cora-Datensatz beinhaltet ca. 37.000 Publikationen und ca. 714.000 Zitationen. Im Idealfall existieren zu jeder Publikation Metadaten, die die bibliographischen Angaben der Publikation enthalten, sowie annotierte Literaturreferenzen, aus denen sich die Metadaten dieser konstruieren lassen. Publikationen, zu denen keine Metadaten exitieren werden im weiteren Verlauf der Experimente nicht berücksichtigt.

Jede Publikation im Datensatz ist darüberhinaus eindeutig mittels einer Zahl referenzierbar. Eine Zitation stellt dabei ein geordnetes Tupel zweier Zahlen dar. Zwar lässt sich

anhand der Publikationen und Zitationen ein Zitationsgraph automatisch rekonstruieren, jedoch existiert keine Zuordnung der Literaturreferenzen zu den eigentlichen Zitationen. Diese wurde mittels eines Verfahren hergestellt. Das eingesetzte Verfahren stellt dabei die Zuordnung einer Literaturreferenz zu einer existierenden Zitation her. Kern des Verfahrens bildet ein Zeichenkettenvergleich. Zunächst wird für eine Zitation der Titel der referenzierten Publikation ermittelt. Anschließend wird eine Liste aller Literaturreferenzen der referenzierenden Publikation ermittelt. Aus dieser Liste wird diejenige Referenz ermittelt, deren Titel die grösste Ähnlichkeit mit dem Titel der Zielpublikation hat. Dieser Schritt wird für alle Zitationen im Datensatz durchgeführt. Durch den Einsatz dieses Verfahrens reduziert sich die Menge der Publikationen auf ca. 17.000 und die Menge der Zitationen auf ca. 72.000.

6.2 Experimentaufbau

Die zur Durchführung der Experimente notwendigen Programme wurden vom Autor selbständig implementiert. Als Interpretersprache wurde Perl gewählt. Bei der zugrundeliegenden Hardware handelt es sich um einen 8 x UltraSPARC-III+ Prozessor mit 1,2 GHz Taktfrequenz und einem 64 Bit Kernel sowie 16 GB RAM. Alle Experimente wurden mit jeweils 10 Wiederholungen durchgeführt. In jedem Versuch wurden die zur Analyse notwendige Menge von Publikationen und Zitationen zufällig gezogen. Die in den Experimenten unterschiedliche Größe dieser Menge wurde durch zwei Parameter bestimmt. Der erste Parameter bestimmt die Anzahl der Publikationen, der zweite bestimmt, wieviele Zitationen maximal auf diese Publikation verweisen. Die Parameter wurden dabei stets im Verhältnis 2:1 gewählt. Zur Evaluierung des MCMC-Algorithmus und des Baseline-Verfahrens wurden Precision, Recall und Jacard verwendet, die in Kapiel 2 eingeführt wurden.

6.3 Ergebnisse

6.3.1 Lernen der Abhängigkeitsstruktur

Für das Lernen der Abhängigkeitsstruktur bestand die zu analysierende Datenmenge aus 200 Publikationen sowie maximal 100 Zitationen pro Publikation. Auf dieser Menge wurden anschliessend ein Abhängigkeitsstruktur gelernt, d.h. für jedes probabilistische Attribut der Klasse P die Elternmenge bestimmt. Dieses wurde für die in Abschnitt 5.2 beschriebenen Ähnlichkeitsmaße α_1 und α_2 durchgeführt.

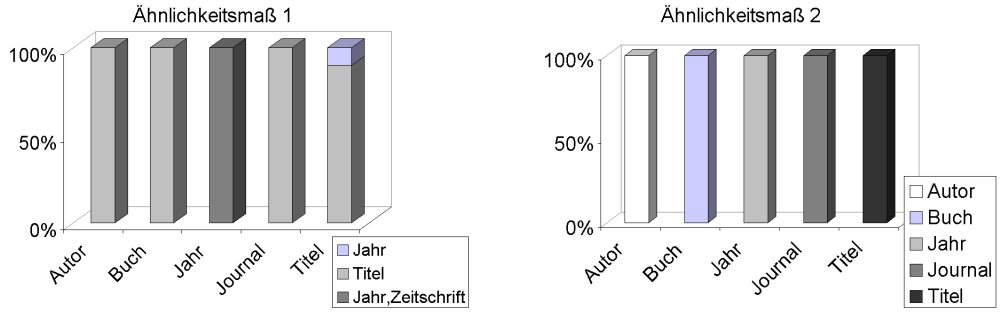

Abbildung 6.1: Gelernte Abhängigkeiten für beide Ähnlichkeitsmaße

Die Ergebnisse dieser Experimente sind in Abbildung 6.1 dargestellt. Auf der x-Achse ist die probabilistische Attributmenge der Klasse P abgebildet, auf der y-Achse ist die kummulierte Wahrscheinlichkeit der einzelnen Elternmengen dargestellt. Die Ergebnisse für das Ähnlichkeitsmaß α_1 zeigen eine deutliche Dominanz des Attributs Titel für die Attribute Autor, Buch, Zeitschrift und Titel. Anders verhält sich daß Attribut Jahr, daß durch die Attributmenge $\{Jahr, Zeitschrift\}$ dominiert ist. Die Ergebnisse für Ähnlichkeitsmaß α_2 zeigen für alle Attribute der Klasse P ausschliesslich Abhängigkeiten von den gleichnamigen Attributen der Klasse D. Darüberhinaus fällt auf, dass indirekte Abhängigkeiten für beide Ähnlichketsmaße keine Rolle spielen. Die weiteren Experimente basieren auf der durch das Ähnlichkeitsmaß α_2 erzeugten Abhängigkeitsstruktur. Auf Basis dieser Struktur S und des Ähnlichkeitsmaßes α_2 wurden die Parameter des Modelles geschätzt und in den weiteren Experimenten angewendet.

6.3.2 Das Verhalten einer Markov-Kette bei verschiedenen Strategien

Auf Basis des PRMs und seiner Parameter wurden anschließend Experimente durchgeführt, mit dem Ziel das Verhalten verschiedener Strategien des in Abschnitt 5.4.4 beschriebenen Algorithmus zu untersuchen. Die Experimente wurden auf einer Datenmenge mit 10 Publikationen und maximal 5 Zitationen pro Publikation durchgeführt. Für diese Experimente wird eine Markov-Kette 1000 mal iteriert. In jedem Iterationsschritt wird der Jacard-Wert des Zustandes, in dem sich die Markov-Kette befindet, berechnet. Der Startzustand der Markov-Kette wird bei allen Versuchen zufällig aus der Menge aller möglichen Zustände gewählt.

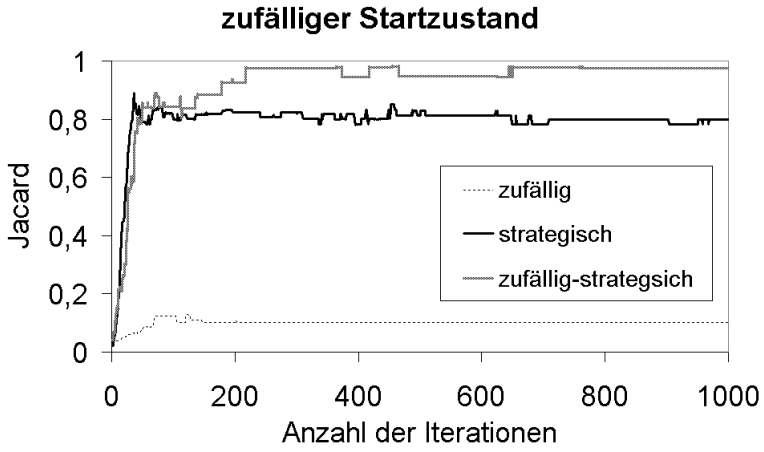

Abbildung 6.2: Vergleich unterschiedlicher Strategien für den MCMC-ALgorithmus

In Abbildung 6.2 ist deutlich zu erkennen, daß strategische und zufällig-strategische Auswahl deutlich besser abschneiden als die zufällige Auswahl. Für die ersten 200 Iterationsschritte schneidet die strategische Auswahl besser als die zufällig-strategische Auswahl ab. Danach ist das Verhalten beider Strategien unterschiedlich. Während die strategische Auswahl sich auf einem suboptimalen Niveau einpendelt, tendiert die zufällig-strategische Auswahl zum optimalen Zustand.

Ähnliche Ergebnisse können beobachtet werden, wenn statt eines zufälligen ein fester Startzustand gewählt wird. Bei einem ,,1:1,,-Startzustand, verweist jede Zitation auf eine eigene Publikation. Wählt man einen ,,n:1,,-Startzustand verweisen alle Zitationen auf

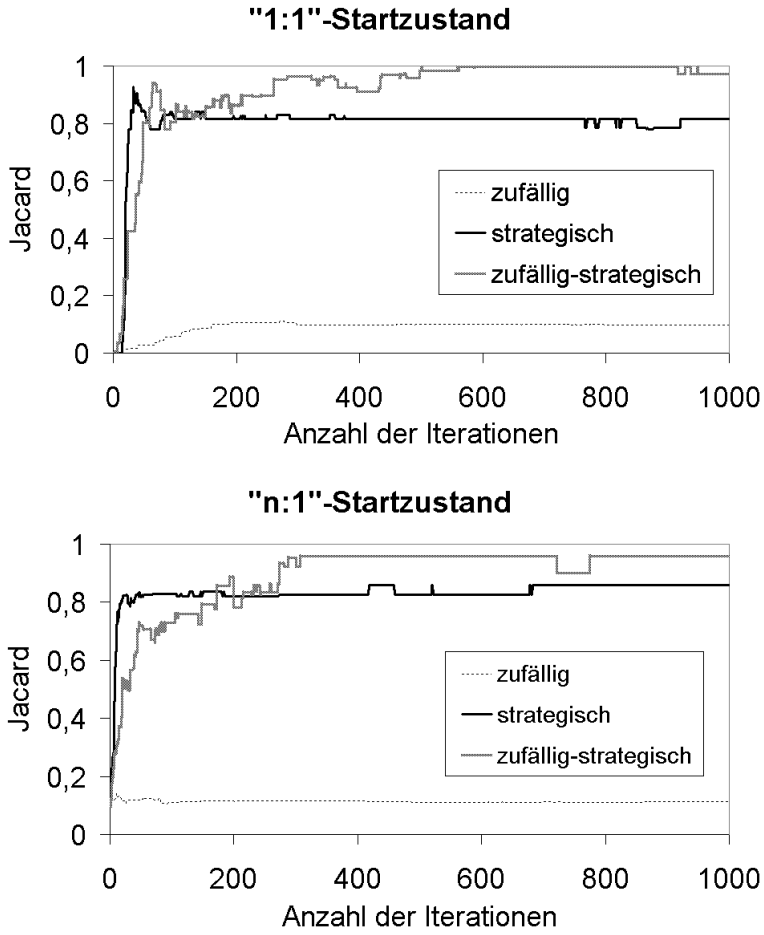

Abbildung 6.3: Vergleich unterschiedlichen Strategien bei der Auswahl des MCMC-Schrittes beim MH-Sampling für einen 1:1 und einen n:1-Startzustand

dieselbe Publikation. Abbildung 6.3 zeigt das Verhalten der drei Strategien für diese beiden Startzustände.

6.3.3 Simulation mehrerer Markov-Ketten

Weitere Experimenten untersuchen den Einfluß der Wahl des resultierenden Zustandes bei der Simulation mehrerer Markov-Ketten. Die Experimente wurden bei paralleler Simulation von zwei und drei Markov-Ketten durchgeführt. Bei der Simulation von zwei Markov-Ketten startete eine Kette im ,,1:1,,-Zustand, die andere Kette im ,,n:1,,-Zustand. Bei der

Simulation von drei Markov-Ketten startete eine Kette im ,,1:1,,-Zustand, eine zweite Kette im ,,n:1,,-Zustand und eine dritte Kette in einem zufälligen Zustand. Die Auswahl des nächsten Zustandes wurde ij allen Fällen zufällig-strategisch durchgeführt. In Abbildung

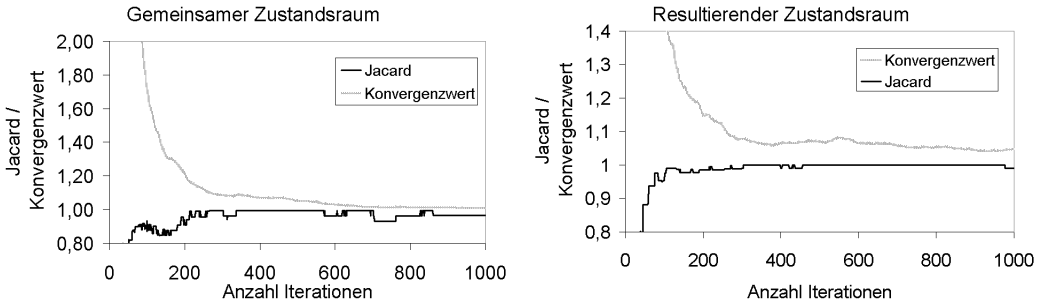

Abbildung 6.4: Ermittlung des resultierenden Zustandes bei zwei Ketten.

6.4 ist das Konvergenzverhalten bei der Simulation von zwei Markov-Ketten sowie die durchschnittliche Qualität des zu jedem Zeitpunkt der Simulation resultierenden Zustand dargestellt. Das linke Diagramm zeigt den durchschnittlichen Jacard-Wert, wenn eine Zustandsverteilung über die Zustände der einzelnen Markov-Ketten ermittelt wird. Es ist zu erkennen, daß eine Konvergenz der Markov-Ketten nicht mit einer Konvergenz des Jacard-Wertes einhergeht. Das rechte Diagramm zeigt den durchschnittlichen Jacard-Wert, wenn eine Zustandsverteilung über den resultierenden Zuständen der Markov-Ketten ermittelt wird. Man kann erkennen, daß die Kurve, die Qualität des resultierenden Zustandes wiederspiegelt gleichmässig auf hohem Niveau verläuft.

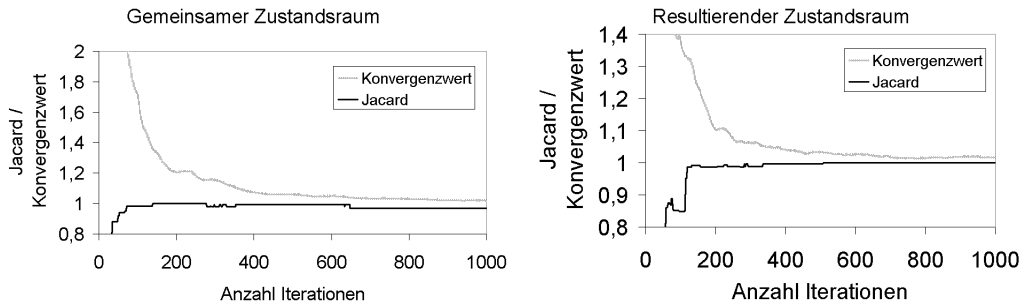

Abbildung 6.5: Ermittlung des resultierenden Zustandes bei drei Ketten.

Abbildung 6.5 zeigt das Konvergenzverhalten und die Qualität bei der Simulation von drei Markov-Ketten. Der Verlauf beider Jacard-Kurven ist ähnlich.

6.3.4 Generierung eines Zitationsgraphen

Bei den Experimenten zur Generierung eines Zitationsgraphen wurde zunächst der Einfluß des Konvergenz-Schwellwertes Zur Evaluierung des Verfahrens wurden die Experimente zur Generierung eines Zitationsgraphen ebenfalls mit einem Baseline-Verfahren durchgeführt, daß auf der Ähnlichkeit zweier Zeichenketten beruht. Dieses vergleicht die Datenpunkte paarweise und gruppiert diese, wenn der Vergleichswert oberhalb eines Schwellwertes liegt. Die Ähnlichkeit basiert auf der Schnittmenge von Wörtern, die in beiden Datenpunkten vorhanden sind. Auf den Einsatz eines standardisierten Ähnlichkeitsmasses wie dem Edit-Abstand wurde aufgrund des hohen Aufwandes, den der zeichenbasierte Vergleich mit sich bringt, bewusst verzichtet.

Algorithm 4 Baseline-Verfahren zur Generierung eines Zitationsgrahen

Require: *epsilon*
  $i \leftarrow AnzahlallerDatenpunkte$
  for $d_j = d_1$ to d_i **do**
    for $d_k = d_1$ to d_{i-1} **do**
      $m \leftarrow$ Anzahl der Wörter in d_j
      $n \leftarrow$ Anzahl der Wörter in d_k
      $o \leftarrow$ Menge gemeinsamer Wörter in d_j und d_k
      if $\frac{m}{o} + \frac{n}{o} \geq 2epsilon$ **then**
        Gruppiere d_i und d_j
      end if
    end for
  end for

Die Qualität des Baseline-Verfahrens ist direkt on der Wahl des Schwellwertes abhängig. Zur Bestimmung des optimalen Schwellwertes wurden Experimente auf einer Menge von 10 Publikationen und je 5 Zitationen durchgeführt. Experimente ergaben, daß dieser bei ca. 0.66 liegt, welches in Abbildung 6.6 dargestellt ist. Der durchschnittliche Jacard-Wert für eine Schwellwert von 0.66 liegt bei 0.8. Dieser liegt deutlich unter dem des MCMC-Algorithmus.

Ein wichtiges Kriterium für ein Verfahren ist die Skalierbarkeit. Um dieses zu messen, wurden die Experimente zur Generierung des Zitationsgraphen auf einer Daten-

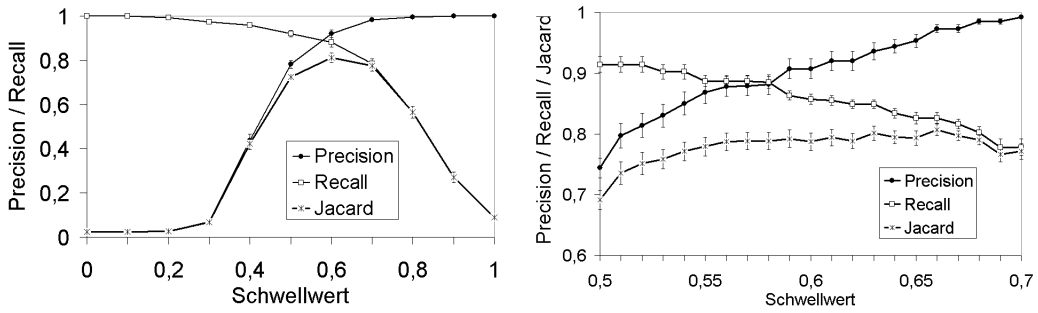

Abbildung 6.6: Generierung des optimalen Schwelwertes für das Baseline-Verfahren

menge durchgeführt, die sukzessive erhöht wurde. Um den Aufwand, den grosse Datenmengen mit sich führen, zu reduzieren, wird eine Divide-and-Conquer Srategie angwendet. Dazu wird die Menge der Datenpunkte zunächst mit Hilfe des Baseline-Verfahrens in einzelne Buckets aufgeteilt. Anschliessend wird jeder Bucket fuer sich optimiert. Sind alle Buckets optimiert, werden diese zum Zitationsgraphen zusammengesetzt. Der MCMC-ALgorithmus wurde dabei durch die Simulation von 3 Markov-Ketten und einem Konvergenz-Schwellwert von 1.2 durchgeführt.

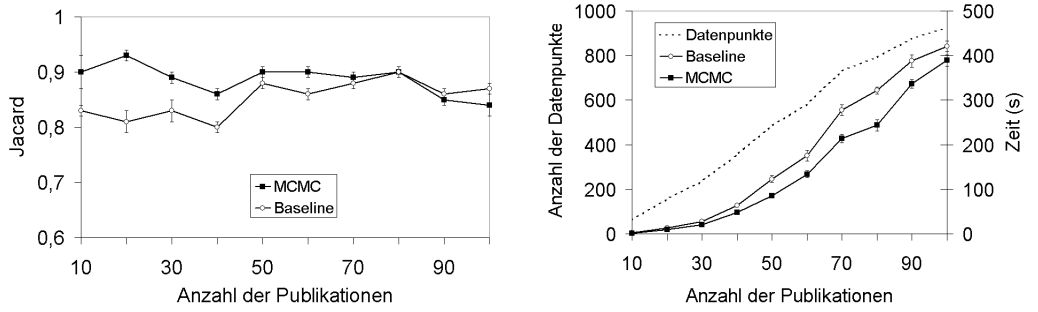

Abbildung 6.7: Skalierbarkeit beider Verfahren in Bezug auf Qualität und Kosten

Die Ergebnisse dieser Experimente sind in Abbildung 6.7 zu sehen. Man erkennt, daß das MCMC-Verfahren bis zu einer Anzahl von 80 Publikationen eine bessere oder gleiche Qualität wie das Baseline-Verfahren aufweist. Ab einer Zahl von 90 Publikationen sinkt die Qualität des MCMC-ALgorithmus unter die des Baseline-Verfahrens. Bezüglich der Kosten weisen beide Verfahren einen nahezu identischen Aufwand auf. Dieser verhält sich linear zur Anzahl der Datenpunkte.

6.4 Zusammenfassung

Die durchgeführten Experimente zeigen, daß die Wahl des Priors α bei der Parameter-Schätzung einen starken Einfluß auf die generierte Abhängigkeitsstruktur hat, wobei die Wahl eines Priors mit einem niedrigen Wert, der durch eine Ähnlichkeitsfunktion charakterisiert ist, intuitiv sinnvollere Strukturen erzeugt, als ein Prior mit einem hohen Wert.Durch die Vorauswahl günstiger Zustände erreicht die Simulation von Markov-Ketten deutlich bessere Werte als bei einer zufälligen Auswahl des Folgezustandes. Auffällig ist, daß ein Greedy-Ansatz, der immer den bestbewertetsten Folgezustand preferiert in einem suboptimalen Zustandsraum verharrt.Weitere Experimente zeigten, daß bei der Simulation mehrerer Markov-Ketten der Einsatz einer Funktion, die alle Zustände in einen eigenen Zustandsraum abbildet, zu besseren Ergebnissen führt, als die Modellierung aller Markov-Ketten in einem gemeinsamen Zustandsraum. Bei Experimenten zur Generierung eines Zitationsgrapen wurden bei der Simulation von drei parallel laufenden Markov-Ketten bessere Ergebnisse erzielt, als bei der Simulation von zwei parallelen Matrkov-Ketten. Experimente zur Skalierbarkeit des Verfahrens zeigten zum einen, daß der Aufwand durch Einsatz einer Divide-and-Conquer Strategie gesenkt werden kann. Der MCMC-ALgorithmus erzielt bei Einsatz dieser Strategie für kleine Datenmengen bessere Ergebnisse als ein Baseline-Verfahren, mit Zunahme der Datenmenge sinkt die Qualität des Verfahrens unter die des Baseline-Verfahrens. Die Kosten, des Algorithmus steigen dabei linear zur Datenmenge, liegen aber unter denen eines Baseline-Verfahrens.

7 Zusammenfassung und Ausblick

In dieser Diplomarbeit wurde die Problemstellung der Entity-Resolution für Zitationen untersaucht. Dazu wurde ein Probabilitisches Relationales Modell entworfen, das eine Erweiterung des klassischen Ansatzes darstellt. Weiterhin wurde ein auf diesem Modell operierender MCMC-ALgorithmus vorgestellt, der durch die wechselseitige Anwendung von Gibbs-Sampler und Metropolis-Algorithmus gekennzeichnet war. Die Ergebnisse, die durch Experimente auf dem Cora-Datensatz erzielt wurden, zeigen dass das Verfahren für kleinere Datenmengen besser als ein Baseline-Verfahren abschneidet. Der Einsatz einer Divide-and-Conquer-Strategie führt dazu, daß der Aufwand linear mit der Datenmenge steigt und für grössere Datenmengen unterhalb dem des Baseline-Verfahrens liegt.

Das vorgestellte Modell bietet viel Spielraum für weitere Untersuchungen. Zum einen könnte der Einfluß einer grösseren Attributmenge auf die Qualität des Verfahrens untersucht werden. Dies könnte in Verbindung mit einer Verfeinerung des Modells einhgergehen wobei jeder Publikationstyp durch eine eigene Klasse modelliert wird. Dieser Ansatz bietet die Möglichkeit, komplexere Abhängigkeiten zu lernen. Darüberhinaus wäre eine andere Handhabung des Wertebereichs interessant, diesen nicht auf Wortfolgen, sondern auf einzelnen Wörtern zu definieren und zu untersuchen, ob sich dadurch andere, komplexere Abhängigkeitsstruktur generieren lassen. Eine interessante Erweiterung des Modells stellt den Einbeziehung eines Themenbezugs dar. Neben Experimenten auf dem Cora-Datensatz sind Versuche auf anderen Datensätzen wie arXiv und CiteSeer möglich.

Literaturverzeichnis

[1] American Psychological Association. Publications manual : Apa, 1994.

[2] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

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

[4] H. L. Dunn. Record linkage. *American Journal of Public Health*, 36:1412–141, 1946.

[5] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[6] International Organization for Standardization. Documentation - bibliographic references - content, form and structure, 1987.

[7] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.

[8] E. Garfield. Citation indexes for science: A new dimension in documentation through association of ideas. *Science*, 122(3159):108–111, July 1955.

[9] A. Gelman. *Markov Chain Monte Carlo in Practice*, chapter Inference and monitoring convergence, pages 131–143. Chapman and Hall, London, 1996.

[10] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell*, 6:721–741, 1984.

[11] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of link structure, 2002.

[12] J. Gibaldi. Mla style manual and guide to scholarly publishing, 1999.

[13] D. Heckerman. A tutorial on learning with bayesian networks, 1995.

[14] C. Iverson. Ama manual of style : a guide for authors and editors, 2007.

[15] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *AAAI/IAAI*, pages 580–587, 1998.

[16] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22, New York, NY, USA, 1999. ACM Press.

[17] S. Lawrence, K. Bollacker, and C. L. Giles. Autonomous citation matching. In Oren Etzioni, editor, *Proceedings of the Third International Conference on Autonomous Agents*, New York, 1999. ACM Press.

[18] K. F. Lorenzen. Das literaturverzeichnis in wissenschaftlichen arbeiten. Technical report, FH Hamburg, FB Bibliothek und Indormation, 1997.

[19] A. A. Markov. *Wahrscheinlichkeitsrechnung*. Teubner, 1912.

[20] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000. www.research.whizbang.com/data.

[21] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Knowledge Discovery and Data Mining*, pages 169–178, 2000.

[22] M. Meilă and D. Heckerman. An experimental comparison of model-based clustering methods. *Mach. Learn.*, 42(1-2):9–29, 2001.

[23] M. Meila. Comparing clusterings, 2000.

[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[25] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[26] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage of vital records. *Science*, 130:954–959, October 1959.

[27] L. Ngo and P. Haddawy. Answering queries from context-sensitive probabilistic knowledge bases. *Theoretical Computer Science*, 171(1–2):147–177, 1997.

[28] Commonwealth of Australia. Style manual for authors, editors and printers, 2002.

[29] Univ. of Chicago Press. The chicago manual of style, 2003.

[30] National Information Standards Organization. Bibliographic references. Technical report, National Information Standards Organization, 2005.

[31] H. Pasula, B. Marthi, B. Milch, S. J. Russell, and I. Shpitser. Identity uncertainty and citation matching, 2002.

[32] D. Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.

[33] S. Sarawagi, V. G. V. Vydiswaran, S. Srinivasan, and K. Bhudhia. Resolving citations in a paper repository. *SIGKDD Explorations*, 5(2):156–157, 2003.

[34] K. Turabian. A manual for writers of term papers, theses, and dissertations, 1996.

[35] Normenausschuss Bibliotheks und Dokumentationswesen (NABD) im DIN Deutsches Institut für Normung. Din 1505 teil2 : Gekürzte titelangaben: Zitierregeln, 1981.

A Danksagung

Mein Dank geht an alle, die auf Ihre Weise einen Teil zum Gelingen dieser Arbeit beigetragen haben.

Ich danke meinem Betreuer Dipl.-Inf Ulf Brefeld für seinen wertvollen fachlichen Rat und den immer geduldigen Beistand, die diese Arbeit erst ermöglichten. Mein Dank gilt auch Prof. Tobias Scheffer für die gute wissenschaftliche Betreuung während dieser Zeit. Ebenfalls möchte ich mich beim gesamten Team des KOBV bedanken, insbesondere bei Stefan Lohrum und Andres Imhof, für die vielen Ideen und interessanten Anregungen im Vorfeld dieser Arbeit.

Ein besonderer Dank geht an meine Lebensgefährtin Anja Lull, die mich mit vielen motivierenden Worten und Taten die letzten Jahre begleitet und unterstützt hat. Selbverständlich möchte ich mich auch bei meinen Eltern bedanken, für ihre langjährige Unterstützung während meiner gesamten Studienzeit. Ein herzlicher Dank gilt Renate Lull für die viele Zeit, die sie sich um unseren Sohn Jasper gekümmert hat.

B Selbständigkeitserklärung und Einverständniserklärung

Selbständigkeitserklärung
Ich erkläre hiermit, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Berlin, den

Einverständniserklärung
Ich erkläre hiermit mein Einverständnis, dass die vorliegende Arbeit in der Bibliothek des Instituts für Informatik der Humboldt-Universität zu Berlin ausgestellt werden darf.

Berlin, den