Datenzentrierte Bestimmung von Assoziationsregeln in parallelen Datenbankarchitekturen

Dissertation

zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

Dipl.-Inf. Thomas Legler
geboren am 2. April 1980 in Freiberg

Gutachter: Prof. Dr.-Ing. Wolfgang Lehner
Technische Universität Dresden
Fakultät Informatik, Institut für Systemarchitektur
Lehrstuhl für Datenbanken
01062 Dresden

Prof. Dr.-Ing. habil. Kai-Uwe Sattler
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
FG Datenbanken und Informationssysteme
Postfach 100 565
98684 Ilmenau

Tag der Verteidigung: 22. Juni 2009

Walldorf, den 24. Juli 2009

Kurzbeschreibung

Die folgende Arbeit befasst sich mit der Alltagstauglichkeit moderner Massendatenverarbeitung, insbesondere mit dem Problem der Assoziationsregelanalyse. Vorhandene Datenmengen wachsen stark an, aber deren Auswertung ist für ungeübte Anwender schwierig. Daher verzichten Unternehmen auf Informationen, welche prinzipiell vorhanden sind. Assoziationsregeln zeigen in diesen Daten Abhängigkeiten zwischen den Elementen eines Datenbestandes, beispielsweise zwischen verkauften Produkten. Diese Regeln können mit Interessantheitsmaßen versehen werden, welche dem Anwender das Erkennen wichtiger Zusammenhänge ermöglichen. Es werden Ansätze gezeigt, dem Nutzer die Auswertung der Daten zu erleichtern. Das betrifft sowohl die robuste Arbeitsweise der Verfahren als auch die einfache Auswertung der Regeln. Die vorgestellten Algorithmen passen sich dabei an die zu verarbeitenden Daten an, was sie von anderen Verfahren unterscheidet.

Assoziationsregelsuchen benötigen die Extraktion häufiger Kombinationen (EHK). Hierfür werden Möglichkeiten gezeigt, Lösungsansätze auf die Eigenschaften moderne System anzupassen. Als Ansatz werden Verfahren zur Berechnung der häufigsten N Kombinationen erläutert, welche anders als bekannte Ansätze leicht konfigurierbar sind. Moderne Systeme rechnen zudem oft verteilt. Diese Rechnerverbünde können große Datenmengen parallel verarbeiten, benötigen jedoch die Vereinigung lokaler Ergebnisse. Für verteilte Top-N-EHK auf realistischen Partitionierungen werden hierfür Ansätze mit verschiedenen Eigenschaften präsentiert.

Aus den häufigen Kombinationen werden Assoziationsregeln gebildet, deren Aufbereitung ebenfalls einfach durchführbar sein soll. In der Literatur wurden viele Maße vorgestellt. Je nach den Anforderungen entsprechen sie je einer subjektiven Bewertung, allerdings nicht zwingend der des Anwenders. Hierfür wird untersucht, wie mehrere Interessantheitsmaßen zu einem globalen Maß vereinigt werden können. Dies findet Regeln, welche mehrfach wichtig erschienen. Der Nutzer kann mit den Vorschlägen sein Suchziel eingrenzen. Ein zweiter Ansatz gruppiert Regeln. Dies erfolgt über die Häufigkeiten der Regelelemente, welche die Grundlage von Interessantheitsmaßen bilden. Die Regeln einer solchen Gruppe sind daher bezüglich vieler Interessantheitsmaßen ähnlich und können gemeinsam ausgewertet werden. Dies reduziert den manuellen Aufwand des Nutzers.

Diese Arbeit zeigt Möglichkeiten, Assoziationsregelsuchen auf einen breiten Benutzerkreis zu erweitern und neue Anwender zu erreichen. Die Assoziationsregelsuche wird dabei derart vereinfacht, dass sie statt als Spezialanwendung als leicht nutzbares Werkzeug zur Datenanalyse verwendet werden kann.

Danksagung

Zu allererst möchte ich meinem Betreuer Prof. Dr. Wolfgang Lehner danken. Ohne seine regelmäßigen Anregungen und Motivationen wäre diese Arbeit nicht möglich gewesen. Außerdem danke ich ihm für die vielen Einblicke in die Welt der Datenverarbeitung und Forschung, welche er mir in den vergangenen fünf Jahren ermöglicht hat. Dies umfasst nicht nur den Kontakt zu meinen jetzigen Arbeitskollegen und die Möglichkeit von Konferenzbesuchen. Vielmehr möchte ich ihm danken, dass er mir geholfen hat, meinen Weg aus dem Studium in die raue Arbeitswelt zu finden. Vielen Dank für diese Chance und für all die Dinge, die sie für mich getan haben! Ich bedanke mich ebenfalls bei Herr Prof. Dr. Sattler und Herr Prof. Dr. Hilbert für ihre Mühen mit dem Gutachten dieser Arbeit.

Des Weiteren möchte ich allen meinen Kollegen und (ehemaligen) Mitstreitern für die Gespräche, Ideen und Gedanken danken. Dies betrifft insbesondere Sven für die ersten Einblicke in die Welt der Wissenschaft, Rainer für die wahrscheinlich viel zu tiefen Einblicke und Maik als stetigem Begleiter. Außerdem bedanke ich mich bei Ines für die Hilfe im Kampf mit der Bürokratie.

Ich danke auch der SAP AG und der GWT TUD GmbH für die Möglichkeit, diese Arbeit verfassen zu können. Ganz besonders gilt dies für die Kollegen, die mich vor 3,5 Jahren aufgenommen haben: der Abteilung TREX. Danken möchte ich allen, aber ich möchte mich an dieser Stelle namentlich auf Roland Kurz und Franz Färber als Herz und Kopf der Abteilung beschränken. Danke für die Chancen, die Unterstützung bei deren Realisierung und den fairen Umgang miteinander.

Besonderer Dank gilt auch all denen, die mir bei der Fertigstellung der Arbeit geholfen haben: meinem Schatz Martina, meinen Eltern Reiner und Angela, Christian, Pascal (SV), Bernhard, Olga und Pascal (H). Sowie all denen, die mich stetig nach dem Stand der Arbeit befragt haben und ob ich nicht bald fertig sein müsste.

Vielen Dank an Alle!

Thomas Legler, 8. April 2009

Inhaltsverzeichnis

1. Einleitung

Alles Wissen besteht in einer sicheren und klaren Erkenntnis.
René Descartes

Der rasche Fortschritt moderner Informationstechnologie erlaubt die Erfassung und Speicherung enormer Datenmengen. So entstehen nach Brodie oder Witten gewaltige Datensammlungen, welche kaum überschaubar sind [Bro07; WMB99a]. Im Jahre 2006 wurden beispielsweise 10^8 Terabyte Daten erstellt oder repliziert [Bro07]. Das übertrifft die gesamten Daten der vorherigen 5.000 Jahre. Dieses Wachstum hält nach Schätzungen diverser Organisationen weiter an und verdreifacht sich alle drei Jahre. Der amerikanische Telekommunikationsdienstleister AT&T vermittelte im Jahre 2004 etwa zwei Billionen Anrufe und musste diese Verbindungen protokollieren und analysieren. Ähnliche Größenordnungen erreicht die amerikanische Handelskette Wal-Mart mit etwa zehn Millionen Transaktionen pro Tag [Bro07].

Betrachtet man diese Daten, wird deutlich, dass deren manuelle Aufbereitung eine unüberwindbare Hürde darstellt. Das in den Daten verborgene Wissen repräsentiert jedoch eine wichtige Informationsquelle der jeweiligen Institution. Verfahren zur Erlangung verschiedenster Analyseziele sind somit dringend erforderlich und werden unter dem Begriff Data Mining zusammengefasst. In [HK05a] wird der Begriff des Data Mining durch Han und Kamber als Prozess zur Erkennung von Datenmustern durch intelligente Methoden definiert. Praxisbezogener drückt sich Cabena [CH97] aus und definiert:

> Data Mining.. is the process of extracting previously unknown, valid, and actionable information from large databases and then using the information to make crucial business decisions. [1]

Diese Definition zeigt die Wichtigkeit klarer und sicherer Erkenntnisse im geschäftlichen Umfeld. Erst durch eine gründliche Untersuchung vorhandener Daten sollten wichtige Entscheidungen getroffen werden. In dieser Arbeit wird Data Mining als Prozess aufgefasst, mit dessen Hilfe aus vorhandenen Datenbeständen Informationen extrahiert werden. Diese sollen zu interessanten, bisher unbekannten Erkenntnissen führen.

[1] deutsch: *Data Mining ist der Prozess der Extraktion bisher unbekannter, gültiger und nutzbarer Informationen aus großen Datenbeständen und die Nutzung dieser Informationen, um kritische geschäftliche Entscheidungen zu treffen.*

Einen wichtigen Bereich des Data Mining stellt die sogenannte Assoziationsregelanalyse dar [AIS93; Zak99]. Diese versucht, in großen Datenbeständen Verbindungen zwischen verschiedenen Elementen, Wertausprägungen oder sonstigen Eigenschaften aufzudecken. Das können beispielsweise gemeinsam verkaufte Produkte oder sich beeinflussende Gensequenzen sein. Dabei müssen die Ausgangsdaten um unwichtige Anteile bereinigt und im Umfang reduziert werden. Nur damit können in einer Analyse interessante Informationen aus den Daten extrahiert und genutzt werden. Bei der rechnergestützten Datenaufbereitung müssen daher einerseits die wichtigen Informationen erhalten bleiben und andererseits die betrachtete Informationsmenge minimiert werden.

Die Bestimmung von Assoziationsregeln ist ein bekanntes und weitreichend untersuchtes Problem der Datenbankenforschung. Entsprechend ist eine Vielzahl von Verfahren für deren Berechnung bekannt. Der Fokus dieser Arbeit liegt auf der Untersuchung von Ansätzen zur einfachen Bedienung der Algorithmen, der automatischen Erkennung interessanter Regeln und Möglichkeiten zur effizienten Erfassung der gefundenen Informationen. Insbesondere sollen sich die verwendeten Verfahren möglichst automatisiert an die jeweils zu untersuchenden Daten anpassen, um deren Auswertung zu erleichtern. Dies betrifft sowohl Datencharakteristiken als auch Eigenschaften der jeweiligen Systemlandschaft.

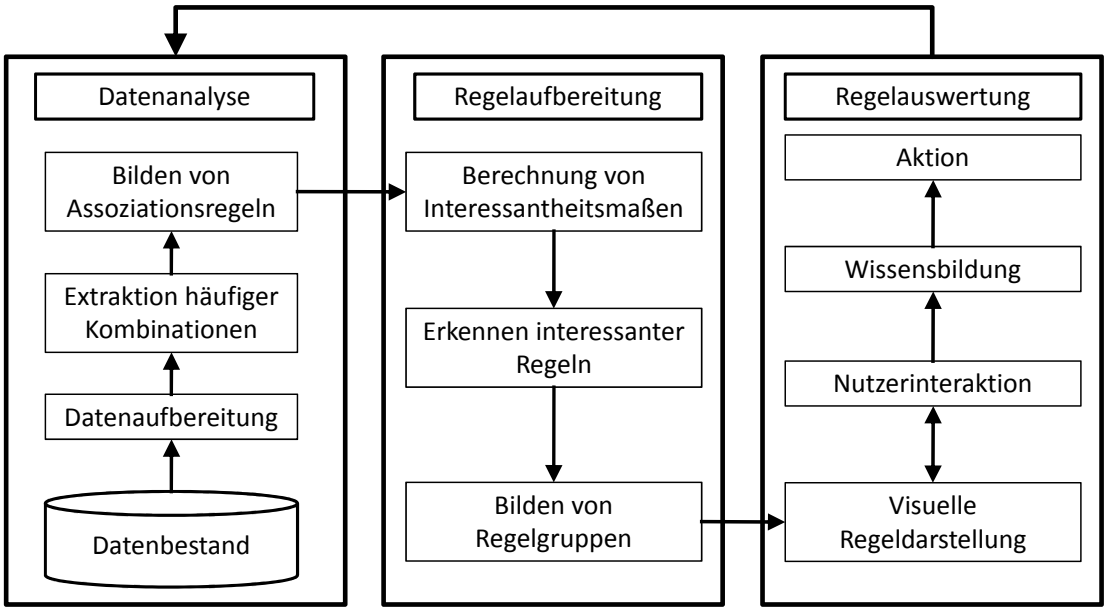


Abbildung 1.1.: Drei Schritte der Assoziationsregelsuche

Abbildung 1.1 zeigt den Ablauf einer Assoziationsregelsuche. Dieser besteht im Wesentlichen aus drei Arbeitsschritten: Datenanalyse, Regelaufbereitung und Regelauswertung. Während die Datenanalyse und Regelaufbereitung (teil-)automatisch erfolgen kön-

nen, werden zur Regelauswertung Nutzerinteraktionen benötigt. Diese Arbeit betrachtet hauptsächlich die automatisiert durchführbaren Schritte der Datenanalyse und Regelaufbereitung.

Die Datenanalyse nimmt bei einer Assoziationsregelanalyse den Großteil der Rechenleistung in Anspruch. Hierbei erfolgt die Verarbeitung umfangreicher Ausgangsdaten. Die nachfolgenden Schritte verarbeiten je ein deutlich geringeres Datenvolumen und sind somit im Bezug auf den erzeugten Berechnungsaufwand unkritisch. In der Fachliteratur vorgestellte Lösungen entsprechen den Anforderungen realistischer Szenarien nur bedingt, da oft erfahrene Spezialisten zur Durchführung der Analysen benötigt werden. Für eine allgemein verständliche Nutzung dieser Algorithmen durch eine breite Anwenderschicht muss die Bedienbarkeit der Verfahren verbessert und die Verarbeitung spezieller, praxisnaher Datencharakteristiken ermöglicht werden. Dies wurde bisher nur unzureichend untersucht und als Lösung auf eine jeweils nötige Vorverarbeitung der Daten verwiesen, um auftretende Unzulänglichkeiten zu vermeiden.

Im Kern der Datenanalyse wird eine Bestimmung von häufigen Elementkombinationen aus der Menge aller Elemente eines Datenbestandes durchgeführt. Die meisten Verfahren zur Assoziationsregelsuche nutzen unter anderem einen Parameter zur Festlegung einer Mindesthäufigkeit für die zu bestimmenden Kombinationen. Mit Hilfe dieses Parameters wird die Menge relevanter Kombinationen im Vergleich zu allen möglichen Kombinationen signifikant beschränkt und auf die häufigen, wahrscheinlich für den Anwender interessanten und relevanten Ergebnisse reduziert.

Für komplexe Szenarien ist die Wahl dieses Parameters ein schwieriges Problem, da dieser sehr stark von den zu untersuchenden Daten und deren untereinander bestehenden Beziehungen abhängt. Die Konfiguration einer gewünschten Suche wird für ungeübte Anwender ein schwieriger und langwieriger Prozess. Dieser kann bei falsch gewählten Werten auch um Größenordnungen mehr Kombinationen als erwartet und damit enorme Algorithmenlaufzeiten und Kosten erzeugen. Außerdem verdeckt eine zu große Menge an Informationen die für den Anwender wichtigen Erkenntnisse und erschwert deren Analyse. In der Praxis wird zur Lösung daher oft ein Regelkreislaufverfahren verwendet, um schrittweise sinnvolle Parameterbelegungen und verwendbare Ergebnismengen zu bestimmen.

Die Durchführung von Regelkreislaufverfahren benötigt versierte Nutzer mit Erfahrungen im Umgang mit dem jeweils vorhandenen Algorithmen, Daten und Rechnerarchitekturen. Entsprechend wird die Verwendung durch unerfahrene Nutzer verhindert oder zumindest erschwert. Nach der Einschätzung des Claremont Report on Database Research [AAB+08] muss in Zukunft genau diesen neuen Nutzergruppen ein Zugang ermöglicht werden, da ein zu geringer Anwenderkreis nicht in der Lage ist, alle wertvollen Erkenntnisse in den vorhandenen Daten von Unternehmen zu finden. In Gesprächen mit Kunden der SAP AG wurde ebenfalls deutlich, dass eine breitflächige Nutzung dieser Technologien für Unternehmen attraktiv ist. Um dies zu erreichen, sollen daher in

dieser Arbeit vorhandene Parameter der Assoziationsregelanalyse reduziert oder durch einfache Top-N-Strategien ersetzt werden. Dabei werden Anfragen gestellt, bei denen der Umfang der gewünschten Ergebnismenge auf die häufigsten N Ergebnisse begrenzt wird. Solche Strategien wurden bereits in einer Vielzahl theoretischer Arbeiten untersucht [Con01; HWLT02; He05; MTW05; TWS04]. Sie sollen an realistische Anforderungen und Besonderheiten praxisnaher Verkaufsanalysen angepasst werden. Diese Erweiterungen betrachten die Daten und deren Charakteristiken als vorgegeben und passen die verwendeten Verfahren an die zu untersuchenden Daten an. Umgekehrt setzen bisherige Ansätze jeweils für die Verfahren günstige Datencharakteristiken voraus [Zak99].

Aus Sicht der Datenverarbeitung ist eine Vorverarbeitung und Aufbereitung von den zu analysierenden Daten meist unerwünscht. Dies bedeutet in den meisten Fällen eine zumindest teilweise Replikation der betrachteten Daten und damit eine deutliche Kostensteigerung, die Gefahr von Dateninkonsistenzen oder mögliche Verletzungen von Vertraulichkeiten. Für praktische Anwendungen ist es somit sinnvoll, eine potenziell benötigte Vorverarbeitung im Rahmen der Assoziationsregelsuche durchzuführen und Analysen direkt auf den bereits vorhandenen Ausgangsdaten zu ermöglichen. Die Aufbereitung der Daten erfolgt als Teil der Berechnung.

Im Jahre 1997 postulierte Gray [GCB+97] Systemanforderungen, um effiziente Datenanalysen durchführen zu können. Diese Anforderungen lauten wie folgt:

1. Datenzugriff soll möglichst direkt sein,

2. Datenstrukturen sollen einfach und im schnellen Hauptspeicher sein,

3. Texte sollen auf ganzzahlige Werte abgebildet sein,

4. Datenbestände sollen zur effizienten Verarbeitung verteilt vorliegen.

Moderne Rechnersysteme, welche eine Massendatenverarbeitung auf parallelen Clusterarchitekturen und ausreichend Hauptspeicher pro Rechenknoten durchführen, können diese Forderungen erfüllen. Speziell angepasste Datenbanksysteme mit Fokus auf lesende Datenzugriffe, wie der SAP BW Accelerator [LLR06], Blink [RSQ+08], C-Store [SAB+05] oder Vertica [Ver07], weisen zusätzlich oft einfache Datenstrukturen und eine interne Datenrepräsentation über ganzzahlige Abbildungen auf. Ist ein direkter, systemnaher Zugriff auf solche Daten möglich, stellen diese Datenbankarchitekturen ein geeignetes Werkzeug zur Analyse großer Datenbestände dar.

Ein grundlegender Gedanke dieser Konzepte ist auch die verteilte Datenverarbeitung. Parallele Berechnungen sind hierbei oft schwieriger zu lösen als unverteilte äquivalente Anfragen. Die Verwaltung einer parallelen Berechnung erzeugt Kosten und kann eine erhöhte Rechenleistung durch mehrere Knoten wieder neutralisieren. Insbesondere komplexe Datenanalysen wie die Assoziationsregelsuche benötigen oft Informationen über die

Gesamtdaten, um Besonderheiten zu entdecken. Im Falle einer verteilten Verarbeitung sind viele Informationen jedoch nur von lokalen Datenpartitionen verfügbar. Eine verteilte Berechnung kann damit erschwert werden, da viele Daten kommuniziert werden müssen.

Diese Arbeit soll eine Anwendung von Assoziationsregelsuchverfahren und deren Auswertung im Bezug auf die Anpassungsfähigkeit und Nutzbarkeit untersuchen. Dafür sollen eine möglichst einfache Verwendung der Verfahren durch eine Vielzahl von Anwendern erreicht werden, welche die Besonderheiten moderner Systemarchitekturen beachtet und nutzt. Insbesondere sollen verteilte Datenbestände untersucht werden können und die Berechnung der Assoziationsregeln soll vollständig im Hauptspeicher erfolgen.

Zudem werden Anwender ohne entsprechende Fachkenntnisse in die Lage versetzt, Assoziationsregeln zu bestimmen und auszuwerten. Dies ist mit vielen bekannten Verfahren durch schwierig zu bestimmende Parameter und schwer vorhersagbare Laufzeiten nur unzureichend möglich. Die Kenntnisse über die zugrunde liegende Systeme, Algorithmen und Parameter sollen dabei dem Nutzer weitestgehend transparent bleiben. Selbst Besonderheiten der Daten sollen von dem Anwender nicht zwingend beachtet werden müssen, um sinnvolle Ergebnisse mit vertretbarem Aufwand zu erreichen. Das endgültige Ziel dieser Arbeit liegt also weniger in der besonders effizienten Ausführung oder der Einführung neuer technischer Kennzahlen zur Bewertung der Regeln. Vielmehr soll die grundlegende Extraktion und Auswertung von Regeln für jedermann ermöglicht werden. Das ist für viele Szenarien bereits ausreichend mächtig, wird bisher aber durch die schwierige Anwendung der Verfahren verhindert.

Aufbau der Arbeit

Diese Arbeit ist in sechs Kapitel unterteilt. Hierbei folgt nach der Einleitung eine Einführung in die Methoden, Algorithmen und Probleme der Assoziationsregelsuche. Außerdem werden die in dieser Arbeit adressierten Szenarien, sowie die dabei entstehenden Probleme erläutert. Der Aufbau des ersten Kapitels orientiert sich an der Abbildung 1.1 und widmet den drei Arbeitsschritten je einen Abschnitt.

Die beiden Schritte Datenanalyse und Regelaufbereitung werden anschließend in je einem Kapitel näher untersucht. Dabei werden Probleme aufgezeigt, welche bei der Vereinfachung der Anwendung bekannter Verfahren und der Verarbeitung praxisnaher Daten entstehen und deren Lösungen vorgestellt.

Die Betrachtung der Datenanalyse erfolgt unterteilt in Lösungen zu Top-N-Verfahren, deren verteilter Berechnung, sowie mögliche Vereinfachungen der Ergebnismengen. Der Schritt der Regelaufbereitung enthält Ansätze zur automatischen Bewertung von Asso-

ziationsregeln und Möglichkeiten zur Gruppierung des Ergebnisses. Die Auswertung der beiden Kapitel erfolgt jeweils im vorletzten Abschnitt vor der Zusammenfassung. Dies erleichtert den Überblick und den Bezug der Auswertung zu den dargestellten Lösungen.

In Kapitel 5 erfolgt die Darstellung des im Rahmen der Arbeit erstellten Prototyps auf der Basis des hauptspeicherbasierten Datenverarbeitungssystems SAP BW Accelerator der SAP AG. Dabei werden sowohl das Konzept des SAP BW Accelerator als auch Details zur Implementierung der in dieser Arbeit gezeigten Verfahren vorgestellt. Zum Abschluss folgt im Kapitel 6 ein zusammenfassender Überblick der Arbeit.

2. Grundlagen

Ein Szenario für die Assoziationsregelanalyse ist die Warenkorbanalyse, welche in dieser Arbeit als primäres Beispiel dienen soll. Hierbei werden die Daten in Warenkörbe oder Transaktionen unterteilt. Übertragen auf reale Anwendungen kann ein solcher Warenkorb beispielsweise eine Bestellung oder auch eine Menge von Eigenschaften eines Produktes sein. Das Ziel der Warenkorbanalyse ist das Finden von Korrelationen zwischen den Einträgen der Warenkörbe. Das Konzept ist auch auf weitere Szenarien, wie beispielsweise die Analyse von Datenströmen mit Protokollinformationen oder Suche interessanter Gensequenzen anwendbar. Dabei können jeweils optimierte Verfahren deutliche Effizienzvorteile bieten [MM02; PCT+03; MCSW06; CTX+04]. In dieser Arbeit wird hauptsächlich die Assoziationsregelsuche in Verkaufsdaten untersucht, wodurch spezialisierte Verfahren anderer Anwendungsgebiete wenig relevant sind.

Dieses Kapitel bietet einen Überblick über bestehende Verfahren und Erkenntnisse zur Assoziationsregelsuche. Hierfür wird die verwendete Notation vorgestellt und anschließend Algorithmen und Verfahren erläutert, welche die Grundlagen zu den Lösungen dieser Arbeit bilden. Zur besseren Übersicht erfolgt jeweils eine Unterteilung des Kapitels in die drei Schritte der Assoziationsregelsuche: Datenanalyse, Regelaufbereitung und Regelauswertung. Zum Abschluss dieses Kapitels werden die Anforderungen praxisnaher Szenarien erläutert und deren Realisierbarkeit gegen die Fähigkeiten bekannter Verfahren abgegrenzt.

2.1. Notation

Zur Definition der Terminologie dieser Arbeit werden in Anlehnung an die Bezeichnungen von Agrawal [AIS93] im folgenden Abschnitt die Grundbegriffe der Assoziationsregelsuche erläutert.

2.1.1. Assoziationsregeln

Assoziationsregeln erleichtern die Darstellung von Zusammenhängen zwischen Kombinationen von Wertausprägungen und bieten eine klare Aussage, leichte Interpretierbarkeit

und Verständlichkeit. Eine Assoziationsregel ist eine Aussage der Form $A \Rightarrow C$, wobei A die Bedingung (engl. antecedent) und C (engl. conclusion) die Schlussfolgerung einer Regel darstellen. A und C bestehen jeweils aus Elementen (engl. items). Diese können z.B. Einträge eines Einkaufsbeleges oder Eigenschaften eines Produktes sein. A und C werden in dieser Arbeit als Kombinationen (engl. itemsets) bezeichnet und repräsentieren eine nichtleere Menge von Elementen.

Im Folgenden soll I die Menge aller Kombinationen darstellen, welche zur Berechnung von Assoziationsregeln benötigt werden. Oft wird diese Menge auf Kombinationen eingeschränkt, welche bestimmten Bedingungen genügen. Für den Bereich der Assoziationsregelsuchen wird meist eine minimal nötige Häufigkeit festgelegt und die entsprechenden Kombinationen als häufige Kombinationen bezeichnet. Für eine Assoziationsregel $A \Rightarrow C$ gilt $A \in I$, $C \in I$, $(A \cup C) \in I$ und $A \cap C = \emptyset$. Eine Regel $Y_1 : A_1 \Rightarrow C$ wird als verallgemeinerte Regel oder Generalisierung von $Y_2 : A_2 \Rightarrow C$ bezeichnet, falls durch das Entfernen von Elementen der Bedingung A_2 die Regel Y_1 entsteht. Y_2 ist entsprechend die spezialisierte Form von Y_1. Wird eine Regel $Y_2 : \{Bier, Wein\} \Rightarrow \{Windeln\}$ betrachtet, wäre eine Regel $Y_1 : \{Bier\} \Rightarrow \{Windeln\}$ deren Generalisierung.

Aus jeder Kombination $X \in I$ mit mindestens zwei Elementen können Assoziationsregeln erzeugt werden, indem für eine aus X erzeugte Regel $A \Rightarrow C$ gilt

$$C = X \setminus A \qquad \emptyset \neq A \subset X$$

Zur Bewertung von möglichen Regeln $A \Rightarrow C$ werden Interessantheitsmaße genutzt, welche die Aussagekraft und den Wert einer Regel erfassen sollen. Die Häufigkeit einer Regel sagt beispielsweise aus, wie oft die Kombination $A \cup C$ einer Regel $A \Rightarrow C$ in einem Datenbestand DB enthalten ist. Die Anzahl der Transaktionen in DB wird durch die Kardinalität $|DB|$ repräsentiert. Ein weiteres Maß stellt die Konfidenz (engl. confidence) einer Regel dar. Sie beschreibt, wie viele Transaktionen des Datenbestandes DB neben der Bedingung A ebenfalls die Schlussfolgerung C enthalten. Die Konfidenz repräsentiert somit die durch C bedingte Häufigkeit von A. Einzelne Interessantheitsmaße zeigen selten vollständig alle Aspekte einer Regel. Sie müssen daher oft kombiniert werden, um eine klare Aussage zu erhalten.

Aus der Menge alle Regel J soll die Teilmenge der nach einem bestimmten Maßstab am besten bewerteten R Regeln als Top-R-Regeln bezeichnet werden. Das heißt, eine Regel Y wird als Top-R-Regel nach einem bestimmten Maß bezeichnet, wenn nicht mehr als $(R-1)$ Regeln in J existieren, welche nach diesem Maß interessanter als Y sind.

Zur Auswahl interessanter Regeln aus der Menge aller erzeugten Regeln können die Interessantheitsmaße mit Schwellwerten versehen werden. Dabei werden nur die Regeln der dem Nutzer präsentierten Ergebnismenge hinzugefügt, welche den jeweiligen Schwellwert überschreiten. Für die Konfidenz wird hierfür die minimale Konfidenz *minConfidence* verwendet. Die Wahl dieser Schwellwerte ist dabei meist von den untersuchten Daten und

den spezifischen Anforderungen des Nutzers abhängig. Das erschwert deren Parametrisierung. Um das Verwerfen wichtiger Regeln zu vermeiden und eine sinnvolle Begrenzung der Regelmenge zu erreichen, sind meist Erfahrungswerte des Anwenders nötig.

Hinzu kommt ein aus der Signalverarbeitung bekanntes Problem: „One person's noise is another person's signal." (deutsch: *Was für den Einen ein Rauschen, ist für den Anderen ein Signal.*). Das bedeutet, der Wert eines Interessantheitsmaßes kann je nach Fokus und Analyseziel seines Betrachters verschieden sein. Das steht einer vollautomatischen, allgemeinen Auswahl „interessanter" Regeln entgegen. Im Rahmen dieser Arbeit sollen Möglichkeiten untersucht werden, welche eine effiziente Erfassung und Auswahl der Regeln durch den Nutzer ermöglichen.

2.1.2. Häufige Kombinationen

Assoziationsregeln werden aus Kombinationen von Elementen und deren Häufigkeit gebildet. Die Verbindung von Elementen zu Kombinationen erfolgt durch die gemeinsame Zugehörigkeit zu einer Gruppe. Das kann die Verbindung zu der Zeile einer Relation oder zu einem Warenkorb sein. Eine solche Menge von Elementen wird definiert als $L = \{e_1, e_2, \ldots, e_n\}$, wobei $n = |L|$ und e ein in DB auftretendes Element bezeichnet. Angewandt auf die Warenkorbanalyse entspricht L einer Transaktion der Transaktionsmenge DB. Die relative Häufigkeit für jedes $X \subseteq L, \emptyset \neq X \in DB$ wird mit $support(X, DB)$ oder zur besseren Lesbarkeit als $support(X)$ definiert. Hierbei ist relevant, ob X in einer Zeile L von DB enthalten ist. Mehrfaches Auftreten von X in L wird einfach gewertet. Die relative Häufigkeit von X im Verhältnis zu DB ist wie folgt definiert:

$$support(X, DB) = \frac{|\{L \in DB | X \subseteq L\}|}{|DB|}$$

Zur Eingrenzung der möglichen X wird ein Schwellwert $minSupport$ für die Häufigkeit von X eingeführt. Jedes X mit $support(X) \geq minSupport$, wird als *häufige Kombination* bezeichnet. Die Menge aller häufigen Kombinationen repräsentiert I. Die Häufigkeit einer Kombination kann äquivalent durch die absolute Häufigkeit von X in $|DB|$ angegeben werden. Zur Unterscheidung wird in dieser Arbeit der Ausdruck $minSupport$ für relative Angaben benutzt, $minSupportAbs$ bezeichnet eine absolute Mindesthäufigkeit. Äquivalent wird für die absolute Häufigkeit von X der Ausdruck $supportAbs(X)$ benutzt. Diese Darstellungen sind wie folgt in einander zu überführen:

$$\begin{aligned} minSupportAbs(DB) &= minSupport \cdot |DB| \\ supportAbs(X, DB) &= support(X) \cdot |DB| \end{aligned}$$

Die Länge einer Kombination X bezeichnet die Anzahl der Elemente in X. X wird als l-Kombination bezeichnet, wenn $|X| = l$.

Als Top-N-Kombinationen werden Kombinationen X bezeichnet, für welche in einem Datenbestand DB nicht mehr als $(N-1)$ Kombinationen existieren, deren Häufigkeit größer als die von X ist. Da lediglich Kombinationen mit mehr als einem Element Assoziationsregeln bilden können, wird in dieser Arbeit zusätzlich die Bedingung definiert, dass nicht mehr als $(N-1)$ häufigere Kombinationen als X in DB existieren dürfen, welche mindestens zwei Elemente enthalten.

Im Bezug auf häufige Kombinationen wird im Folgenden der Begriff der *nichtleeren Teilkombination von X* vereinfacht als *Teilkombination von X* verwendet. Für alle Kombinationen X gilt jederzeit $|X| > 0$. Die folgenden Abschnitte sollen nun Verfahren zur Bestimmung häufiger Kombinationen erläutern.

2.2. Lösungsansätze zur Bestimmung häufiger Kombinationen

Abbildung 2.1.: Erster Schritt der Assoziationsregelsuche

Dieser Abschnitt soll die Grundlagen für den ersten Teil des Assoziationsregelsuchprozesses vorstellen: die Datenanalyse. Ausgehend von den zu untersuchenden Daten wird dieser Prozess in Anlehnung an Abbildung 1.1 in drei Schritte unterteilt:

1. Datenaufbereitung,
2. Extraktion häufiger Kombinationen,
3. Regelbildung.

Die Datenaufbereitung umfasst beispielsweise die Selektion und Bereinigung der Ausgangsdaten oder die Eliminierung von Duplikaten. Außerdem werden in diesem Schritt die Informationen in eine zur Extraktion der häufigen Kombinationen günstige Datenstruktur überführt. Dieser Arbeitsschritt ist stark von den jeweilig nachfolgenden Algorithmen abhängig und wird in der Literatur meist wenig beachtet. Die zu untersuchenden Daten werden hier oft im jeweils passenden Format als Eingabe der Kombinationsextraktion erwartet. Erläuterungen zur Datenaufbereitung zu der in dieser Arbeit erstellten Lösung erfolgen in Abschnitt 5.4.1.

Wie bereits in Abschnitt 2.1.1 erläutert, werden Assoziationsregeln genutzt, um aus einem Datenbestand Beziehungen zwischen Elementen und die Art ihrer Korrelation zu extrahieren. Als Beispiel kann im Einzelhandel ein Datenbestand mit den Abverkäufen eines bestimmten Zeitraumes auf oft gemeinsam verkaufte Produkte untersucht werden. Damit kann beispielsweise erkannt werden, dass Kunden mit Äpfeln und Pfirsichen in 80 Prozent der Fälle auch Orangen kaufen. Oder das Kunden mit Windeln und Babypuder im Einkaufswagen eine bestimmte Biermarke bevorzugen. Alternativ können auch Korrelationen zwischen mehreren Attributen einer Tabelle extrahiert werden. Beispielsweise, dass Kunden einer Altersgruppe in einem spezifischen Gebiet überproportional häufig ein bestimmtes Produkt erwerben.

Um alle potenziell wichtigen Assoziationsregeln finden zu können, müssen sämtliche vorhandenen Kombinationen geprüft werden. Zusätzlich werden für diese Kombinationen die jeweiligen Häufigkeiten des Auftretens benötigt, um deren Wert beurteilen zu können.

Ohne Wissen und Einschränkungen über die gesuchten Ergebnisse wächst die Anzahl möglicher Kombinationen stark an. Das Ermitteln von Kombinationen und deren Häufigkeiten ist komplexitätstheoretisch problematisch, da sich die Anzahl möglicher Elementkombinationen I mit jedem Element verdoppelt. Sei E die Menge aller betrachteten Elemente und e ein einzelnes Element, ergeben sich folgende Zusammenhänge:

$$\begin{aligned} E &= e_1, e_2, e_3, \ldots, e_n \\ I &= \mathcal{P}(E) \setminus \{\emptyset\} = \{U | U \subseteq E\} \setminus \{\emptyset\} \\ |I| &= 2^{|E|} - 1 \end{aligned}$$

In der Praxis können Szenarien mehrere tausend Elemente enthalten und ermöglichen extrem viele Kombinationen. Dabei wird bereits ersichtlich, dass der Umfang einer solchen Berechnung maßgeblich von der Menge der betrachteten Kombinationen abhängt. Daraus folgt, dass diese Menge reduziert und auf wichtige Kombinationen beschränkt werden muss. Das Ziel ist, möglichst viele Produkte und deren Kombinationen von der Betrachtung auszuschließen, sollten diese für das Endergebnis wenig relevant sein. Dies kann direkt durch Vorgaben des Nutzers erreicht werden. Oder ein entsprechender Algorithmus muss selbstständig über die Relevanz einer Elementkombination entscheiden. Die manuelle Steuerung kann durch den Parameter *minSupport* erfolgen. Eine automatische Auswahl ist durch einen Vergleich mit bereits bekannten Ergebnissen möglich. Beispielsweise kann eine Kombination als wichtig erachtet werden, wenn sie im Vergleich zu den meisten anderen Kombinationen des Datenbestandes sehr häufig auftritt.

Den wichtigsten und meist rechenintensivsten Anteil des kompletten Assoziationsregelsuchprozesses stellt somit die Extraktion häufiger Kombinationen (*EHK*) dar. Die Ergebnisse dieses Prozesses können bereits wichtige Informationen für einen Nutzer darstellen und bilden zusätzlich die Grundlage zur Berechnung von Assoziationsregeln und deren Interessantheitsmaßen.

Im Wesentlichen bestehen für das Problem der EHK drei Lösungsansätze:

1. *APRIORI*-basierte Algorithmen, vorgestellt von Agrawal und Srikant [AS94],

2. Baumbasierte Verfahren wie der *FP-GROWTH*-Algorithmus, vorgestellt von Han, Pei und Yin [HPY00; PHM00] und

3. Tiefensuchebasierte Verfahren auf vertikalen Datenstrukturen, wie beispielsweise der von Zaki vorgestellte Algorithmus *ECLAT* [ZH99; ZPOL97].

Ein Überblick zu Suchstrategien, Algorithmen, deren Komplexität und die Problemstellungen bei der Suche nach häufigen Kombinationen wird von Zaki [Zak99] und Tan [TSK06a] vorgestellt. Einen erweiterten Fokus auf verteilte Datenbestände bieten Cheung und Xiao [CX98]. Des Weiteren sind ebenfalls Methoden bekannt, eine EHK mit Hilfe von weit verbreiteten Schnittstellen wie SQL zu ermöglichen [SS05a; SS05b]. Deren Realisierungen greifen jedoch meist nur indirekt, und damit langsam, auf die zu untersuchenden Daten zu. Für die betrachteten Szenarien in dieser Arbeit sind direkt Zugriffe möglich. Daher konzentrieren sich die folgenden Abschnitte auf die gebräuchlichsten Verfahren mit direktem Datenzugriff zur EHK (APRIORI, FP-GROWTH, ECLAT), sowie deren Optimierungen und Erweiterungen (BUC und PARTITION).

2.2.1. Apriori

Der APRIORI-Algorithmus ist einer der bekanntesten Algorithmen zur Generierung von häufigen Kombinationen und wurde erstmals von Agrawal und Srikant [AS94] vorgestellt. Seitdem dient er in seinen verschiedenen Implementierungen in einer Vielzahl von Arbeiten als Referenz für neue Verfahren zur Suche häufiger Kombinationen, um deren Leistungsfähigkeit zu bewerten (beispielsweise bei Borgelt [Bor03]).

Von Agrawal werden zwei wichtige Eigenschaften häufiger Kombinationen übernommen, welche in dieser oder abgewandelter Form die Grundlage für viele weitere Algorithmen dieses Problembereiches bilden.

Eigenschaft 2.1 (Antimonotonie)**:** Eine Teilkombination einer Kombination ist mindestens so häufig wie die Kombination selbst.

Eigenschaft 2.2 (Downward-Closure-Eigenschaft)**:** Aus der Eigenschaft 2.1 folgt, dass alle Teilkombinationen einer häufigen Kombination ebenfalls häufig sind.

Aus der Eigenschaft 2.1 kann gefolgert werden, dass beim Ermitteln einer häufigen Kombination X alle Kombinationen $X' \in (\mathcal{P}(X) \setminus \{\emptyset\})$ ebenfalls Teil der Ergebnismenge sind.

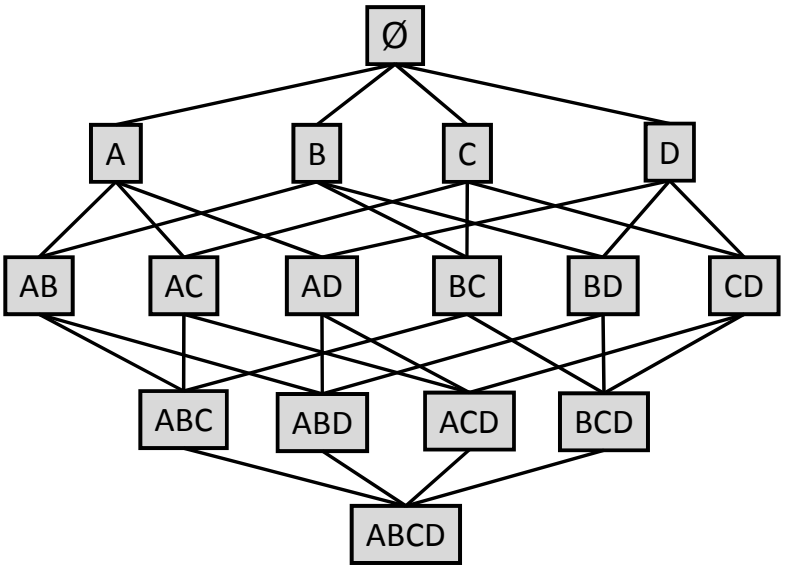


Abbildung 2.2.: Suchraumgitter für $I = \mathcal{P}(\{A,\ B,\ C,\ D\})$

Abbildung 2.2 zeigt den Suchraum, den APRIORI für die häufigen Elemente A, B, C und D durchläuft und analysiert. Dabei erfolgt die Verarbeitung jeweils pro Ebene, beginnend bei den Einzelelementen über 2-Kombinationen usw. bis hin zu Kombinationen mit der maximal möglichen Länge. Alle Kombinationen einer bestimmten Länge werden in einem gemeinsamen Arbeitsschritt von APRIORI untersucht. Der Algorithmus nutzt somit eine Breitensuche.

Der Algorithmus bestimmt im ersten Betrachten eines Datenbestandes alle häufigen Elemente bzw. 1-Kombinationen. Darauf aufbauend erfolgt in jedem weiteren Durchlauf k aus den im Durchlauf $(k-1)$ gefundenen $(k-1)$-Kombinationen die Generierung neue Ergebniskandidaten G_k. Deren Häufigkeiten in den Ausgangsdaten werden ermittelt und auf das Erreichen der Mindesthäufigkeit $minSupport$ überprüft. Wird diese erreicht, werden die Kandidaten dem Endergebnis hinzugefügt. Aus den akzeptierten k-Kombinationen können anschließend weitere $(k+1)$-Kandidaten erstellt und auf Häufigkeit geprüft werden. Sind keine neuen Kandidaten möglich, wird die Suche beendet.

Die Algorithmen 2.1 und 2.2 stellen diesen Ablauf dar. Die im Algorithmus 2.1 erläuterte Kandidatenerzeugung ermittelt häufige k-Kombinationskandidaten durch die Vereinigung von jeweils zwei $(k-1)$-Kombinationen. Unterscheiden sich zwei dieser $(k-1)$-Kandidaten in genau einem Element, bildet deren Vereinigung einen neuen k-Kandidaten. Im Verbundschritt erfolgt dieses Vorgehen für alle aus den $(k-1)$-Kombinationen ableitbaren k-Kombinationen. In der Filterphase werden diese Kandidaten mit Hilfe der Downward-Closure-Eigenschaft validiert. Nicht häufige Einträge werden aus Kandida-

Algorithmus 2.1 APRIORI-gen

Require: häufige $(k-1)$-Kombinationen G_{k-1}
1: $G_k = \emptyset$ /* Verbundphase */
2: **for all** $x \in G_{k-1}$ **do**
3: **for all** $y \in G_{k-1}$ **do**
4: **if** $|x \setminus y| == 1$ **and** $(x \cup y) \notin G_k$ **then**
5: $G_k = G_k \cup \{(x \cup y)\}$
6: **end if**
7: **end for**
8: **end for**
9: /* Filterphase */
10: **for all** $x \in G_k$ **do**
11: **for all** $y \in \mathcal{P}(x)$ mit $|y| = (k-1)$ **do**
12: **if** $y \notin G_{k-1}$ **then**
13: $G_k = G_k \setminus x$
14: **end if**
15: **end for**
16: **end for**
17: **return** G_k

tenmenge entfernt. Dies ist der Fall, wenn nicht alle ihre $(k-1)$-Teilkombinationen in den häufigen $(k-1)$-Kombinationen enthalten sind.

Durch den Algorithmus 2.2 werden die Häufigkeiten aller Ergebniskandidaten pro k bestimmt und diese bei Erreichen der Mindesthäufigkeit *minSupport* in die Ergebnismenge I übernommen. Der Algorithmus endet, sobald aus der Menge der häufigen $(k-1)$-Kombinationen keine weiteren k-Kombinationen gebildet werden können. Nach Han [HPY00] weist der APRIORI-Algorithmus folgende Nachteile auf:

1. Die Durchsuchung des Datenbestandes muss mehrfach durchgeführt werden, um die Häufigkeit der Kandidaten zu bestimmen (mindestens eine Suche pro Kandidatenlänge).

2. Die Kandidatenmenge kann sehr groß werden, da beispielsweise 10.000 häufige 1-Kombinationen $\binom{10.000}{2} \approx 5 \cdot 10^7$ Kandidaten mit zwei Elementen erzeugen können, deren Häufigkeit ermittelt werden muss.

3. Lange häufige Kombinationen können jeweils viele Kandidaten erzeugen, da diese im Extremfall bei jedem Elemente mit einer anderen Kombination zu einem neuen Kandidaten vereinigt werden können.

4. Niedrige Mindesthäufigkeitswerte erzeugen durch einen exponentiell wachsenden Suchraum sehr lange und schwer vorhersagbare Algorithmenlaufzeiten.

Algorithmus 2.2 APRIORI

Require: Datenbestand DB
Require: Mindesthäufigkeit $minSupport$
Require: Menge häufiger 1-Kombinationen G_1

```
1:  k = 1
2:  I = G_1
3:  while G_k ≠ ∅ do
4:     k++
5:     G_k = apriori-gen(G_{k-1})
6:     for all x ∈ G_k do
7:        zaehler = count(x, DB) //Bestimme Häufigkeit von x in DB
8:        if zaehler < minSupport then
9:           G_k = G_k \ x
10:       end if
11:    end for
12:    I = I ∪ G_k
13: end while
14: return I
```

Bei Goethals [GZ03a] oder Kosters [KP03] kann nachgelesen werden, wie diese Probleme durch geschickte Datenstrukturen und Implementierungen gemildert werden können. Es wird gezeigt, dass APRIORI trotz vorhandener Schwächen eine sinnvolle Alternative zur EHK im Vergleich zu nachfolgend erläuterten Verfahren darstellen kann. Insbesondere, wenn wenige häufige Kombinationen extrahiert werden, kann der Verzicht auf die Berechnung komplexer Datenstrukturen die Nachteile von APRIORI leicht ausgleichen. Die folgenden Abschnitte stellen Algorithmen vor, bei denen eine Generierung von Kandidaten vermieden wird.

2.2.2. FP-Growth

Ohne ein schrittweises Durchsuchen des Datenbestandes nach bestimmten Kombinationen bzw. Kombinationskandidaten arbeiten Kombinationsbäume (engl. frequent pattern trees, kurz *FP-Tree*). Diese stellen eine kompakte Datenstruktur zur Erzeugung häufiger Kombinationen dar [HPY00]. Mit Hilfe eines Kombinationsbaumes wird der Datenbestand bereits beim Aufbau der Struktur stark eingeschränkt. Die Häufigkeiten von Kombinationen bleiben hierbei erhalten. Wenig relevante Informationen, wie z.B. die einzelnen Transaktionen, werden entfernt.

Ein Kombinationsbaum kann als eine teilweise vorberechnete Datenstruktur zur Bestimmung von Kombinationshäufigkeiten betrachtet werden. Der Aufbau des Kombinationsbaumes erfordert zwei Zugriffe auf die Ausgangsdaten. Dies bietet gegenüber dem

APRIORI-Algorithmus Vorteile, da dieser pro Kombinationslänge auf die vollständigen Ausgangsdaten zugreifen muss. Der erste Zugriff ermittelt alle Elemente, welche eine geforderte Mindesthäufigkeit erreichen. Im zweiten Durchlauf werden die gefundenen häufigen Elemente gemeinsam mit ihrer Transaktionsnummer aus den Ausgangsdaten gelesen und pro Transaktion nach der Häufigkeit der Elemente im gesamten Datenbestand sortiert in den Kombinationsbaum integriert.

Im Detail besteht ein Kombinationsbaum aus folgenden Bestandteilen:

1. Jeder Kombinationsbaum besitzt einen Wurzelknoten, eine Menge von Teilbäumen und ein *Elementverzeichnis* mit Zeigern pro Element auf das jeweils erste Vorkommen dieses Elementes im Kombinationsbaum.

2. Jeder Knoten der Baumstruktur besitzt drei Einträge: *Bezeichner*, *Zähler* und einen *Knotenverweis*. Der Bezeichner definiert den entsprechenden Knoten, der Zähler speichert die Häufigkeit dieses Knotens und der Knotenverweis verbindet alle Knoten mit gleichem Bezeichner.

3. Im Elementverzeichnis besteht jeder Eintrag aus einem *Element* und einem Verweis in den Kombinationsbaum auf den ersten Knoten mit gleichem Bezeichner.

Um eine Kombination in einen solchen Baum einzufügen, wird gemäß den aufeinander folgenden Elementen der Kombination der Baum traversiert. Pro betrachtetem Knoten wird dessen Zähler inkrementiert. Ist ein Knoten noch nicht vorhanden, wird er erstellt. In diesem Fall muss ebenfalls eine Anpassung der Knotenverweise und des Elementverzeichnisses erfolgen, um den neuen Knoten korrekt in die Struktur zu integrieren. Ist der Baum vollständig erstellt, können alle Knoten mit einem geringeren Zähler als die Mindesthäufigkeit entfernt werden. Der Baum wird damit um nicht häufige Kombinationen bereinigt.

Abbildung 2.3 zeigt in der linken Hälfte einen solchen Kombinationsbaum mit Elementverzeichnis basierend auf den Daten aus Tabelle 2.1. Die Pfeile repräsentieren in vereinfachter Form die Knotenverweise. Die Angaben in Klammern zeigen den Zähler des Knotens.

Aus einem Baum können effizient häufige Kombinationen durch Anwendung des FP-GROWTH-Algorithmus extrahiert werden. Ermöglicht wird dies durch die Eigenschaft der Datenstruktur, dass es für jedes häufige Element möglich ist, die Häufigkeiten von Kombinationen mit diesem Element zu erhalten. Dabei gilt die Bedingung, dass dieses Element das Seltenste in den zu ermittelnden Kombinationen ist.

Zur Berechnung häufiger Kombinationen für ein Element muss dessen Verweispfad gefolgt werden. Alle häufigen Kombinationen, bei denen dieses Element das Seltenste ist, können in einen Teilbaum materialisiert werden. Dazu müssen die Pfade von jedem Knoten im

TID	Elemente	TID	Elemente
T01	a b c d	T07	b c d
T02	c	T08	d
T03	d	T09	a b c
T04	d	T09	a b c
T05	d	T11	a b c d
T06	d	T12	c

Tabelle 2.1.: Transaktionsdatenbestand DB

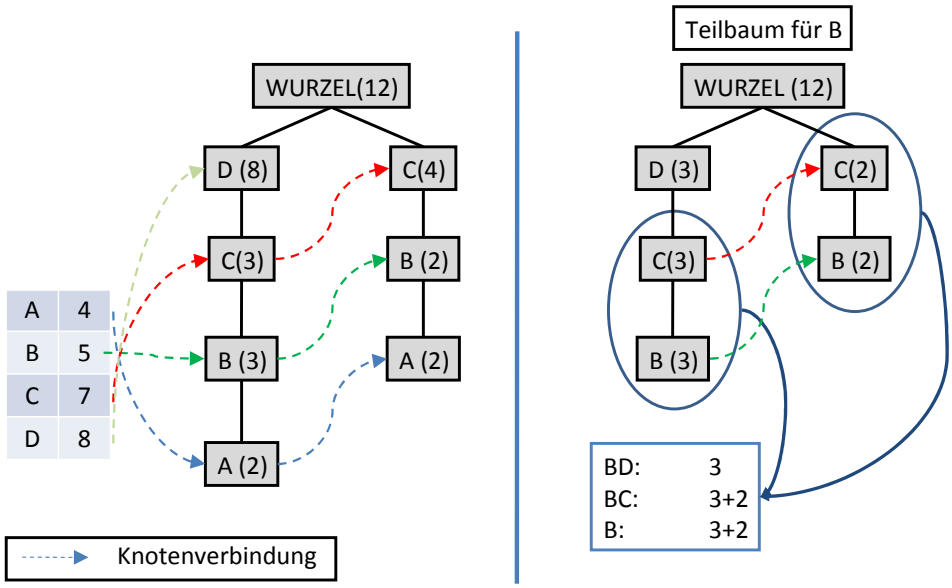


Abbildung 2.3.: Beispiel *FP-GROWTH*

Verweispfad zum Wurzelknoten betrachtet werden und alle dessen Zähler auf den Wert des Verweispfadknotens gesetzt werden. Durch Vereinigen der einzelnen Pfade ergibt sich der Teilbaum für das untersuchte Element. Abbildung 2.3 zeigt dies rechts für das Element B. Aus diesem Teilbaum können anschließend effizient alle häufigen Kombinationen mit dem seltensten Element B extrahiert werden. Wird diese Analyse für jedes weitere Element ausgeführt, kann die gesamte Menge der häufigen Kombinationen ermittelt werden.

Ein Kombinationsbaum ist meist eine signifikant kompaktere Darstellung des ihn erzeugenden Ausgangsdatenbestandes [HPY00]. Gleiches gilt für die bedingten Kombinationsbäume, da diese bereits ausschließlich häufige Elemente enthalten. Durch Vermeiden der Generierung unnötiger Ergebniskandidaten und die damit verbundenen wiederholten Datenzugriffe ist FP-GROWTH effizient. Außerdem ist von Vorteil, dass ein Kombinationsbaum für mehrere Suchen mit verschiedenen Mindesthäufigkeiten genutzt werden

kann. Dies ist jedoch nur möglich, wenn der Kombinationsbaum noch nicht mit einer größeren Mindesthäufigkeit beschnitten wurde.

Erweiterungen von FP-GROWTH werden in [GZ03b; GZ03c] präsentiert, wobei Anpassungen des Verfahrens für dünn besetzte Datenbestände vorgenommen werden. Dabei wird versucht, durch eine Abwandlung der Datenstruktur unnötige Rekursionsaufrufe zu vermeiden, da diese den Großteil der Laufzeit von FP-GROWTH beanspruchen. Eine Anpassung von FP-GROWTH an moderne Rechnerarchitekturen mit Mehrkernprozessoren werden in [LLZT07] beschrieben.

Insgesamt erlaubt die sehr kompakte Datenstruktur des Kombinationsbaumes eine effiziente Suche nach häufigen Kombinationen. Wird die Erzeugung des Baumes jedoch sehr aufwändig, können andere Verfahren, wie das nachfolgend beschriebene ECLAT, eine bessere Alternative darstellen.

2.2.3. Eclat

Die bereits in den vorangegangenen Abschnitten vorgestellten Algorithmen APRIORI und FP-GROWTH verwenden zur Bestimmung häufiger Kombinationen eine horizontale Darstellung der zu untersuchenden Transaktionen. Das bedeutet, dass jede Transaktion aus einem Transaktionsbezeichner TID sowie den dazugehörigen Elementen besteht. Durch Zaki, Parthasarathy, Ogihara und Li [ZPOL97] wird der *Equivalence-CLAss-Transformation*-Algorithmus ($ECLAT$) vorgestellt, dessen Besonderheit in der Verwendung einer vertikalen Repräsentation der Transaktionen besteht. Hierfür wird zu jedem Element aus E eine Liste mit Transaktionsnummern vorgehalten, in denen dieses Element vorkommt. Der Vorteil bei der Verwendung dieser vertikalen Darstellung besteht darin, dass die Häufigkeit eines Elementes direkt aus der Kardinalität der entsprechenden Liste ableitbar ist. Für die Erzeugung der häufigen Kombinationen wird zunächst die Schnittmenge der Transaktionslisten von je zwei häufigen Elementen gebildet. Enthält die Schnittmenge mehr Einträge als die geforderte Mindesthäufigkeit, kann die Vereinigung der beiden betrachteten Einzelelemente in die Ergebnismenge übernommen werden.

Aus den entstandenen häufigen 2-Kombinationen werden durch Bilden der jeweiligen Schnittmengen der Transaktionslisten zweier Kombinationen aus einer Äquivalenzklasse neue Kombinationen inklusive ihrer Häufigkeiten erzeugt. Eine Äquivalenzklasse besteht aus den Kombinationen mit demselben Präfix, wobei davon ausgegangen wird, dass die Elemente in einer festgelegten Sortierung vorliegen. Falls beispielsweise die drei Einträge A, B und C in einem Datenbestand auftreten, gehören die Kombinationen AB und AC zu einer Äquivalenzklasse, wenn als Sortierung die Ordnung $A < B < C$ definiert ist. Aus den Transaktionslisten von AB und AC können anschließend die Transaktionslisten für ABC durch Schneiden der Ausgangslisten ermittelt werden.

Alternativ können statt der Kombination zweier Einträge einer Äquivalenzklasse auch gefundene k-Kombinationen um weitere, noch nicht enthaltene, häufige Elemente erweitert werden. Die Häufigkeit der entstehenden Kombinationen wird äquivalent durch Schneiden der jeweiligen Transaktionslisten ermittelt. Einer Kombination werden damit schrittweise weitere Elemente hinzugefügt, bis die Mindesthäufigkeit nicht mehr erreicht wird.

Die Implementierung der Transaktionslisten ist der entscheidende Faktor für den Speicherverbrauch von ECLAT, insbesondere, wenn eine große Anzahl an Transaktionen analysiert werden soll. Durch Nutzung von Bitvektoren und ähnlichen Kompressionsmethoden lässt sich der Speicherverbrauch je nach der in dem Datenbestand vorhandenen Informationsdichte optimieren. Eine weitere von Zaki [ZG03] vorgestellte Möglichkeit besteht darin, zu jeder $(k+1)$-Kombination, welche aus einer k-Kombination erzeugt wurde, die Differenz der Transaktionsnummern der beiden Kombinationen vorzuhalten. Dies kann den benötigten Speicherplatz des Algorithmus deutlich verbessern, wenn sich die beiden Listen in großen Teilen überschneiden. Die reduzierte Datenmenge kann sich ebenfalls positiv auf die Effizienz des Algorithmus auswirken.

Für ECLAT existieren eine große Anzahl an Abwandlungen und auf bestimmte Probleme optimierte Versionen. Eine Implementierung zur Auslagerung der Berechnungen von ECLAT auf Grafikkarten-Prozessoren wird beispielsweise in [GHH+08] evaluiert.

2.2.4. BUC

Im Gegensatz zu APRIORI arbeitet der von Beyer [BR99] vorgestellte Bottom-Up-Computation-Algorithmus (BUC) mit Tiefensuchen statt einer Suche in die Breite des Suchraumes. Das bedeutet, es wird versucht, aus kurzen Kombinationen längere Kombinationen abzuleiten. BUC weist damit Ähnlichkeiten zu ECLAT auf, ist jedoch nicht für Datenbestände mit Transaktionen geeignet. BUC dient zum effizienten Auffinden häufiger Attributwertkorrelationen und ermittelt Kombinationen $(e_1, e_2, e_3, \ldots, e_n)$ basierend auf den Ergebnissen einer Kombinationen $(e_1, e_2, e_3, \ldots, e_{n-1})$.

Die Anwendung von BUC erfolgt für Suchen nach häufigen Kombinationen von Attributwertausprägungen für mehrere verschiedene Attribute einer Transaktion oder Zeile. Gesucht werden beispielsweise Korrelationen zwischen Farbe und Ausstattung eines Produktes, nicht die Verbindung zwischen Produkten. Im Gegensatz zu einer Warenkorbanalyse entspricht die Anzahl an Elementen der Anzahl der zu betrachtenden Attribute einer solchen Zeile und ist somit pro Suche konstant. Die Einsatzfähigkeit von BUC ist damit eingeschränkt und weniger generisch als beispielsweise die Algorithmen ECLAT und APRIORI. Allerdings sind Effizienzsteigerungen möglich.

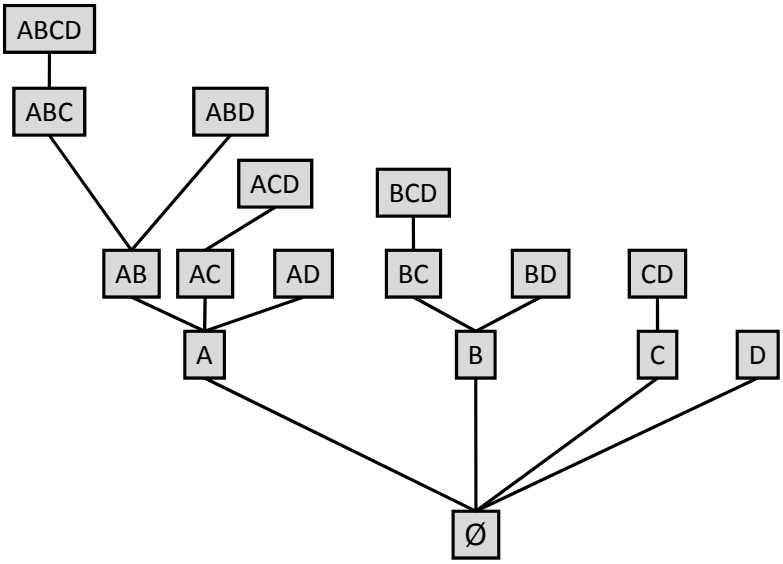


Abbildung 2.4.: Suchraum für die Attribute A, B, C, D mit BUC

Abbildung 2.4 erläutert den Ablauf von BUC. Kombinationen werden von unten nach oben, beginnend von links nach rechts im dargestellten Baum evaluiert. Begonnen wird mit der Bestimmung der Häufigkeiten der Einzeleinträge A, B, C und D. Deren Ermittlung kann in einem einzelnen Schritt parallel erfolgen. Ausgehend von den ermittelten Zeilen- bzw. Transaktionsnummern für A kann der linke Ast AB, ABC und $ABCD$ schrittweise rekursiv evaluiert werden. Äste ohne gleiche Teilbereiche sind jeweils voneinander unabhängig. Sie sind ebenfalls parallel durchsuchbar.

Zur Auswertung werden für alle Ausprägungen eines betrachteten Attributes die darauf folgenden Teilbäume untersucht. Dies erfolgt in Abhängigkeit von dem aktuell betrachteten Element rekursiv für alle Elemente der weiteren Attribute. Als Beispiel soll der Ast $\emptyset \rightarrow C \rightarrow CD$ dienen. Begonnen wird mit den Elementen c_1 bis c_n von Attribut C. Erreicht c die Mindesthäufigkeit, wird in Abhängigkeit von jedem c die Häufigkeiten aller Attributausprägungen d_1 bis d_m von Attribut D bestimmt. Überschreitet eine Kombination die geforderte Mindesthäufigkeit, wird sie in das Endergebnis übernommen. Beim Verfehlen von *minSupport* wird der Abstieg zu weiteren Attributen beendet.

Da in jedem Schritt mit wachsender Kombinationslänge die Menge der zu betrachtender Zeilen verringert wird, erfolgt die Bestimmung der Häufigkeit längerer Kombinationen durch die Berücksichtigung der vorhergegangenen Kombinationen effizient. Der Abstieg innerhalb eines Astes wird beendet, sobald die Mindesthäufigkeit durch die Kombination $X = (e_1, e_2, e_3, \ldots, e_n)$ nicht mehr erreicht wird oder die Kombination nicht um einen weiteren Eintrag auf $(e_1, e_2, e_3, \ldots, e_n, e_{n+1})$ erweitert werden kann. Entsprechend wird mit einer absoluten Mindesthäufigkeit von 1, und damit ohne vorzeitigen Abbruch der Suche, die Häufigkeit aller vorhandenen Kombinationen berechnet.

Um die Analyse des Suchraumes zu optimieren, werden die zu betrachtenden Attribute nach der Anzahl ihrer verschiedenen Ausprägungen sortiert. In Abbildung 2.4 ist beispielsweise A das Attribut mit der größten Anzahl verschiedener Ausprägungen und Attribut D das mit der geringsten Anzahl. Beyer [BR99] beschreibt diese Optimierung. Er nutzt die Annahme, dass für gleichmäßig verteilte Ausprägungen die durchschnittliche Häufigkeit dieser Einträge mit steigender Anzahl möglicher Einträgen sinkt. Unter der Annahme, dass diese Abschätzung der Realität des zu betrachtenden Datenbestand entspricht, bewirkt die reduzierte Anzahl gefundenen Einträge für die Ausprägung a im Attribut A weniger Treffer in Kombination mit den Ausprägungen in den Attributen B, BC und BCD. Die Häufigkeit der entsprechenden Kombinationen $(a_1, b_1), (a_1, b_1, c_1)$ bis (a_q, b_r, c_s, d_t) wird effizienter bestimmt, da die Anzahl an Zeilen von a bereits reduziert ist. Die Menge der insgesamt betrachteten Zeilen wird verringert. Im schlimmsten Fall kann bei dieser Heuristik der Fall eintreten, dass ein Attribut mit vielen verschiedenen Werten sehr geringe Häufigkeiten aufweist, aber einige wenige Werte überproportional häufig sind. In diesem Fall wird dieses Attribut früher abgearbeitet, als dies bei lediglich den zwei dominanten Ausprägungen erfolgt wäre. Die „Optimierung“ kann die Anzahl der untersuchter Zeilen in dem Fall deutlich steigern.

Weitere Heuristiken, wie Sortierungen in Abhängigkeit von den Attributwertverteilungen, sind ebenfalls möglich. Im Rahmen dieser Arbeit wurde lediglich eine Sortierung nach Anzahl verschiedener Attributausprägungen untersucht. Nach Beyer können für praxisnahe Datenbestände auch keine deutlichen Unterschiede bei anderen Sortierungsstrategien festgestellt werden. Neben der Bestimmung von häufigen Kombinationen kann das Konzept von BUC ebenso auf die Berechnung von (Teil-)Datenwürfeln [HK05b] angewendet werden, indem das Zählen des Auftretens einer Kombination durch das Errechnen einer bestimmten Kennzahl ersetzt wird.

2.2.5. Partition

Ist ein Datenbestand zu umfangreich, um auf einem einzelnen Rechenknoten verarbeitet werden zu können oder ist aus anderen Gründen eine paketweise Verarbeitung der Daten erforderlich, muss die Suche nach häufigen Kombinationen verteilt erfolgen. Eine der grundlegenden Ideen zur Lösung dieses Problems wurde von Savasere, Omiecinski und Navathe unter dem Namen PARTITION vorgestellt [SON95].

Der PARTITION-Algorithmus besteht im Wesentlichen aus den folgenden Schritten:

1. Suche auf allen Datenpartitionen nach lokal häufigen Kombinationen mit der global geforderten relativen Mindesthäufigkeit.

2. Vereinige die lokalen Ergebnisse aller Partitionen (lokale Häufigkeiten werden aufsummiert).

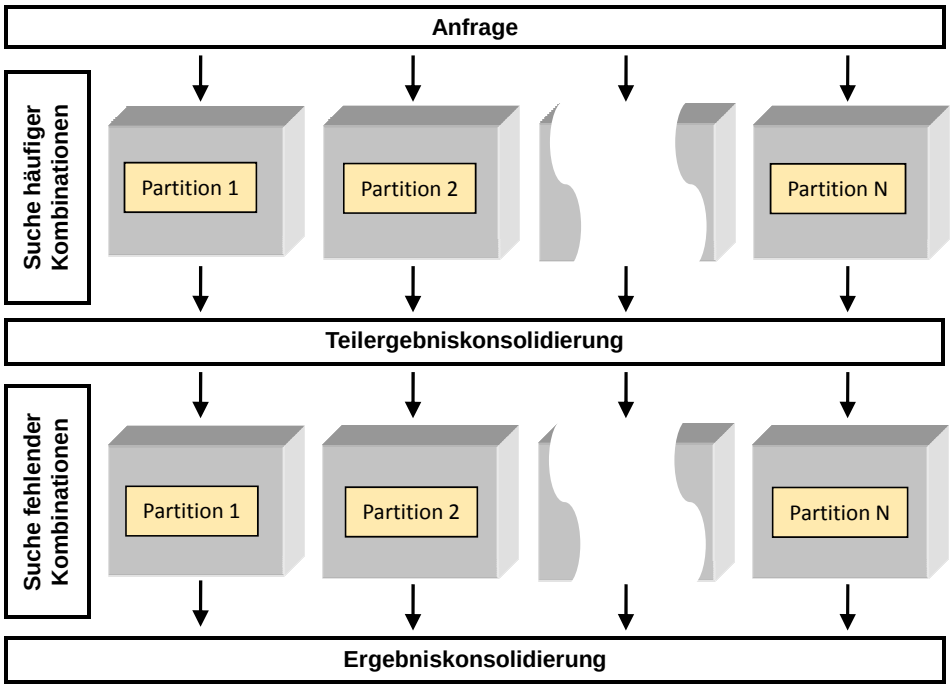


Abbildung 2.5.: Übersicht PARTITION

3. Suche Kombinationen nach, welche auf mindestens einer, aber nicht auf allen Partitionen gefunden wurden.

4. Vereinige die Ergebnisse von 2. und 3. zum globalen Gesamtergebnis.

Abbildung 2.5 zeigt den Ablauf von PARTITION in der Übersicht. Die Wahl des Algorithmus für den ersten Schritt ist hierbei frei, solange alle in der jeweiligen Partition vorhandenen Kombinationen oberhalb der Mindesthäufigkeit erfolgreich ermittelt werden. Wird die Mindesthäufigkeit relativ angegeben, ist sie direkt zu allen Einzelpartitionen übertragbar. Absolute Angaben müssen im Verhältnis der jeweiligen Partitionsgröße zum Umfang des gesamten Datenbestandes angepasst werden.

Durch Savasere wird gezeigt, dass jede global häufige Kombination mit diesem Verfahren garantiert gefunden wird. Eine Kombination erreicht auf mindestens einer lokalen Partition die relative Mindesthäufigkeit, wenn sie global ebenfalls häufig ist. Sobald mindestens eine Partition eine Kombination als Ergebniskandidat liefert, wird diese auf allen Partitionen nachgesucht. Damit kann die globale Häufigkeit dieser Kombination ermittelt werden. Sie wird anschließend auf das Erreichen der globalen Mindesthäufigkeit überprüft und gegebenenfalls der Ergebnismenge hinzugefügt.

Zur Optimierung des zweiten und dritten Schrittes kann eine differenzierte Betrachtung der ermittelten Zwischenergebnisse vorgenommen werden. Zum Zeitpunkt des zweiten Schrittes ist spätestens die Anzahl der insgesamt beteiligten Transaktionen bekannt. Somit kann an dieser Stelle ermittelt werden, welche absolute Häufigkeit zum Erreichen der globalen Mindesthäufigkeit benötigt wird. Dieser Wert kann als absoluter Schwellwert gesetzt werden. Unter dessen Nutzung kann eine Kombination mit mindestens einer fehlenden Teilhäufigkeit die folgenden Eigenschaften aufweisen:

1. Sie kann bereits global häufig sein, aber potenziell noch nicht die korrekte globale Häufigkeit aufweisen.

2. Sie kann global noch nicht häufig sein, aber die Mindesthäufigkeit mit Hilfe der fehlenden lokalen Häufigkeiten erreichen können.

3. Sie kann global nicht häufig sein und die globale Mindesthäufigkeit auch nicht durch fehlende lokale Häufigkeiten erreichen.

Die Kombinationen des dritten Falls können bereits nach dem zweiten Schritt verworfen werden. Wenn keine globalen Häufigkeiten benötigt werden und die Klassifizierung als häufige Kombination ausreichend ist, können die Kombinationen des ersten Falles bereits dem Endergebnis hinzugefügt werden.

Ein Vorteil von PARTITION besteht in der von den untersuchten Daten unabhängigen festen Anzahl an Arbeitsschritten. Diese erleichtern den Umgang mit PARTITION, da der Arbeitsverlauf im Vorfeld abschätzbar ist. Der Schritt der Vereinigung von lokalen Ergebnissen kann jedoch einen Schwachpunkt des Verfahrens bilden, da hier keine verteilte Berechnung stattfindet. Dieser Rechenknoten muss somit in der Lage sein, die nötige Kommunikation, die Vereinigung aller Teilergebnisse und die Steuerung der restlichen Knoten zu leisten. Außerdem stellt er einen Synchronisationspunkt aller beteiligter Knoten dar, da erst bei vollständig verfügbaren Teilergebnissen eine Vereinigung erfolgt und sich einzelne Knoten zeitweise im Leerlauf befinden. Die Verwendung von PARTITION ist somit nicht in allen Anwendungsgebieten, wie z.B. in Netzwerken mit sehr vielen Knoten, sinnvoll. Die Verwendung bei einer vergleichsweise geringen Anzahl beteiligter Knoten ist aber möglich.

Nachdem einige Algorithmen zur Berechnung häufiger Kombinationen erläutert wurden, sollen die folgenden Abschnitte mögliche Vereinfachungen der Verfahren untersuchen. Damit soll sowohl die Ausführung einer EHK durch nicht speziell geschulte Anwender als auch ein robusteres Laufzeitverhalten der Algorithmen ermöglicht werden.

2.2.6. Top-N-Berechnungen

Während die in den vorherigen Abschnitten vorgestellten Verfahren zur EHK den Suchraum jeweils durch die Definition einer Mindesthäufigkeit einschränken, versuchen andere Lösungsansätze die EHK durch Angabe der gewünschten Ergebniskardinalität zu steuern. Ein oft verwendeter Ansatz zur Lösung solcher Problemstellungen besteht in der Verwendung von Branch-and-Bound-Verfahren [LD60]. Begonnen wird hierbei ohne Einschränkungen möglicher Lösungen auf den vollständigen Daten. Aus den im Laufe der Verarbeitung ermittelten Erkenntnissen können dann Teilbereiche des Suchraumes (engl. branches) ausgeschlossen und begrenzt (engl. bound) werden. Wird dieses Konzept auf die Suche nach den häufigsten N Ergebnissen übertragen, ergibt sich folgende mögliche Vorgehensweise

1. Suche nach Kombinationen mit minimalem Schwellwert ohne Einschränkungen.

2. Durchlaufen des Suchraumes und Einsortierung gefundener Ergebnisse nach ihrem Wert in eine Ergebnisliste.

3. Bei Erreichen von mindestens N Ergebniseinträgen wird der Schwellwertes jeweils auf die Häufigkeit des N-ten Ergebnisses gesetzt. Neue Kandidaten müssen diesen Schwellwert erreichen, um in die Ergebnisliste zu gelangen.

4. Ergebniskandidaten, welche im Laufe der Steigerung des Schwellwertes diesen nicht länger erreichen, werden aus der Liste der Ergebniskandidaten entfernt.

5. Die endgültigen Einträge der Ergebnisliste bilden das Top-N-Ergebnis.

Dieses Verfahren ist effizient, wenn sich der Schwellwert schnell an die Häufigkeit der tatsächlichen N-ten Kombination angleicht und damit nur wenige für das Endergebnis irrelevante Teile des Suchraumes betrachtet werden. Dies kann nicht garantiert werden und führt bei einem nur langsam steigenden Schwellwert zu erhöhten Laufzeiten und unnötig gespeicherten Ergebniskandidaten.

Es ergeben sich folgende positiven Eigenschaften von Top-N-Berechnungen:

- Bedienung und Parameterdefinition sind auch für nicht spezialisierte Anwender einfach.

- Parameterdefinition erfolgt unabhängig von den zu untersuchten Daten. Die Umsetzung der Top-N-Parameter auf die Daten obliegt dem jeweiligen Algorithmus.

- Ergebnisumfang ist bekannt und eine sinnvolle Vorhersage der Laufzeiteigenschaften und des Ressourcenverbrauches möglich.

Negativ ist hingegen:

- Bei verteilten Berechnungen kann aus lokalen Top-N nur bedingt auf die globalen Top-N-Ergebnisse geschlossen werden.
- Die Mindesthäufigkeit kann unter bestimmten Bedingungen langsam ansteigen. Dies kann potenziell vermeidbare Kosten bewirken.
- Kandidaten werden erzeugt, welche nachträglich wieder verworfen werden.
- Die Ergebnisliste muss sortiert vorliegen und kontinuierlich gepflegt werden.

Die Top-N-Bestimmung häufiger Kombinationen mit spezifischen Charakteristiken wurde in anderen Arbeiten bereits untersucht. Beispielsweise spezialisiert sich Cong [Con01] auf eine Suche nach Top-N-Kombinationen mit einer vordefinierten Mindestkombinationsbreite. Eine solche Optimierung ist für diese Arbeit wenig relevant, da nach praktischen Erfahrungen meist kurze Kombinationen gesucht werden und außerdem zur Bestimmung von Assoziationsregeln immer die Häufigkeiten der Teilkombinationen einer gefundenen häufigen Kombination nötig sind. Beschränkt man die Suche auf ausschließlich lange Kombinationen und liefert weder kurze Kombinationen noch deren Häufigkeiten, ist die Berechnung von Interessantheitsmaßen einer solchen Assoziationsregel nicht ohne Nachsuchen der zugrunde liegenden Teilhäufigkeiten möglich.

Für praxisnahe Verkaufsanwendungen, wie hauptsächlich in dieser Arbeit betrachtet, werden meist kleine Kombinationen mit ein bis zwei Elementen in Bedingung und Schlussfolgerung einer Regel bevorzugt, wodurch sich die benötigte Länge einer zu suchenden Kombination auf maximal $2 + 2 = 4$ Elemente beschränkt. Damit sind alle zur Berechnung von Interessantheitsmaßen nötigen Häufigkeiten verfügbar. Trotzdem bleibt der Grundgedanke von Cong relevant, da er interessante Möglichkeiten für eine optimierte Top-N-EHK veranschaulicht. Er soll daher kurz erläutert werden.

Die wesentliche Besonderheit des Verfahrens von Cong besteht im Vorhalten einer Liste von N Ergebniskandidaten inklusive je einer Liste mit deren Transaktionsnummern. Ausgehend von der Liste wird der jeweils häufigste nicht mit Status *verarbeitet* markierte Ergebniskandidat um weitere mögliche Elemente erweitert und in den Status *verarbeitet* versetzt. Diese neuen Ergebniskandidaten werden evaluiert und in die Liste der bestehenden Kandidaten mit ihren Häufigkeiten eingefügt. Alle Kandidaten außerhalb der sich daraus ergebenden Top-N-Liste werden entfernt. Dieser Vorgang wiederholt sich, bis sich alle Ergebniskandidaten im Status *verarbeitet* befinden. Ein solcher Algorithmus ist sehr effizient beim Erzeugen von Ergebniskandidaten, aber entgegen der Funktionsprinzipien der meisten EHK-Verfahren müssen sehr viele Zwischenergebnisse gespeichert werden. Unnötige Informationen werden nicht schnellstmöglich verworfen und stattdessen sehr lange vorgehalten. Für die in dieser Arbeit adressierten typischen SAP-Szenarien mit $N \geq 10.000$ und Millionen von Transaktionen ist dieses Verfahren ungeeignet. Ein ähn-

licher Ansatz für das auf einem Kombinationsbaum basierende Verfahren CLOSET wird von Han [HWLT02] unter dem Namen TFP beschrieben.

Betrachtet man die Bestimmung von Häufigkeiten einer Kombination als Aggregation mit Bestimmung der Gruppengrößen, zeigen sich große Ähnlichkeiten. Prinzipien und Algorithmen für Aggregationsverfahren können daher eine sinnvolle Ausgangslage zur Top-N-EHK bilden. Für Aggregationsoperationen sind sowohl unverteilte als auch verteilte Top-N-Strategien verfügbar. Der Anwendungsfall betrachtet meist jedoch sehr einfache und schnelle Aggregationen und legt Wert auf eine Optimierung der übertragenen Datenmengen. Die betrachtet Systemarchitektur besteht hierbei meist aus sehr vielen unabhängigen, kleinen Rechenknoten. Diese sind in einem Netzwerk untereinander verbunden. Da vor allem die Kommunikation zwischen den Knoten betrachtet wird, werden lokale Aggregationen als schnell ermittelbare, für die Laufzeit unbedeutende Operationen betrachtet.

Ein typisches Beispiel einer verteilten Aggregation in solchen Netzen stellt die Bestimmung der häufigsten angefragten Netzwerkadressen aus einer großen Menge Verteilern dar. Während jeder Knoten seine wichtigsten Adressen leicht feststellen kann, sind globale Ordnungen nicht lokal ableitbar. Die Kommunikation aller lokalen Ordnungen wird nötig, um globale Aussagen treffen zu können. Dies dominiert leicht die Kosten der lokalen Sortierungen und wird damit zum Hauptziel von Laufzeitoptimierungen.

Top-N-Aggregationen unterscheiden sich daher in ihrer Verarbeitung deutlich gegenüber der Bestimmung von Top-N-Kombinationen. Zwischen beiden bestehen signifikante Laufzeitunterschiede, welche eine direkt Übertragung dieser Verfahren von einer Aggregation auf eine EHK verhindert. Die Laufzeiten für verteilte Aggregationen werden als kurz definiert. Die Bestimmung der Häufigkeiten aller möglichen Kombinationen während einer EHK ist hingegen oft ein schwieriger und langwieriger Prozess.

Die Behandlung von verteilten einfachen Aggregationen weist jedoch Gemeinsamkeiten zur EHK auf, welche im Folgenden näher dargelegt werden. Insbesondere die Grundlagen einer verteilten Top-N-Berechnung sind hierbei wichtig. Hierfür relevante Arbeiten befassen sich vor allem mit Top-N-Aggregationen in Peer-to-Peer-Netzwerken (P2P). Diese bestehen üblicherweise aus sehr vielen Rechenknoten mit eingeschränkter Kommunikationsfähigkeit und stark partitionierten Datenbestände. Auf diesen Rechnerarchitekturen ist damit nur selten die Ermittlung eines vollständigen Ergebnisses möglich oder sinnvoll und oft nicht erforderlich. Eine Möglichkeit, eine Top-N-Berechnung durchzuführen, stellen Synopsen dar. Sie werden benutzt, um entstehende Fehler einzuschränken und beschreiben zu können [HKSZ05]. Alternativ nutzen aktuelle Lösungen Skyline-Operationen [BKS01], um ein sinnvolles Ergebnis von Top-N-Berechnungen mit vertretbarem Aufwand zu erreichen [VDNV08]. Dies sind jedoch sehr spezielle Verfahren für P2P-Netzwerke.

Vorhandenen Verfahren zur Bestimmung verteilter Top-N-Aggregationen in kleinen Netzwerken sind in der Gruppe der Threshold-Algorithmen (TA) zu finden. Diese wurden ur-

sprünglich vor allem im Multimedia-Bereich verwendet [FLN01; GBK00; NR99]. Davon ausgehend wurde von Cao und Wang [CW04] eine Erweiterung mit dem Namen *TPUT* eingeführt, welche die Bestimmung verteilter Top-N-Aggregationen in maximal drei Runden garantiert. Diese wurde von Michel, Triantafillou und Weikum [MTW05; TWS04] erweitert und erlaubt mittels Bloom-Histogrammen [Blo70] eine Schätzung des Endergebnisses mit Qualitätsangaben nach zwei Runden mit deutlich verringerter Laufzeit und Datenübertragungskosten. Beide Verfahren optimieren die insgesamt übertragene Datenmenge und setzen den Aufwand zur Berechnung lokaler Aggregationen als gering voraus. Dies steht im deutlichen Widerspruch zur EHK, wodurch eine direkte vollständige Übertragung des Verfahrens verhindert wird. Der grundlegende Ablauf kann aber zur Lösung einer verteilt ermittelten EHK genutzt werden.

Aufbauend auf den in den vorherigen Abschnitten eingeführten Algorithmen zur EHK wurden in der Literatur bereits eine Vielzahl an Abwandlungen zur Lösung spezieller Problemstellungen vorgestellt. Es existieren beispielsweise Spezialisierungen für die Suche nach maximalen Kombinationen [Bay98] oder geschlossenen Kombinationen [PBTL99]. Unter maximalen Kombinationen werden nur Kombinationen verstanden, welche nicht Teilkombination einer anderen häufigen Kombination sind. Geschlossene Kombinationen sind solche, welche nicht Teil einer anderen gleichhäufigen Kombination sind. Maximale Kombinationen sind in dem Kontext einer Assoziationsregelsuche nicht relevant, da eine Vielzahl der zur Regelbildung nötigen Kombinationen und deren Häufigkeiten aus der Ergebnismenge entfernt werden und nicht rekonstruierbar sind. Diese Einschränkung gilt nicht für geschlossene Kombinationen. Für diese entsteht kein Informationsverlust gegenüber einer vollständigen Suche. Die Beschränkung der Ergebnismenge auf ein solches Konstrukt kann die Aufbereitung der von einer EHK ermittelten Informationen deutlich erleichtern.

2.2.7. Geschlossene Kombinationen

Aus der Downward-Closure-Eigenschaft häufiger Kombinationen ist direkt ableitbar, dass alle Teilkombinationen X' einer Kombination X mindestens genauso häufig auftreten wie X selbst. Dabei kann der Fall eintreten, dass die Häufigkeit von X mit der einer seiner Teilkombinationen X' übereinstimmt. Das sind alle Fälle, bei denen die Assoziationsregel $X' \Rightarrow (X \setminus X')$ eine Konfidenz von 100% aufweist. In diesem Fall ist die gespeicherte Information über X' redundant, da sie vollständig aus den Informationen von X abgeleitet werden kann. Die Menge H aller $X \in I$, welche keine gleich häufige Obermenge in I besitzen, werden als geschlossene Kombinationen (engl. frequent closed itemsets (FCI)) bezeichnet [PBTL99]. Diese Menge kann gegenüber der Ausgangsmenge I eine deutlich reduzierte Kardinalität aufweisen und es gilt stets $|H| \leq |I|$.

Definition 2.1 (Geschlossene Kombination)**.** *Eine Kombination X' wird als geschlossene Kombination bezeichnet, wenn keine echte Obermenge X von X' existiert, welche in allen*

Transaktionen L in einem Datenbestand DB welche X' enthalten, ebenfalls vorhanden ist.

Aus der Menge der geschlossenen häufigen Kombinationen ist die Rekonstruktion der Menge aller häufigen Kombinationen vollständig möglich. Zur Wiederherstellung wird hierfür die Abwesenheit von Kombinationen genutzt, die nach der Downward-Closure-Eigenschaft vorhanden sein müssten. Da X' eine Teilkombination von X ist, muss sie ebenfalls in einer Ergebnismenge I vorhanden sein, in der X auftritt. Ist dies nicht der Fall, wurde X' durch die Reduktion auf geschlossene Kombinationen entfernt und kann zur Rekonstruktion von I aus H wieder zu den Ergebnissen hinzugefügt werden. Die Häufigkeit von X' entspricht dabei der maximalen Häufigkeiten aller Kombinationen X, welche Obermenge von X' in H sind.

Daraus ergeben sich für geschlossene Kombinationen folgende Eigenschaften:

1. Die Menge der geschlossenen Kombinationen H ist eine Teilmenge der Ausgangskombinationen I.

2. Die Menge der ursprünglichen Ausgangskombinationen kann aus der Menge der geschlossenen Kombinationen vollständig rekonstruiert werden, indem alle Teilkombinationen der geschlossenen Kombinationen in H gebildet werden.

In der Praxis ist eine Einschränkung auf geschlossenen Kombinationen oft sinnvoll. Wird von einem gesteigerten Informationsgehalt durch weitere Elemente bei gleicher Häufigkeit ausgegangen, ist die Interessantheit der geschlossenen Kombination für einige wichtige Interessantheitsmaße, wie Häufigkeit und Konfidenz, gleich oder höher als bei allen von ihr dominierten, nicht geschlossenen Teilkombinationen. Ist dies nicht der Fall, kann die vollständige Menge häufiger Kombinationen leicht rekonstruiert werden. Zudem sind die aus geschlossenen Kombinationen gebildeten Assoziationsregeln meist eine wichtige Teilmenge aller möglichen Kombinationen [PTB+05; Zak04].

In der Literatur sind für viele der allgemeinen Verfahren zur EHK auch Abwandlungen für die Suche nach geschlossenen Kombinationen verfügbar [YHN06; HGN00; TCRB06]. In dieser Arbeit sollen stellvertretend Varianten für die Algorithmen FP-GROWTH und ECLAT erläutert werden.

CLOSET

Der CLOSET-Algorithmus, eine Implementierung der Suche nach geschlossenen häufigen Kombinationen basierend auf dem FP-GROWTH-Algorithmus, wurde von Pei, Han und Mao [PHM00] vorgestellt. Wie FP-GROWTH basiert CLOSET auf Kombinationsbäumen. Die Häufigkeiten alle vorkommenden Kombinationen werden bei der Erstellung der

Baumstruktur schrittweise implizit hinterlegt bzw. können effizient aus dieser ermittelt werden.

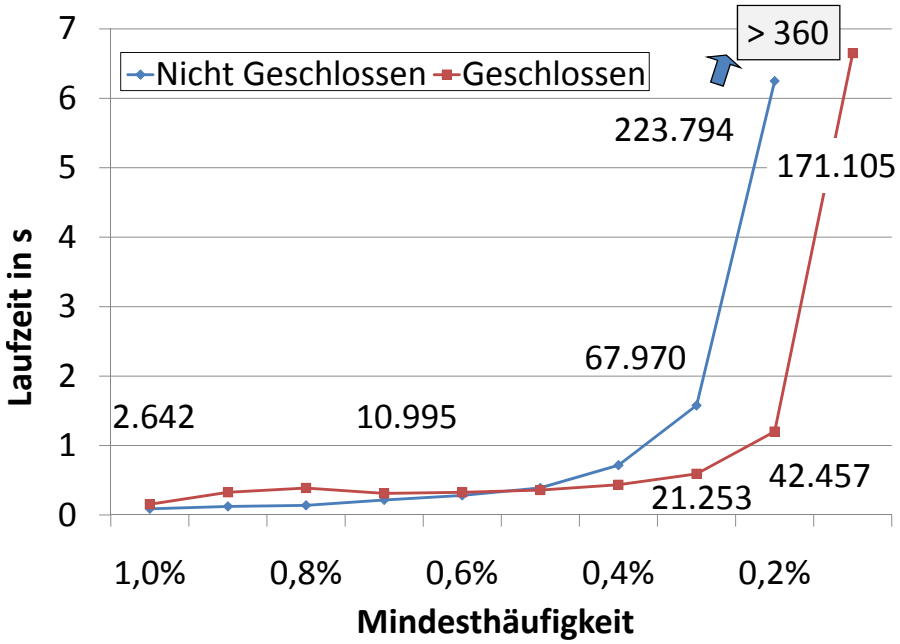
Zur Berechnung geschlossener Kombinationen wird die Baumstruktur pro Evaluationsschritt rekursiv in disjunkte Teilbäume zerlegt, welche dann einzeln untersucht werden. Bleibt die Häufigkeit für einige Zweige trotz Abstieg im Baum konstant, wird dieser Zweig als geschlossen markiert. Zusätzlich wird für jede Kombination geprüft, ob sie durch eine bereits bekannte Kombination repräsentiert wird oder diese repräsentieren kann. Hierfür sind mehrere Optimierungen möglich [PHM00; WHP03].

CHARM

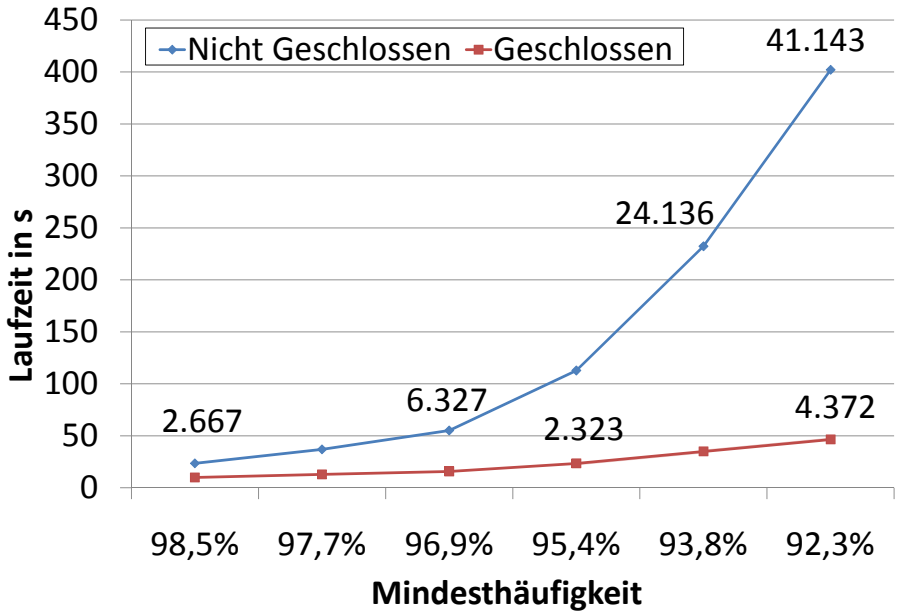
Der CHARM-Algorithmus ist ein Verfahren zur Suche nach geschlossenen Kombinationen basierend auf dem ECLAT-Algorithmus. Anders als beim ursprünglichen ECLAT werden nicht Transaktionslisten verwendet, sondern jeweils nur die Änderungen der Listen gegenüber dem jeweiligen Elternknoten betrachtet. Diese werden als *Diffsets* bezeichnet. Eine entsprechende Abwandlung für ECLAT ist als dECLAT bekannt [ZG03]. Die Besonderheit gegenüber ECLAT besteht darin, dass der Suchraum rekursiv jeweils von einer bereits evaluierten Kombination nach größeren Kombinationen durchsucht wird. Ist eine solche Kombination gleich häufig wie eine bereits bekannte Kombination, ist die ursprüngliche Kombination nicht geschlossen und kann verworfen werden. Auch hier wird für jede Kombination geprüft, ob sie bereits im Ergebnis repräsentiert wird oder ein bekanntes Ergebnis ersetzen kann.

Eine weitere Besonderheit entsteht durch die Verwendung der Diffsets, da diese effiziente Tests auf Geschlossenheit erlauben. Dabei wird eine Abbildungsfunktion auf die zu einer gefundenen Kombination gehörende Transaktionsliste angewandt und mit der Kombination zusammen temporär gespeichert. Vor dem Einfügen eines neuen Kandidaten als geschlossene Kombination muss geprüft werden, ob dieser Kandidat nicht bereits von einem der bereits vorhandenen geschlossenen Kombination repräsentiert wird. Da der Kandidat mit einer ihn repräsentierenden Kombination die gleichen Transaktionslisten aufweisen muss, ist deren Abbildungsfunktionswert gleich und eine Übereinstimmung kann über diesen effizient ermittelt werden. Die Abbildung muss dabei eindeutig zu einer entsprechenden Transaktionsliste zugeordnet werden können. Da Kollisionen möglich sind, muss für den Fall eines gleichen Abbildungsfunktionswertes eine zusätzliche Überprüfung auf die jeweiligen Transaktionslisten erfolgen.

Die Diagramme in Abbildung 2.6 geben einen Eindruck über die Auswirkungen einer Beschränkung auf geschlossene Kombinationen anhand zweier Beispiele. Dabei werden jeweils die entstehenden Laufzeiten für eine ECLAT/CHARM-Implementierung in Abhängigkeit von der zu der entsprechenden EHK definierten Mindesthäufigkeit gezeigt. Die verwendeten Daten werden im Anhang auf Seite 183 erläutert. Die Zahlen an den Knoten des Graphen repräsentieren die für diese Messung erzeugte Kardinalität der Ergebnismenge. Es wird gezeigt, dass im Bezug auf die Algorithmenlaufzeit die Kosten zur



(a) Datenbestand SynthA



(b) Datenbestand Connect-4

Abbildung 2.6.: Ergebnismengen bei der Suche geschlossener Kombinationen

Reduktion von I auf H den Gewinn übersteigen können. Für geringe Mindesthäufigkeiten und sehr dichte Datenbestände kann aber eine signifikante Laufzeitverbesserungen bewirkt werden. Die teilweise signifikante Verkleinerung der erzielten Ergebnismenge bietet weitere Vorteile. Die Interpretation der Ergebnismenge wird beispielsweise vereinfacht. Der Wert dieser Reduktion kann damit die potenziell gesteigerten Laufzeiten je nach Anwendung rechtfertigen.

In [JYP03; XHYC05; CKN06; PDZH04] werden unter der Bezeichnung *Condensed* oder *δ-tolerant Itemsets* Erweiterungen des Konzeptes der geschlossenen Kombinationen vorgeschlagen. Hierbei sind auch Kombinationen X' durch andere Kombinationen X repräsentiert, wenn diese nicht die exakt gleichen Häufigkeiten aufweisen. In [PDZH04] wird eine feste Abweichung zwischen den absoluten Häufigkeiten zweier Kombinationen X' und X toleriert. Bei [JYP03; PDZH04] wird pro unterschiedlichem Element zweier Kombinationen eine relative Abweichung δ der Häufigkeit erlaubt. Die Funktionsweise der Algorithmen ist angelehnt an CHARM und CLOSET. Lediglich die strikte Überprüfung auf Geschlossenheit wird durch einen unscharfen Test ersetzt.

Neben der Verwendung geschlossener Kombinationen sind weitere Möglichkeiten zur Eingrenzung der zu suchenden Kombinationen möglich, wovon einige in den folgenden Abschnitten betrachtet werden.

2.2.8. Regel- und Kombinationsvorlagen

Aus Begrenzungen der minimalen und maximalen Kombinationslänge für Bedingung und Schlussfolgerung einer Assoziationsregel lassen sich mögliche Einschränkungen für die EHK ableiten. Werden häufige Kombinationen beispielsweise nur zur Erzeugung von Assoziationsregeln und deren Interessantheitsmaßen benötigt, können diese gezielt gesucht werden. Diese Arten der Beschränkungen werden in der Literatur als *Constraints* bezeichnet [SVA97; NLHP98]. Andere Ansätze nutzen Taxonomien [SA97] oder Diskretisierungen [SA96; AL99], um die Menge der betrachteten Produkte bzw. Elemente zu reduzieren. Diese Verfahren sind als hierarchische bzw. quantitative Assoziationsregelanalysen bekannt. Im Folgenden soll sich jedoch auf die Begrenzung der Länge der zu suchenden Kombinationen beschränkt werden.

Alle Kombinationen mit weniger Elementen als die Mindestlänge von Bedingung oder Schlussfolgerung der gesuchten Regeln können je nach gewünschten Interessantheitsmaßen ignoriert werden. Außerdem kann die maximal zu suchende Kombinationslänge beschränkt werden, wenn die Regeln auf eine maximale Anzahl beteiligter Elemente beschränkt sind. Es gilt z.B. für Regeln mit Konfidenz als Interessantheitsmaß die Einschränkungen

1. Minimale Kombinationslänge = Minimum aus minimaler Länge der Bedingung und minimale Länge der Schlussfolgerung

2. Maximale Kombinationslänge = Summe aus maximaler Länge der Bedingung und maximaler Länge der Schlussfolgerung

Mit Hilfe der gefundenen Kombinationen ist die Bildung von Assoziationsregeln möglich. Eine Vielzahl von Interessantheitsmaßen von Assoziationsregeln benötigen zur Berechnung die Häufigkeiten der einzelnen Regelteile. All diese Informationen sind trotz dieser Einschränkungen vorhanden. Mit definierten Grenzen für die Länge der gesuchten Kombinationen kann der Suchraum gezielt reduziert werden.

Inwiefern sich dies in Optimierungen von Algorithmen umsetzen lässt, ist stark von der Arbeitsweise der Verfahren abhängig. ECLAT-basierende Algorithmen durchlaufen eine Baumstruktur ausgehend vom Wurzelknoten, auf APRIORI basierende Verfahren ermitteln rekursiv aus bereits gefundenen Ergebnissen längere Kombinationen. Beide nutzen pro Arbeitsschritt ausgehend von einzelnen Elementen jeweils die aktuell gewonnenen Teilergebnisse, um weitere Kombinationen zu untersuchen. Daher müssen auch die laut einer Regelvorlage nicht benötigten kurzen Kombinationen erzeugt werden. Bei diesen Algorithmen kann somit kein signifikanter Laufzeitgewinn durch Definition einer Mindestkombinationslänge erreicht werden. Der Vorteil des Verzichts auf Speicherung dieser nicht der Vorlage entsprechenden kurzen Kombinationen bleibt jedoch erhalten.

Eine Obergrenze für die Länge einer Kombination kann hingegen genutzt werden, um die Evaluierung weitere Kandidaten bei Erreichen der Maximallänge zu beenden. Der Suchraum wird auf diese Art deutlich eingeschränkt und kann oft auch im ungünstigsten Fall noch vollständig durchsucht werden. Im Umfeld der Analyse von Verkaufsdaten werden oft Kombinationen aus ein bis zwei Elementen pro Bedingung und Schlussfolgerung gewünscht. In diesen Fällen werden maximal vier Elemente, die Vereinigung von Bedingung und Schlussfolgerung, pro Kombination benötigt.

Ohne Längenbegrenzung bilden alle Teilmengen X der Menge aller Elemente E mit $|X| > 0$ möglicher Ergebniskandidaten. Damit ergeben sich maximal $2^{|E|} - 1$ Ergebniskandidaten. Alle Kandidaten gelten hierbei als potenziell häufig und somit als Teil von I. Wird diese Menge auf Kombinationen mit bestimmten Kardinalitäten begrenzt, gilt pro Kardinalität l:

$$|I_l| = \binom{|E|}{l} = \frac{|E|!}{l!(|E| - l)!}$$

Die Kardinalität von I ergibt sich durch die Summe aller gewählten I_l. Für ein Beispiel mit 100 Produkten und gesuchten Kombinationen mit $1 \leq |X| \leq 4$ ergeben sich als vollständiger und maximaler Suchraum ≈ 4 Millionen mögliche Kombinationen. Diese sind im Extremfall berechenbar. Ohne eine Obergrenze für die Kombinationslänge vergrößert sich diese Anzahl auf $2^{100} \approx 10^{30}$ Möglichkeiten, was signifikant mehr Aufwand erzeugt.

Eine weitere mögliche Reduzierung des Suchraumes besteht in der Vorgabe von erlaubten oder zwingend nötigen Elementen in einer Assoziationsregel. Eine solche Einschränkung kann direkt in Form einer Selektion der erlaubten Elemente in dem betrachteten Datenbestand realisiert werden und reduziert diesen entsprechend. Die Vorgabe von erforderlichen Elementen in einer gesuchten Assoziationsregel kann zur Einschränkung der EHK genutzt werden, wenn sowohl für die Bedingung als auch die Schlussfolgerung der Regel zwingende Elemente gegeben sind. Dann kann die EHK auf Kombinationen eingegrenzt werden, welche genau aus einem Teil dieser Elemente bestehen. Kombinationen ohne eines der Elemente werden nicht benötigt und brauchen nicht betrachtet zu werden.

Mit Vorlagen für gesuchte Assoziationsregeln werden vielfältige Optimierungen ermöglicht. Diese sind aber oft sehr stark von den Anforderungen des Anwenders und der weiteren Nutzung der Ergebnisse abhängig. Allgemeingültige Optimierungen der EHK durch Regelvorlagen können die potenziell möglichen Optimierungen unter Beachtung szenariospezifischer Eigenheiten nur bedingt erreichen.

2.2.9. Aufwandsreduktion durch Stichproben

Unter der Annahme, dass bei einer EHK und der Bestimmung von Assoziationsregeln die exakten Häufigkeiten der Kombinationen oft nicht erforderlich sind, können die Ausgangsdaten durch Nutzung einer Stichprobe reduziert werden. Die Auftrittswahrscheinlichkeiten der Kombinationen werden zur Bildung des gewünschten Ergebnisses anhand dieser Stichprobe geschätzt. Dabei können nach Toivonen Intervalle definiert werden, in welchen die ermittelten Häufigkeiten mit dem tatsächlichen Wert bei einer bestimmten Wahrscheinlichkeit übereinstimmen [Toi96]. Toivonen zeigt, dass für eine Stichprobengröße von $\ln(2/\sigma)/(2\mu^2)$, bei $\mu > 0$ und $\sigma < 1$ der Fehler μ mit einer Wahrscheinlichkeit von σ nicht überschritten wird. Die Eigenschaften $\mu = 0,02$ und $1-\sigma = 90\%$ werden beispielsweise ab einer Stichprobengröße von 3.745 Einträgen erreicht. Ein solches Vorgehen ist unabhängig von dem zur EHK genutzten Verfahren und soll daher zur Betrachtung der möglichen Leistungssteigerungen herangezogen werden.

Für die Warenkorbanalyse ist es zwingend nötig, die Stichprobe auf der Granularität einer Transaktion durchzuführen. Werden willkürlich z.B. Zeilen einer Relation entfernt, ändert sich die durchschnittliche Breite einer Transaktion. Diese deutliche Veränderung der Datencharakteristik bedeutet auch signifikante Abweichungen bei den Ergebnissen einer EHK.

Unter der Annahme, dass die Häufigkeitsbestimmung einer Kombination durch das Zählen der beteiligten Transaktionen erfolgt, kann die Gesamtlaufzeit der Suche durch die Multiplikation der Anzahl gefundener Kombinationen mit der durchschnittlichen Zeit zur Bestimmung und Speicherung der Häufigkeit abstrahiert werden. Bei Nutzung von Stich-

proben sollen die geschätzten möglichst den korrekten Ergebnissen entsprechen. Deren Anzahl bleibt unter dieser Annahme nahezu konstant.

Entstehen mit der Verkleinerung des betrachteten Datenbestandes keine signifikanten Änderungen der Datencharakteristiken, verhält sich die Gesamtlaufzeit einer EHK somit linear zum Umfang des Datenbestandes. Das ist darin begründet, dass die Datenaufbereitung linear von der Menge durchschnittlich zu untersuchenden Transaktionen abhängt und bei der EHK die Menge der ermittelten Kombinationen bei konstanten Datencharakteristiken ähnlich bleibt. Die im Durchschnitt pro Kombination gezählten Transaktionen und damit die Laufzeit verringern sich äquivalent zum Umfang der Stichprobe.

Im Falle von auf Kombinationsbäumen basierenden Verfahren wie FP-GROWTH wird die Bestimmung von Kombinationen nicht durch die Verwendung von Stichproben beschleunigt. Hier werden die einzelnen Baumknoten lediglich mit jeweils kleinerem Zähler versehen. Ein geringerer Datenbestand wirkt sich jedoch auf die Zeit zur Konstruktion des Baumes aus. Dabei müssen je nach Umfang der Stichprobe weniger Kombinationen in den Kombinationsbaum eingefügt werden. Im Wesentlichen wird dessen Konstruktion somit linear mit der Verkleinerung der Stichprobe beschleunigt. Ist der Baum bereits vorhanden, sind Stichprobenverfahren nicht nötig.

Die Vorverarbeitung der Daten zur Durchführung einer EHK werden durch Stichproben ebenfalls in einigen Teilschritten reduziert. Für die Nachbearbeitung und Regelerzeugung bestehen keine direkten Auswirkungen. Im Idealfall ist das Ergebnis einer EHK auf einer Stichprobe sehr ähnlich dem korrekten Ergebnis. Die Aufwände der Ergebnisnachbearbeitung bleiben für diesen Fall konstant.

Überwiegt der exponentiell abhängige Laufzeitanteil der Kombinationssuche einer EHK, was in den meisten theoretischen Untersuchungen als Vorbedingung angenommen wird, ist deren Optimierung oft sinnvoller als die lineare Beschleunigung durch Stichprobenverfahren. Die Eliminierung von x möglichen Elementen aufgrund von Kontextwissen, kann die Laufzeit einer EHK um den Faktor 2^x reduzieren, da die Menge der möglichen Kombinationen exponentiell verkleinert wird. Das entspricht z.B. bei $x = 4$ einer Laufzeitverbesserung gemäß der Nutzung einer Stichprobengröße von $6,25\%$ der Ausgangsdaten, welche in diesem Fall meist deutliche Abweichungen in den Häufigkeiten der ermittelten Kombinationen aufweist.

Des Weiteren ergeben sich Probleme, wenn die Stichprobengröße zu klein wird. Die Fehlergröße der Häufigkeiten auf diesen Stichproben wächst nach Haas mit sinkendem Stichprobenumfang [Haa97]. Da die zur Bestimmung häufiger Kombinationen benötigte Mindesthäufigkeit ebenfalls entsprechend sinkt, wirken sich Ungenauigkeiten bei sehr kleinen Stichprobengrößen deutlich auf das ermittelte Ergebnis aus. Die damit bestimmten Häufigkeiten können signifikant vom korrekten Wert abweichen. Der Effekt wird noch verstärkt, wenn absolute Häufigkeiten durch zu geringe Stichproben sehr kleine Werte

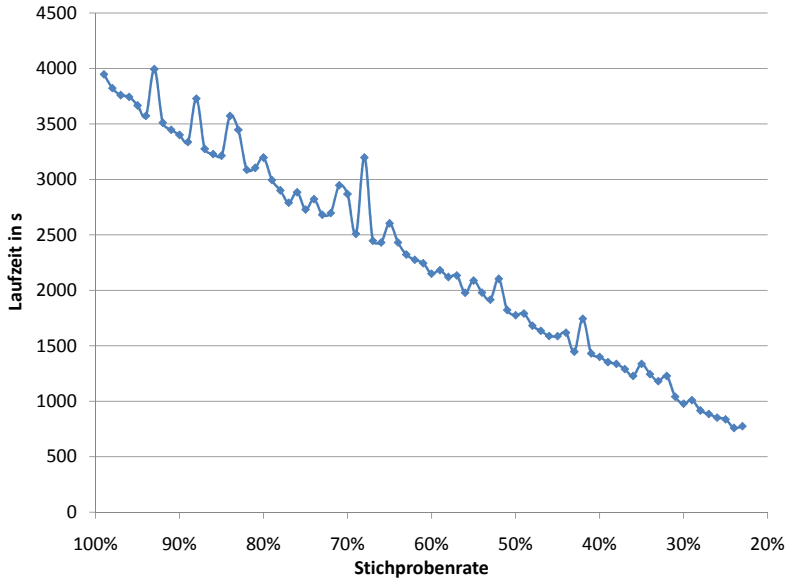


Abbildung 2.7.: Skalierung ECLAT mit veränderter Stichprobengröße

annehmen. Da Häufigkeiten stets ganzzahlig sind, treten hierbei signifikante Rundungsfehler auf.

Abbildung 2.7 zeigt das Laufzeitverhalten einer ECLAT-Implementierung in Abhängigkeit von der Stichprobengröße. Die Darstellung wurde um Seiteneffekte wie Speicherverwaltung und Ergebnisablage bereinigt. Für große Datenmengen ist dieser Laufzeitanteil vernachlässigbar gering. Ebenso wird die nötige Aufbereitung der Daten nicht betrachtet. Diese entsteht für Verfahren mit und ohne Stichproben und kann für EHKs in Verkaufsdaten der dominierende Laufzeitfaktor werden. An dieser Stelle sollen jedoch nur die Auswirkungen auf den eigentlichen EHK-Prozess vorgestellt werden. In der Abbildung ist im Wesentlichen ein linearer Laufzeitgewinn mit sinkender Stichprobengröße ersichtlich. Abweichungen können mit zufälligen Änderungen von Datencharakteristiken begründet werden. Als Beispiel dient der auf Seite 183 näher beschriebene Datenbestand *KundeA2* und als Mindesthäufigkeit ist $0,01\%$ gewählt. In diesem Beispiel mit 1,3 Millionen Transaktionen können nach [Toi96] selbst die seltensten gefunden Kombinationen bei 20% Stichprobenrate noch nahezu exakt ermittelt werden. Eine sinnvolle Stichprobe für dieses Szenario wäre im Bereich von $1\% - 5\%$ anzusiedeln. Abbildung 2.7 soll jedoch lediglich den durch Stichproben zu erwartenden Laufzeitgewinn verdeutlichen.

Während die in diesem Abschnitt vorgestellten Verfahren auf einmal berechneten Stichproben arbeiten, wird durch Chen und Haas eine Optimierung des Stichprobenverfahrens zur EHK unter dem Namen *FAST* vorgeschlagen [CHS02]. Das Ziel dieser Arbeit besteht in der dynamischen Anpassung der Stichproben anhand der Eigenschaften der EHK und erlaubt bei gleicher Stichprobengröße eine durchschnittlich geringere, als in diesem Ab-

schnitt vorgestellte Ergebnisabweichung. Der zu FAST ähnliche Algorithmus EASE wird von Brönnimann vorgeschlagen [BCD+03]. Ein Vergleich von beiden Verfahren bezüglich Laufzeit und Genauigkeit erfolgt in [BCD+04].

Durch Toivonen [Toi96] wird eine weitere Möglichkeit der Nutzung von Stichprobenverfahren zur EHK gezeigt. Mit Hilfe einer Stichprobe werden hier Kandidaten für häufige Kombinationen erzeugt und durch eine direkte Suche im vollständigen Datenbestand deren korrekte Häufigkeit ermittelt. Damit werden Ungenauigkeiten bei den Häufigkeiten der Kombinationen vermieden. Potenziell fälschlicherweise nicht bestimmte Kombinationen werden in einem zweiten Suchlauf ermittelt. Das Ergebnis ist somit vollständig und korrekt.

Insgesamt betrachtet ist die Verwendung von Stichproben eine sinnvolle Optimierung, wenn die Ausgangsdaten sehr umfangreich sind. Der Vorteil besteht in diesem Fall nicht nur in einer entsprechend beschleunigten Auswertung. Wichtig ist hierbei auch meist der verringerte Speicherverbrauch der Verfahren, was die entsprechende Analyse unter Umständen erst ermöglicht. Um die Laufzeit einer EHK zu verbessern ist eine Verfeinerung der Anfragespezifikation einer EHK aber meist wirkungsvoller, da hier durch wenig Anwendungswissen deutlichere Beschleunigungen möglich sind. Zudem werden diese Ergebnisse nicht verfälscht und weisen somit gegenüber einigen stichprobenbasierten EHK-Verfahren eine höhere Qualität auf.

2.3. Aufbereitung von Assoziationsregeln

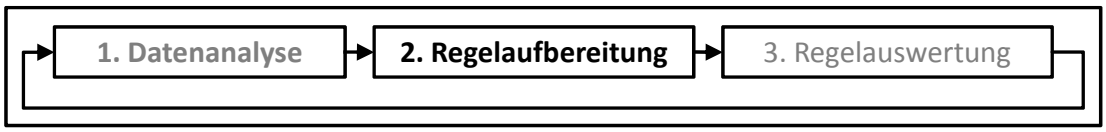


Abbildung 2.8.: Zweiter Schritt der Assoziationsregelsuche

Mit Hilfe der in den vorherigen Abschnitten dargestellten Verfahren können die aus einem Datenbestand extrahierten Assoziationsregeln bereits oft deutlich eingeschränkt werden, da lediglich Kombinationen betrachtet werden, welche eine Mindesthäufigkeit aufweisen. Gemäß Abbildung 2.8 folgt nach der Datenanalyse die Regelaufbereitung. Dieser Schritt bewertet die ermittelten Assoziationsregeln und soll eine verbesserte Verwendbarkeit durch den Anwender ermöglichen. Hierfür werden im folgenden Abschnitt Ansätze zur Beurteilung von Assoziationsregeln in Form von Eigenschaften sinnvoller Regeln spezifiziert.

Abbildung 2.9 zeigt in einer Übersicht, welchen Maßstäben ein Interessantheitsmaß nach Einschätzung verschiedener Autoren genügen soll und wie diese kategorisiert werden. Sil-

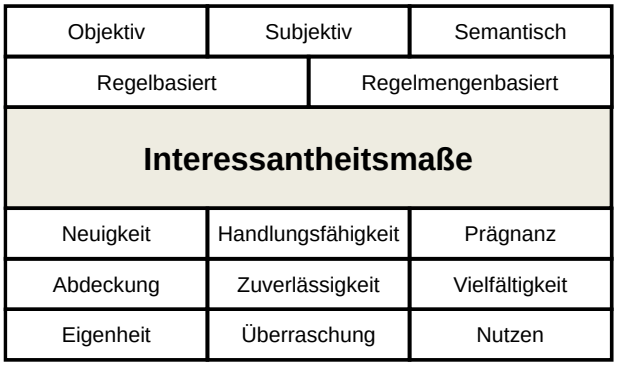


Abbildung 2.9.: Einteilungsmöglichkeiten von Interessantheitsmaßen

berschatz [ST96] fordert lediglich die beiden Punkte *Neuigkeit* und *Handlungsfähigkeit* als wichtige Eigenschaften von Assoziationsregeln. Von Hamilton [GH06] werden auch die Anforderungen *Prägnanz*, *Abdeckung*, *Zuverlässigkeit*, *Vielfältigkeit*, *Eigenheit*, *Überraschung* und *Nutzen* als wichtig erachtet.

- **Neuigkeit:** Bietet eine Regel einem Nutzer eine bisher unbekannte Information, besitzt sie Neuigkeit. Diese Eigenschaft ist nur schwer messbar, da das Wissen eines Nutzers oft lediglich in Teilen dargestellt wird und damit das Fehlen von Wissen nicht immer erkannt werden kann.

- **Handlungsfähigkeit:** Eine Regel ist handlungsfähig, wenn durch sie Entscheidungen innerhalb einer Domäne getroffen werden können. Aus einer Regel sollte sich unmittelbar eine nützliche Aktion ableiten lassen. Beispielsweise kann die Analyse verkaufter Produkte zu Bewerbung oder Bündelung von Produkten führen. Dafür müssen Äquivalenzklassen erstellt werden, welchen Handlungen zugewiesen werden. Jede Regel resultiert dann entsprechend der ihr zugewiesenen Äquivalenzklasse in vordefinierten Aktionen. Der Suchraum ist jedoch meist nicht vollständig definiert, wodurch die Zuweisung von Aktionen zu Äquivalenzklassen unterbunden wird.

- **Prägnanz:** Eine Regel, aus welcher sich einfach Informationen und Aktionen erstellen lassen, wird als prägnant bezeichnet. Ist eine Regel komplex, wird dieser Vorgang erschwert. Dabei bezieht sich die Komplexität meist auf die Anzahl der Elemente in Bedingung und Schlussfolgerung einer Regel oder die Gruppengröße einer Menge gleichartiger Regeln. Je einfacher eine Regel ist, umso prägnanter wird sie.

- **Abdeckung:** Eine Regel wird wichtiger, je öfters sie bzw. eine Regelgruppe in einem Datenbestand zu finden ist. Ein häufiges Auftreten einer Regel muss jedoch differenziert betrachtet werden, da dies oft lediglich offensichtliche Zusammenhänge aufzeigt.

- **Zuverlässigkeit:** Kann eine Regel in mehreren Variationen beobachtet werden, ist diese zuverlässig.

- **Vielfältigkeit:** Eine Regel ist vielfältig, wenn die beteiligten Elemente sich untereinander deutlich unterscheiden. Falls die einzelnen Regeln einer Regelgruppe verschieden sind, ist diese Gruppe wiederum vielfältig.

- **Eigenheit:** Eine Regel besitzt Eigenheit, wenn sie nach einer definierten Distanzmetrik keine ähnlichen Nachbarregeln aufweist. Oft werden solche Regeln auch als Ausreißer (engl. outlier) bezeichnet und wurden in vorherigen Untersuchungen nicht beachtet. Dies kann ihre Interessantheit steigern.

- **Überraschung:** Als überraschend gelten Regeln, wenn sie den Erwartungen und dem bereits bekannten Wissen des Nutzers widersprechen oder im Widerspruch zu anderen Regeln stehen. Überraschende Regeln sind interessant, da sie neues Wissen für einen Anwender darstellen.

- **Nutzen:** Eine Regel ist von Nutzen, wenn ein Anwender aus dieser Regel Informationen und Aktionen zur Erreichung eines Zieles ableiten kann. Der Nutzen ist von der Sicht des Anwenders abhängig und damit anwendungsspezifisch.

Ausgehend von diesen Eigenschaften, lassen sich Interessantheitsmaße in drei Kategorien unterteilen.

1. **Objektive Interessantheitsmaße:** Objektive Interessantheitsmaße sind unabhängig von der Sichtweise eines Nutzers. Sie sind vollständig aus dem betrachteten Datenbestand ableitbar und nur von diesem abhängig. Die zugrunde liegenden Betrachtungen basieren meist auf mathematischen Forschungsgebieten, wie Statistik, Wahrscheinlichkeitsrechnung und Informationstheorie. Die Interpretation solcher Interessantheitsmaße erweist sich für nicht spezialisierte Anwender oft als schwierig. Derartige Interessantheitsmaße sind häufig zueinander korreliert.

2. **Subjektive Interessantheitsmaße:** Diese Maße beziehen zusätzlich zu den Daten auch kontextbezogenes Wissen des Anwenders ein. Dies kann entweder durch Interaktion oder durch eine explizite Repräsentation des Wissens über die Anwendungsdomäne erfolgen. Da der direkte Bezug zu dem vorhandenen Wissen des Nutzers betrachtet wird, ist die durch eine Regel repräsentierte Information dem Anwender oft direkt ersichtlich. Ein Vergleich von Regeln aus verschiedenen Datenbeständen ist bei subjektiven Interessantheitsmaßen nicht immer möglich, da diese stark von dem jeweiligen Szenario abhängen.

3. **Semantische Interessantheitsmaße:** Semantische Interessantheitsmaße beurteilen die Bedeutung von Regeln. Die Eigenschaften *Handlungsfähigkeit* und *Nutzen* werden als semantische Maße bezeichnet. Anders als bei subjektiver Interessantheit,

welche gefundenes mit bestehendem Wissen vergleicht, versuchen semantische Maße, das Optimum einer Nutzenfunktion zu erreichen.

Interessantheitsmaße sind ebenfalls in *regelbasierte* und *regelmengenbasierte* Maße einteilbar. Regelmengenbasierte Interessantheitsmaße beziehen sich beispielsweise auf die Anzahl unterschiedlicher Kombinationselemente oder die Anzahl unterschiedlicher beteiligter Kombinationen einer Regelgruppe. Ebenso können komplexere Eigenschaften, wie die Anzahl der Ausreißer pro Gruppe oder syntaktische Nachbarschaften zu anderen Regelgruppen als Maßzahl dienen. Regelbasierte Interessantheitsmaße beziehen sich nur auf Eigenschaften der betrachteten Regel und nicht auf deren Umfeld. Diese Maße sind mit Hilfe ihrer in dem nachfolgenden Abschnitt vorgestellten Kontingenz- bzw. Vierfeldermatrix berechenbar.

Im Rahmen dieser Arbeit sollen vorrangig objektive Interessantheitsmaße betrachtet werden, da eine allgemein gültige Lösung mit möglichst wenig Nutzerinteraktion und Nutzervorgaben angestrebt wird. Objektive Interessantheitsmaße erfüllen diese Anforderungen und bilden daher die Grundlage der Lösungen dieser Arbeit. Der folgende Abschnitt stellt eine Auswahl dieser Maße detailliert vor.

Die objektiven Interessantheitsmaße von Assoziationsregeln repräsentieren statistische Merkmale des zugrunde liegenden Datenbestandes. Sie nutzen die Häufigkeiten der einzelnen Teile einer Regel in dem betrachteten Datenbestand und deren Beziehungen zueinander. Mit Hilfe von objektiven Interessantheitsmaßen der Assoziationsregeln kann ohne Interaktion des Anwenders eine Begrenzung, Filterung, Bewertung und Sortierung der Regeln automatisiert erfolgen.

2.3.1. Repräsentation als Kontingenzmatrix

Zur Berechnung einer Vielzahl objektiver Interessantheitsmaße wird eine 2×2 Kontingenzmatrix genutzt, welche die benötigten Häufigkeitswerte zur Beurteilung einer Assoziationsregel enthält. Tabelle 2.2 zeigt eine solche Kontingenzmatrix für eine Regel $A \Rightarrow C$. K ist dabei die Kardinalität von DB. Die Felder der Matrix repräsentieren die Kombinationen von Bedingung und Schlussfolgerung der Regel. Der Eintrag der Spalte A und Zeile C mit $|A \cup C|$ enthält beispielsweise die Anzahl der Transaktionen, welche sowohl die Bedingung A als auch die Schlussfolgerung C enthalten. Dies entspricht der Häufigkeit der Regel $A \Rightarrow C$. Der Ausdruck $|A \cup \neg C|$ zeigt die Anzahl der Transaktionen, die A, jedoch nicht C enthalten. Die Summe aller vier Felder ergibt die Menge aller Transaktionen K. Sowohl A als auch C sind Kombinationen von Elemente und können somit beliebig viele Einträge enthalten. Dabei muss beachtet werden, dass sämtlichen Einträge der Matrix absolute Häufigkeiten bezeichnet. Mit Hilfe von K sind relative Werte jedoch leicht bestimmbar.

	A	$\neg A$	
C	$\|A \cup C\|$	$\|\neg A \cup C\|$	$\|C\|$
$\neg C$	$\|A \cup \neg C\|$	$\|\neg A \cup \neg C\|$	$\|\neg C\|$
	$\|A\|$	$\|\neg A\|$	K

Tabelle 2.2.: Kontingenzmatrix zu der Assoziationsregel $A \Rightarrow C$

	Butter, Milch	¬ *(Butter, Milch)*	
(Ei, Mehl)	10	60	70
¬ *(Ei, Mehl)*	15	100	115
	25	160	185

Tabelle 2.3.: Beispiel: Kontingenzmatrix

Tabelle 2.3 zeigt ein Beispiel für die Regel *(Butter, Milch)* ⇒ *(Ei, Mehl)*. Dabei treten in einem Datenbestand mit 185 Transaktionen *(Butter, Milch, Ei, Mehl)* zehn mal auf. *(Butter, Milch)* insgesamt 25 mal. *(Ei, Mehl)* ist 70 mal vorhanden, was 60 Vorkommen ohne *(Butter, Milch)* entspricht. Insgesamt 100 Transaktionen enthalten weder *(Butter, Milch)* noch *(Ei, Mehl)*.

Die folgenden Erläuterungen einiger wichtiger objektiver Interessantheitsmaße für Assoziationsregeln können mit Hilfe der Kontingenzmatrix ermittelt werden. Der Ausdruck $p(X)$ stellt im Folgenden äquivalent zu $support(X)$ die relative Häufigkeit einer Kombination X im Verhältnis zu K bzw. $|DB|$ dar.

2.3.2. Regelhäufigkeit

Die *Häufigkeit* einer Regel $A \Rightarrow C$ bezogen auf insgesamt K Transaktionen besagt, in wie vielen Transaktionen die Kombination $A \cup C$ auftritt. Dieses Interessantheitsmaß kann zum Begrenzen des Suchraumes von häufigen Kombinationen genutzt werden. Durch die Wahl einer Mindestregelhäufigkeit kann bei der Suche nach relevanten Kombinationen eine Untergrenze für die Häufigkeit einer Kombination angegeben werden. Die Regelhäufigkeit ist außerdem im Allgemeinen für Anwender leicht verständlich. Sie wird wie folgt definiert:

$$ruleSupport(A \Rightarrow C) = support(A \cup C)$$

Eine alleinige Betrachtung dieses Interessantheitsmaßes zur Bewertung einer Regel ist oft unzureichend.

1. Seltene Kombinationen oder Regeln werden nicht betrachtet, obwohl diese wichtig sein können.

2. Selten auftretende Elemente mit möglicherweise hoher Wichtigkeit werden ignoriert.

3. Überproportional häufig verkaufte Produkte sind oft wenige relevant oder bereits anderweitig betrachtet. Diese Produkte sind jedoch oft Teil von Regeln mit hoher Regelhäufigkeit.

Die Regelhäufigkeit sollte entsprechend zur Bewertung von Assoziationsregeln nur in Kombination mit weiteren Maßen genutzt werden. Sie stellt jedoch eine wichtige grundlegende Wertung dar, welche vor allem bei sehr geringer Häufigkeit ein wichtiges Ausschlusskriterium für eine Regel sein kann.

2.3.3. Konfidenz

Die *Konfidenz* einer Regel stellt die bedingte Wahrscheinlichkeit der Schlussfolgerung bei gegebener Bedingung dar. Sie wird wie folgt definiert:

$$conf(A \Rightarrow C) = \frac{support(A \cup C)}{support(A)}$$

Da die Konfidenz die Häufigkeit der Schlussfolgerung C nicht direkt betrachtet, bleibt sie nach [GGP98] unter folgenden Bedingungen konstant:

1. $support(A)$ und $support(A \cup C)$ sind konstant und $support(C)$ variiert,

2. $support(A)$, $support(A \cup C)$ und $support(C)$ bleiben konstant, $support(\neg A \cup \neg C)$ verändert sich,

3. $support(A)$ und $support(A \cup C)$ ändern sich proportional.

Regeln mit um Größenordnungen geringerer oder größerer Häufigkeit erreichen somit beispielsweise die jeweils gleiche Konfidenz. Sie verhält sich wie die Regelhäufigkeit antimonoton und kann zur effizienten Begrenzung zu erzeugender Regeln genutzt werden.

Die Konfidenz wird häufig in praktischen Szenarien als Kriterium gefordert und ist in ihrer Aussage einem Anwender meist leicht verständlich. Eine Konfidenz von 100% stellt den maximal erreichbaren Wert dar. Eine sehr hohe Konfidenz zeigt jedoch funktionale Abhängigkeiten zwischen Bedingung und Schlussfolgerung. Diese sind im Falle einer Assoziationsregel oft negativ zu bewerten oder bekannt. Der hohe Wert der Konfidenz kann entsprechend falsch gedeutet werden oder wichtige Regeln mit sinnvollen Konfidenzwerten verdecken. Die Häufigkeit der Schlussfolgerung fließt nicht direkt in die Konfidenz einer Regel ein, wodurch ebenfalls Fehldeutungen ermöglicht werden.

2.3.4. Lift

Der *Lift* (oder *Interest* nach Brin, Motwani und Silverstein [BMS97]) stellt das Verhältnis zwischen unabhängiger Wahrscheinlichkeit und tatsächlicher Wahrscheinlichkeit einer Regel dar. Diese Kennzahl erweitert die Konfidenz um eine Abhängigkeit zur Häufigkeit der Schlussfolgerung. Dadurch entfällt je nach Betrachtungsweise ein Vorteil oder eine Schwäche der Konfidenz. Der Lift wird folgendermaßen definiert:

$$lift(A \Rightarrow C) = \frac{support(A \cup C)}{support(A) \cdot support(C)} = \frac{conf(A \Rightarrow C)}{support(C)}$$

Der Lift zeigt, wie viel häufiger eine Regel im Verhältnis zu ihrem statistisch unabhängigen zu erwartenden Auftreten in einem Datenbestand vorkommt. Ist der Wert $0 \leq lift(A \Rightarrow C) \leq 1$, korrelieren Bedingung A und Schlussfolgerung C negativ. Die positive Korrelation, bei $lift(A \Rightarrow C) > 1$, steigt linear mit dem Wert von $lift(A \Rightarrow C)$.

Ein Problem beim Lift als Interessantheitsmaß besteht in seinem Wertebereich. Während negative Korrelationen auf Werte zwischen $0 \ldots 1$ abgebildet werden, können positive Korrelationen Werte von 1 bis K annehmen. Bei gleicher Wertigkeit von negativen und positiven Korrelationen müssen $lift(A \Rightarrow C) < 0$ durch Berechnung des jeweiligen Reziproken ineinander überführt werden.

Der variable Wertebereich und die unterschiedliche Wertung negativer und positiver Korrelationen erschwert eine Deutung. Der Wertebereich des Lifts ist abhängig von K. Regeln können anhand ihres Lifts somit nur bedingt szenarioübergreifend verglichen werden.

2.3.5. Überzeugung

Von Brin, Motwani, Ullman und Tsur [BMUT97] wird ein Interessantheitsmaß namens *Überzeugung* (engl. conviction) eingeführt, welches die Nachteile der Interessantheitsmaße Konfidenz und Lift beheben soll. Der Ausdruck $A \Rightarrow C$ wird äquivalent auch durch $\neg(A \Rightarrow \neg C)$ dargestellt. Damit kann das Verhältnis vom abhängigen zum unabhängigen Fall betrachtet werden. Die Überzeugung wird wie folgt definiert:

$$conv(A \Rightarrow C) = \frac{support(A) \cdot support(\neg C)}{support(A \cup \neg C)}$$

Sind Bedingung und Schlussfolgerung nicht korreliert, liefert $conv(A \Rightarrow C) = 1$ zurück. Besteht eine vollständige Korrelation, liefert die Überzeugung ihren Maximalwert ∞. Der Einfluss von $support(C)$ zeigt eine Implikation und nicht ausschließlich die Korrelation zwischen Bedingung und Schlussfolgerung, wie es beim Lift der Fall ist. Die oft unerwünschte Beständigkeit der Konfidenz gegen Veränderungen von einigen der ihr zugrunde liegenden Häufigkeiten wird durch die Integration von $support(A)$ und $support(C)$ aufgelöst.

2.3.6. Chi-Quadrat-Test und Pearson-Koeffizient

Der *Chi-Quadrat*-Test erlaubt Aussagen über die statistische Unabhängigkeit von Merkmalen und kann somit die Signifikanz einer Regel feststellen. Der Chi-Quadrat-Hypothesentest ist ursprünglich ein Werkzeug der Statistik, um Abhängigkeiten zwischen Daten zu zeigen. Die Verwendung zum Erkennen der Abhängigkeit von Bedingung und Schlussfolgerung einer Regel ist damit ebenfalls möglich [BMS97; BP99; MFM+98]. Als signifikant gilt hierbei eine Aussage, wenn eine gegebene Verteilung mit großer Wahrscheinlichkeit nicht zufällig entstanden ist. Die zugehörige formale Darstellung lautet wie folgt ($p = support$):

$$\frac{\chi^2(A \Rightarrow C)}{K} = \frac{(|A \cup C| - |A| \cdot |C|/K)^2}{|A| \cdot |C|} + \frac{(|A \cup \neg C| - |A| \cdot |\neg C|/K)^2}{|A| \cdot |\neg C|} + \frac{(|\neg A \cup C| - |\neg A| \cdot |C|/K)^2}{|\neg A| \cdot |C|} + \frac{(|\neg A \cup \neg C| - |\neg A| \cdot |\neg C|/K)^2}{|\neg A| \cdot |\neg C|}$$

Für jedes Feld der Kontingenzmatrix wird die Differenz zwischen der erwarteten Wahrscheinlichkeit und der tatsächlich Wahrscheinlichkeit quadriert. Dieser Wert wird durch die erwartete Wahrscheinlichkeit dividiert. Durch Summieren dieser Einzelwerte ergibt sich der Chi-Quadrat-Gesamtwert χ^2 einer Assoziationsregel.

Die Interpretation von χ^2 ist abhängig von einem vorher definierten Signifikanzniveau α. Dieses stellt eine maximale Irrtumswahrscheinlichkeit dar. Entsprechend bildet $1 - \alpha$ die Vertrauenswahrscheinlichkeit. Der Wert für $\chi^2(A \Rightarrow C)$ kann mit einer Signifikanztabelle (Ausschnitt siehe Tabelle 2.4) in Abhängigkeit von α verglichen werden. Liegt χ^2 über dem entsprechenden Wert in der Signifikanztabelle, gelten A und C als signifikant linear abhängig und damit als korreliert.

$1 - \alpha$	0,900	0,950	0,975	0,990	0,995	0,999
	2,71	3,84	5,02	6,63	7,88	10,83

Tabelle 2.4.: Ausschnitt Signifikanztabelle (erster Freiheitsgrad)

Eine Erweiterung des Chi-Quadrat-Tests stellt der *Pearson-Koeffizient* (oder auch Φ-Korrelation) dar. Dieser bestimmt die Stärke der Assoziation zwischen A und C. Der Wert berechnet sich wie folgt [GH06]:

$$pearson(A \Rightarrow C) = \frac{p(A \cup C) - p(A) \cdot p(C)}{\sqrt{p(A) \cdot p(C) \cdot p(\neg A) \cdot p(\neg C)}}$$

Der Pearson-Koeffizient besitzt einen Wertebereich von $-1 \leq pearson(A \Rightarrow C) \leq 1$, wobei -1 eine vollständig negative Korrelation, 1 eine vollständig positive Korrelation und 0 das Fehlen einer Korrelation ausdrückt.

2.3.7. Minimale Verbesserung

Die *minimale Verbesserung* (engl. minimum improvement) erläutert die Verbesserung einer Regel Y_1 gegenüber einer einfacheren Regel Y_2. Sie sagt aus, wie sich ein Interessantheitsmaß durch die Hinzufügen weiterer Elemente ändert. Vorgeschlagen wurde sie von Bayardo, Agrawal und Gunopulos [BAG99]. Dieses Interessantheitsmaß ist nicht allein aus der Kontingenzmatrix ableitbar, sondern benötigt zusätzlich die Häufigkeiten der Teilkombinationen von A. Die zugehörige Formel lautet:

$$impr(A \Rightarrow C) = min(\{conf(A \Rightarrow C) - conf(A' \Rightarrow C) | A' \subset A, A' \neq \emptyset\})$$

Ist $impr(A \Rightarrow C) > 0$, bewirkt eine Verkürzung der Regelbedingung eine Verschlechterung der Konfidenz. Daraus folgt, dass jedes Element der Bedingung wichtig für diese Regel ist. Ist der Wert negativ, besitzt eine Teilkombination der Regelbedingung eine stärkere Aussagekraft als die Regelbedingung selbst. Die Aussage der Regel $A \Rightarrow C$ wird damit durch eine andere Regel mit kürzerer Bedingung besser repräsentiert.

2.3.8. Zusammenfassung

Unter anderem in den Arbeiten von Hamilton und Tan [GH06; HH03; TKS02; TSK06b] werden Übersichten über eine Vielzahl statistische Maße vorgestellt. Tabelle 2.5 zeigt in Anlehnung an diese Arbeiten eine Übersicht einiger populärer Interessantheitsmaße. Zur Ergänzung der in den vorherigen Abschnitten eingeführten häufig verwendeten Maße werden nachfolgend noch weniger populäre Maße kurz vorgestellt. Zur besseren Übersicht wird wiederum $p = support$ verwendet.

Von Shapiro wurde in [PSF92] ein Interessantheitsmaß eingeführt, welches nach [LFZ99] auch als *Novelty* bezeichnet wird. Es zeigt die Differenz zwischen der erwarteten und der beobachteten Wahrscheinlichkeit einer Assoziationsregel. Je größer die Differenz ist, desto überraschender ist das Auftreten dieser Assoziation. Für negative Werte bis $-0,25$ besteht ein negativer Zusammenhang zwischen A und C. Der Fall tritt ein, wenn A und C jeweils in der Hälfte aller Transaktionen auftreten, aber nicht gemeinsam. A und C schließen sich somit maximal gegenseitig aus. Ist Novelty$= 0,25$, besteht eine perfekte Abhängigkeit zwischen A und C. Der Wert tritt für $p(A) = p(C) = p(A \cup C) = 50\%$ ein. Bedingung und Schlussfolgerung sind ohne Auswirkungen auf den Wert der Novelty vertauschbar. Novelty weist insgesamt große Ähnlichkeiten zum Maß Lift auf, bietet aber einen festen Wertebereich.

Mit *Change of Support* wird von Yao die Differenz zwischen den Wahrscheinlichkeiten von C und A berechnet [YZ99]. Der ebenfalls von Yao vorgestellte *Mutual Support* misst die Stärke, mit der die Bedingung A ausschließlich die Konklusion C verursacht. *Odds Ratio* ist eine aus der Statistik bekannte Größe und berechnet den Quotienten zwischen

Name	**Formel**	**Werte-bereich**
Häufigkeit	$p(A \cup C)$	$0 \ldots 1$
Konfidenz	$p(C\|A) = p(A \cup C)/p(A)$	$0 \ldots 1$
Lift	$p(A \cup C)/(p(A)p(C))$	$0 \ldots 1 \ldots K$
Überzeugung	$p(A \cup \neg C)/(p(A)p(\neg C))$	$0 \ldots \infty$
Shapiro / Novelty	$p(A \cup C) - p(A)p(C)$	-0,25...0,25
J-Measure	$p(C\|A)\log(\frac{p(C\|A)}{p(C)}) + p(\neg C\|A)\log(\frac{p(\neg C\|A)}{p(\neg C)})$	$0 \ldots 1$
Prevelance	$p(C)$	$0 \ldots 1$
Coverage	$p(A)$	$0 \ldots 1$
Change of Support	$p(C\|A) - p(C)$	$0 \ldots 1$
Mutual Support	$p(A \cup C)/(p(A) + p(C) + p(A \cup C))$	$0 \ldots 1$
Φ-Correlation	$\frac{p(A \cup C) - p(A)p(C)}{\sqrt{p(A)p(C)p(\neg A)p(\neg C)}}$	-1...0...1
Odds Ratio	$(p(A \cup C)p(\neg A \cup \neg C))/(p(\neg A \cup C)p(A \cup \neg C))$	$0 \ldots 1 \ldots \infty$
Recall	$p(A\|C) = p(A \cup C)/p(C)$	$0 \ldots 1$
Certainty Factor	$(p(C\|A) - p(C))/p(\neg C)$	-1...0...1
Yule's Q	$\frac{p(A \cup C)p(\neg A \cup \neg C) - p(A \cup \neg C)p(\neg A \cup C)}{p(A \cup C)p(\neg A \cup \neg C) + p(A \cup \neg C)p(\neg A \cup C)}$	-1...0...1
Relative Risk	$p(C\|A)/p(C\|\neg A)$	$0 \ldots 1$
Accuracy	$p(A \cup C) + p(\neg A)p(\neg C)$	$0 \ldots 1$
Minimum Improvement	$\min(\{conf(A \Rightarrow C) - conf(A' \Rightarrow C) \| A' \subset A, A' \neq \emptyset\})$	-1...0...1
All-Confidence	$p(A \cup C)/\max\{p(A') \| A' \subset (A \cup C), A' \neq \emptyset\}$	$0 \ldots 1$

Tabelle 2.5.: Übersicht Interessantheitsmaße

der Wahrscheinlichkeit von C mit und ohne A. Das durch Smyth und Goodman [SG92] vorgestellte Maß *J-Measure* stammt aus der Informationstheorie. Damit wird der Informationsgehalt einer Regel bestimmt, welcher durch den Vergleich der Wahrscheinlichkeit eines Ereignisses vor und nach Eintreten errechnet wird.

Die Maße *Coverage* und *Prevalence* bestimmen jeweils die Wahrscheinlichkeit der Bedingung bzw. Schlussfolgerung einer Regel. Sie entsprechen somit direkt $support(A)$ bzw. $support(C)$. Ein Maß aus der deskriptiven Statistik wird als *Relative Risk* bezeichnet. Hier wird das Verhältnis zwischen der Wahrscheinlichkeit von C bei gegebenem und fehlendem A betrachtet.

Der *Certainty Factor* wurde von Shortliffe und Buchanan [SB75] zur Messung der Ungewissheit von Regeln vorgeschlagen. Ursprünglich bezog sich dieses Maß auf ein medizinisches Expertensystem. Certainty Factor ist ein Maß für die Veränderung der Wahrscheinlichkeit $support(C)$ unter der Bedingung A. Ein positiver Wert zeigt dabei die Reduktion der Wahrscheinlichkeit, dass C in einer Transaktion gemeinsam mit A enthalten ist. *Yu-*

le's Q ist ein weiteres aus der Statistik bekanntes Maß, dessen Verhalten dem von *Odds Ratio* ähnelt. Yule's Q erreicht Werte zwischen $-1 \ldots 1$. Der Wert -1 stellt hierbei eine perfekt negative Korrelation dar, 1 eine perfekt positive Korrelation und 0 die statistische Unabhängigkeit zwischen A und C.

Alle Interessantheitsmaße betrachten den Wert einer Regel jeweils aus verschiedenen Sichtweisen. Während beispielsweise die Regelhäufigkeit oft ein wichtiges Maß darstellt, muss dies nicht zwangsläufig immer gelten. Je nach Szenario und Suchziel muss ein Anwender somit jeweils passende Maße wählen und gegeneinander gewichten. Um dies zu erreichen, kann eine grafische Aufbereitung hilfreich sein.

2.4. Darstellung von Assoziationsregeln

Abbildung 2.10.: Dritter Schritt der Assoziationsregelsuche

Nachdem vorherige Abschnitte bereits erläutert haben, wie wichtige Informationen aus untersuchten Daten extrahiert und maschinell bewertet werden können, soll dieser Abschnitt den in Abbildung 2.10 ersichtlichen letzten Schritt der manuellen Wissensextraktion aus den Assoziationsregeln beleuchten.

Der Fokus dieser Arbeit liegt auf den ersten zwei Schritten, da diese größtenteils technischer Natur sind. Die Regelauswertung umfasst in weiten Teilen psychologisch und kognitiv orientierte Prozess durch den Anwender. Der Bezug zu einer technischen Implementierung ist gering und soll daher nur im Überblick erläutert werden.

Die Möglichkeiten zur Analyse von Assoziationsregeln sind vielseitig und umfassen neben der Nutzung und Anpassung von OLAP-Technologien [LZBX06] vor allem die Verwendung von intuitiven und nach persönlichen Erfahrung gesteuerten manuellen Interaktionen. Der Vorteil eines Anwenders gegenüber automatischen Auswertungen ist dessen Wissen über Syntax und Semantik der untersuchten Daten. Dieser kann nur bedingt durch technische Lösungen erreicht werden. Außerdem ist bei der Suche nach interessanten Informationen das Ziel nicht immer exakt im Vorfeld spezifizierbar. Es ergibt sich vielmehr interaktiv im Verlauf der Suche, durch Auffälligkeiten in einer grafischen Aufbereitung oder im Vergleich zur Erwartungshaltung des Anwenders.

Das Kontextwissen und Annahmen eines Anwenders über die Informationen in den Daten können durch die Komplexität dieser Angaben und ohne Einschränkungen auf die be-

trachtete Domäne nur bedingt in der automatischen Vorverarbeitung verwendet werden. Die Nutzung der Fähigkeiten eines menschlichen Betrachters ist somit oft unumgänglich, um subjektiv wichtige Informationen zu erkennen.

In den ersten beiden Schritten einer Analyse erfolgte bereits eine signifikante Reduktion der Ausgangsdaten auf eine für einen Nutzer verwertbaren Menge an Assoziationsregeln. Geht man von der maximalen Menge aller möglichen Kombinationen und Regeln aus, kann durch die Definition einer Mindesthäufigkeit bereits eine Vielzahl an vermutlich uninteressanten Kombinationen entfernt werden, was die daraus erzeugten Assoziationsregeln ebenfalls reduziert. Diese Kombinationen weisen aufgrund ihres seltenen Auftretens für den Nutzer meist eine geringe Relevanz auf. Sind bei $|E|$ betrachteten Elementen $2^{|E|} - 1$ Elementkombinationen möglich, können diese mit Hilfe einer Mindesthäufigkeit signifikant reduziert werden. Die dabei entstehende Menge von Kombinationen kann dabei bereits wichtige Informationen verloren haben. Unter der Annahme, dass der Wert einer Regel stark von der Häufigkeit ihrer Einzelkomponenten anhängt, ist diese Einschränkung aber meist sinnvoll und tragbar.

Aus derartig erzeugten Kombinationen kann wiederum ein Vielfaches an Regeln erzeugt werden, da sämtliche Teilkombinationen $X' \subset X$ einer betrachteten Kombination X als Bedingung einer Regel genutzt werden können. Viele dieser Regeln sind redundant, trivial, unwichtig oder liegen nicht im Fokus des Anwenders. Durch eine automatische Aufbereitung und Evaluierung nach statistischen Merkmalen ist der potenzielle Wert einer Regel erkennbar und erlaubt eine Vorsortierung nach den wahrscheinlich für den Betrachter interessantesten Aussagen. Auch in diesem Arbeitsschritt besteht die Gefahr, wichtige Informationen im Rahmen einer automatischen Filterung zu verlieren. Als Einstiegspunkt für eine manuelle Evaluation der Regelmenge ist dieser Schritt jedoch sinnvoll, um einem Anwender die Erfassung der Informationen zu erleichtern. Nach einer automatischen Aufbereitung ermittelter Regeln kann die dem Nutzer präsentierte Regelmenge auf einige hundert Regelgruppen oder -repräsentanten reduziert werden.

Aus der Wahrnehmungspsychologie ist bekannt, dass der Mensch lediglich je 7 ± 2 Einheiten (engl. chunks) gleichzeitig erfassen und verarbeiten kann [Mil56]. Eine Einheit entspricht dabei einem atomaren Informationsobjekt, z.B. einer Zahl, einem Buchstaben oder einer kurzen Kombination. Eine direkte Darstellung hunderter Assoziationsregeln ist damit für einen Nutzer nur schwer verwertbar. Zumindest das Erkennen von Ähnlichkeiten und Verknüpfungen zwischen den einzelnen Elementen ist nur bedingt zu erreichen.

Ohne das spezifische Domänenwissen des Anwenders, sowie dessen kognitiver Fähigkeit und Erfahrung ist das Finden relevanter Assoziationsregeln nicht möglich. Das Ziel ist demnach, einem Nutzer die effiziente Navigation und Fokussierung auf die für ihn subjektiv wichtigen Erkenntnisse zu ermöglichen.

Es bestehen zur Lösung dieser Problematik Ansätze zur Repräsentation der Regeln mittels einer Menge visueller Formen (geometrische Gebilde) und deren Eigenschaften (Far-

be, Form, Neigungswinkel, Größe) [CMS99]. Außerdem wurde beispielsweise die Einbeziehung der räumlichen Wahrnehmung des Menschen als Erkennungshilfe untersucht [AZJ01].

Die folgenden Teilabschnitte zeigen die in der Praxis oft verwendeten Ansätze syntaktischer oder semantischer Regelaufbereitung. Hierbei bedeutet der Syntax einer Regel deren Inhalt, d.h. die darin enthaltenen Elemente. Semantik wird in diesem Zusammenhang mit den Eigenschaften der Regelteile bzw. der Elemente verknüpft. Semantische Eigenschaft sind beispielsweise objektive Interessantheitsmaße. Sowohl für syntaktische als auch semantische Darstellungen sind eine Vielzahl möglicher Varianten vorhanden, die jeweils als ein Blickwinkel oder eine Sicht auf die Daten definiert werden sollen. Für den Nutzer sinnvolle Information liegt dabei oft verdeckt in einer Gruppe von Sichten und erst eine Kombination ermöglicht die Erlangung von Wissen.

Daher wird in dieser Arbeit eine parallele Darstellung mehrere Sichten vorgeschlagen, welche die jeweils in einer Ansicht gewählten Einträge und Regeln in den anderen verfügbaren Darstellungen optisch hervorhebt. Auf diesen kann eine gezielte mehrdimensionale Einschränkung der Ausgangsregelmenge auf einige dem Nutzer wichtige Regeln erfolgen. Die Regeln können anschließend einzeln auf ihren praktischen Wert untersucht werden.

2.4.1. Syntaktische Betrachtung

Unter der syntaktischen Darstellung von Assoziationsregeln versteht man die Darstellung einer Regel durch Zerlegung in Bedingung und Schlussfolgerung, sowie in deren Interessantheitsmaße. Abbildung 2.11 zeigt eine Kombinationsmatrix. Während die X und Z-Achse jeweils alle darzustellenden Bedingungen (B1...B4) und Schlussfolgerungen (S1...S4) zeigen, wird auf der Y-Achse ein Interessantheitsmaß dargestellt [BKK97; HCC+97]. Sowohl Bedingungen als auch Schlussfolgerungen können hierbei aus mehreren Elementen bestehen. Die Abbildung zeigt, dass diese Darstellung intuitiv ist, aber einige Probleme aufweist.

1. Darstellungen große Regelmengen werden unübersichtlich.

2. Darstellungen sind jeweils nur für ein Interessantheitsmaß möglich.

3. Wichtige Informationen können je nach Betrachtungswinkel der 3D-Darstellung verdeckt werden.

4. Ähnlichkeiten oder auffällige Gruppierungen werden nicht ersichtlich.

5. Korrelationen zwischen ähnlichen Bedingungen/Schlussfolgerungen werden nicht erfasst.

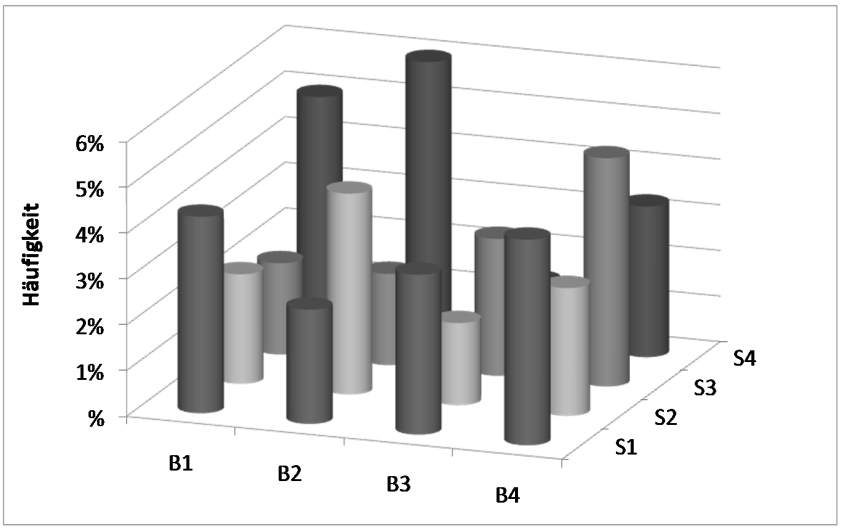
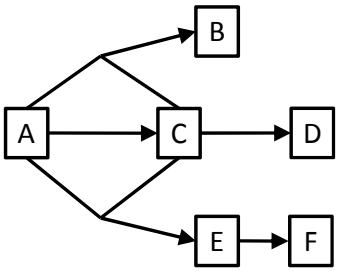
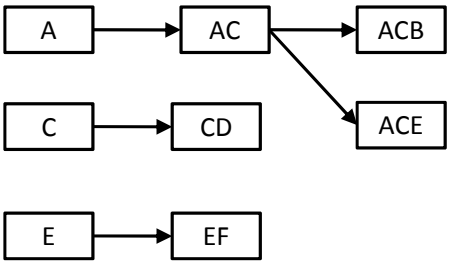


Abbildung 2.11.: Kombinationsmatrix



(a) Regelgraph, Elemente als Knoten



(b) Regelgraph, Kombinationen als Knoten

Abbildung 2.12.: Regelgraphen für $A \Rightarrow B$, $AB \Rightarrow C$, $AB \Rightarrow D$ und $D \Rightarrow E$

Erweiterungen dieser Darstellung erlauben zusätzliche Dimensionen, indem beispielsweise die Farbe oder der Durchmesser der Balken die Ausprägung jeweils eines weiteren Interessantheitsmaßes darstellt. Eine derartige Aufbereitung mehrerer hundert Assoziationsregeln ist jedoch auf Grund der hohen Informationsdichte nur bedingt sinnvoll. Die Erkennung von Besonderheiten und Ähnlichkeiten wird deutlich erschwert.

Weitere Möglichkeiten der syntaxbasierten Darstellung von Assoziationsregeln bestehen in einer Aufbereitung der Regeln zu gerichteten Graphen. Abbildung 2.12 zeigt zwei Variationen solcher Graphen, welche beispielsweise in [BKK97; HCC+97; KMR+94] und [KGLB00] verwendet werden. Je nach Variation können Knoten hierbei sowohl einzelne Elemente als auch komplette Kombinationen darstellen. Die Regel wird durch Verbindungen zwischen den Knoten repräsentiert, wobei jeweils ein Pfeil die Bedingung und Schlussfolgerung trennt.

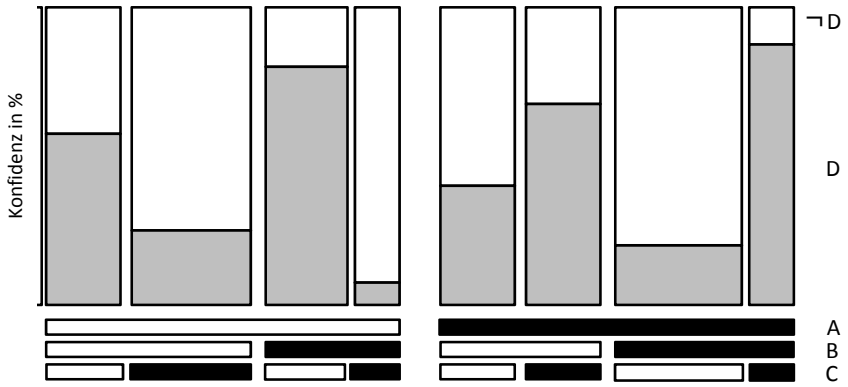


Abbildung 2.13.: Mosaikdarstellung aller Regeln $\mathcal{P}(\{A, B, C\}) \Rightarrow D$

Die Güte und der Wert einer Regel kann in beiden Darstellungen nicht erkannt werden. Es wird lediglich die Existenz der Regel gezeigt. Außerdem impliziert diese Darstellung eine transitive Abhängigkeit zwischen Regeln, wenn diese über mehrere Kanten miteinander verbunden sind. Dies ist nicht der Fall und damit irreführend. Für eine große Menge darzustellender Regeln wird diese Repräsentation zudem sehr schwierig optisch erfassbar.

Eine weitere Methode zur Darstellung einer Regelmenge wird von Friendly [Fri92] vorgestellt. Mosaikdarstellungen, wie beispielsweise in Abbildung 2.13 gezeigt, ermöglichen eine aussagekräftige Visualisierung einer Regelteilmenge. Dabei wird aus der Bedingung einer betrachteten Regel die Potenzmenge gebildet. Weiße bzw. schwarze Balken zeigen das Vorhandensein des entsprechenden Elementes. Die grauen Balken repräsentieren ein Interessantheitsmaß, in diesem Beispiel die Konfidenz der entsprechenden Regel. Die Fläche der Balken stellt die Häufigkeit dieser Regel dar. Jeder Teilkombination der Bedingung kann damit ein Balken des Interessantheitsmaßes zugeordnet werden. Ziel dieser Darstellung ist die Veranschaulichung von Zusammenhängen zwischen einer betrachteten Regel und all ihren generalisierten Formen. Abbildung 2.13 zeigt dies beispielsweise für die Regel $ABC \Rightarrow D$.

Diese Methode ist geeignet zum Erkennen von auffälligen Eigenschaften einer speziellen ausgewählten Regel. Für eine größere Regelmenge kann sie meist nicht verwendet werden, da hier eine Vielzahl an Regelgruppen in jeweils einem Diagramm dargestellt werden müssen. Außerdem kann die Mosaikdarstellung ebenfalls nur ein Interessantheitsmaß sinnvoll zeigen, wodurch eine hinreichende Bewertung einer Regel nur bedingt möglich ist.

2.4.2. Semantische Betrachtung

Semantische Aufbereitungen von Assoziationsregeln können wie syntaktische Darstellungen ebenfalls wichtige Eigenschaften von Regeln zeigen. Von Unwin [UHB01] wird ei-

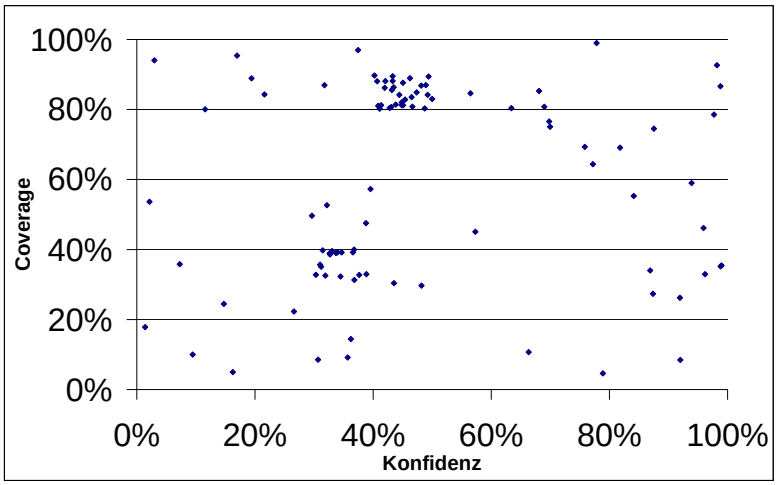


Abbildung 2.14.: Streudiagramm für Konfidenz/Coverage

ne Visualisierung vorgeschlagen, welche auf den Interessantheitsmaßen der untersuchten Regeln beruht. Dabei werden die Regeln als Punkte in einem Streudiagramm (engl. scatterplot) mit je einem Interessantheitsmaß pro Achse dargestellt. Die Punkte werden je nach der Dichte der dargestellten Regeln in einem bestimmten Gebiet eingefärbt. Durch die Auswahl eines Bereiches in diesem Koordinatensystem werden alle Assoziationen im nahen Umfeld markiert Regeln mit wahrscheinlich einer gemeinsamen Domäne erfasst. Abbildung 2.14 zeigt ein Beispiel, in welchem eine Reihe von Assoziationsregeln mit Hilfe ihrer Konfidenz und Coverage dargestellt werden. Jeder Punkt in dem Diagramm stellt eine Regel dar. Die syntaktische Bedeutung der Regel ist nicht Teil der Darstellung.

Mit diesem Ansatz kann eine große Anzahl an Regeln dargestellt werden. Die Navigation erfolgt durch die Selektion einer Regelmenge mit ähnlichen Interessantheitsmaßen. Auch diese Darstellung erlaubt lediglich pro Darstellung zwei Interessantheitsmaße und reduziert die Regeln auf ihre semantischen Informationen. Die einzelnen Elemente und Kombinationen der Regelbestandteile werden nicht beachtet.

In ähnlicher Weise kann pro Interessantheitsmaß eine Darstellung als Säulendiagramm erfolgen. Die Y-Achse zeigt dann ein Interessantheitsmaß und die X-Achse die einzelnen Regeln. Durch die Darstellung mehrerer Säulendiagramme mit verschiedenen Interessantheitsmaßen können durch die gleichzeitige Markierung einer Regel in allen Darstellungen auffällige Verteilungen der Regeln erkannt werden.

Die Syntax der ausgewählten Regeln wird in dieser Darstellung vernachlässigt. Sie ist jedoch oft zur Beurteilung der praktischen Relevanz der Regel nötig. Nach einer semantischen Filterung der Regelmenge kann deren Ergebnis nachfolgend syntaktisch aufbereitet und untersucht werden.

2.5. Anforderungen in SAP-Verkaufsszenarien

Nachdem vorherige Abschnitte das Vorgehen bei der Assoziationsregelanalyse erläutert haben, wird der folgende Abschnitt die Umsetzbarkeit dieses Ansatzes betrachten. Diese Arbeit setzt sich das Ziel, bereits bestehende Konzepte zur Analyse von Firmendaten auf realistischen SAP-Szenarien mit den dabei verwendeten Datenstrukturen und -mengen, sowie Anwendungsanforderungen zu übertragen. Dieser Abschnitt soll die Unterschiede zwischen den theoretischen Anforderungen an Daten und Szenarien und den Wünschen von Anwendern darlegen.

2.5.1. Beschreibung der Zielplattform

Als Ziel dieser Arbeit sollen die erarbeiteten Lösungen im SAP Business Warehouse Accelerator (BWA) implementiert werden. Dieses Produkt der SAP AG wird zur effizienten Aggregation großer Datenmengen genutzt. Der BWA ist in der Lage, die Datenschemas von SAP BI zu verarbeiten. Er weist somit deutliche Ähnlichkeiten zu relationalen Datenbanken auf. Implementierungsrelevante Details über dessen Funktionsweise folgen in Kapitel 5. An dieser Stelle sollen jedoch bereits grundlegende Eigenschaften des BWA erläutert werden. Diese stellen wichtige Anforderungen an die im BWA zu integrierenden Lösungen.

Der BWA nutzt einen Verbund von Einzelrechnern, um die ihm zur Verfügung gestellten Daten zu verarbeiten. Die Berechnung einer Anfrage erfolgt somit potenziell parallelisiert. Das System arbeitet hauptspeicherbasiert. Das bedeutet, dass Zugriffe auf externe Speichermöglichkeiten vermieden werden und alle Daten im Hauptspeicher des Systems vorliegen.

Relationale Datenbanken legen ihre Daten meist zeileorientiert ab. Eine wichtige Besonderheit besteht beim BWA und vergleichbaren Lösungen in der spaltenweisen Ablage der Daten [Ver07; SAB+05]. Dies erlaubt für einige Operationen einen sehr schnellen Zugriff auf relevante Daten und bewirkt oft eine sehr effiziente Komprimierbarkeit.

Der Aufgabe des BWA besteht hauptsächlich in der Aggregation von Daten. Diese Funktion darf entsprechend in keiner Weise gestört werden. Insbesondere dürfen die im BWA gespeicherten Daten nicht modifiziert werden. Die in diesen Fällen oft verwendete Replikation von Daten ist beim BWA nicht erlaubt [BM00; BPT97]. Der entstehende Mehrverbrauch an Hauptspeicher könnte die zur Verfügung stehenden Ressourcen leicht erschöpfen.

Eine in den BWA integrierte Lösung muss somit in der Lage sein, die Daten in der Form zu verarbeiten, in welcher sie gegeben sind. Für eine einfache Verwendung der

zu erarbeitenden Lösungen müssen technische Besonderheiten vor dem Nutzer verdeckt werden. Damit ergeben sich die folgenden, technisch bedingten Anforderungen, um eine Verwendung der Assoziationsregelsuche im BWA zu ermöglichen:

1. Veränderungen der Daten sind nicht erlaubt.

2. Replikationen der Daten sind aufgrund der zusätzlich benötigten Ressourcen zu minimieren.

3. Berechnungen müssen auf verteilten Daten möglich sein.

Zusätzlich zu diesen durch die Zielplattform geforderten Eigenschaften an eine Assoziationsregelsuche sind vor allem die Anforderungen der Anwender wichtig. Dafür wurden vier Kunden der SAP AG befragt, was im folgenden Abschnitt vorgestellt wird.

2.5.2. Beschreibung der Anwendungsszenarien

Kunde1 ist eine in Westeuropa aktive Handelskette mit derzeit 1.500 Filialen und 3 Mrd. € Jahresumsatz. Dessen Filialen erzeugen pro Tag etwa 500.000 Transaktionen und damit rund 150 Millionen Transaktionen pro Jahr. Beteiligt sind daran etwa 60.000 Artikel. Derzeit erfolgen Assoziationsregelanalysen zu unregelmäßigen Zeitpunkten jeweils auf dem Datenbestand des letzten halben Jahres. Das entspricht ca. 75 Millionen Transaktionen. Eine solche Analyse benötigt derzeit jeweils mehrere Stunden. Ein wünschenswertes Ziel von Kunde1 sind Laufzeiten von unter fünf Minuten, um eine breitere und flexiblere Verwendung der Verfahren zu ermöglichen.

Kunde2 unterhält weltweit etwa 3.000 Filialen in 30 Ländern mit Schwerpunkt in den Vereinigten Staaten von Amerika. Der Jahresumsatz beträgt ca. 1 Mrd. US-Dollar. Verkauft werden ca. 30.000 verschiedene Artikel.

Kunde3 ist eine südamerikanische Handelskette mit rund 3 Mrd. US-Dollar Umsatz pro Jahr in ca. 60 Filialen. Kunde4 ist ein Unternehmen der Erdölindustrie mit mehr als 2.000 Tankstellen in Europa. Die Verkaufsdaten enthalten dabei vor allem Abverkäufe der Tankstellenshops.

Zur Anonymisierung werden diese Kunden im Folgenden nicht einzeln untersucht, sondern konsolidiert betrachtet. Alle verwendeten Verkaufsdaten liegen in Form einer Menge von Verkaufstransaktionen vor. Die Beispiele zeigen Anwendungen aus dem Umfeld von SAP-Verkaufsanalysen. Diese stellen ein wichtiges Szenario für Assoziationsregelsuchen dar. Auch wenn eine Vielzahl weitere Szenarien möglich sind, sollen in dieser Arbeit hauptsächlich diese Verkaufsdaten betrachtet werden.

2.5.3. Benötigte Analyseverfahren

Die Daten der Kunden werden in SAP BI-Systemen kontextbasiert verteilt auf mehreren sogenannten *SAP InfoCubes* vorgehalten, den SAP-Implementierungen eines erweiterten Schneeflockenschemas. Eine Transaktion für einen Einkaufsbeleg besteht aus den Attributen Datum, Ort und Transaktionsschlüssel zur eindeutigen filialeübergreifenden Identifizierung einer Transaktion. Neben dem verkauften Produkt wird ein Datensatz mit weiteren Kennzahlen und Merkmalen wie Preis, Menge, Verkäufer, Kasse usw. pro Produkt angereichert. Merkmale einer kompletten Transaktion, wie der Verkäufer oder das Verkaufsdatum, werden redundant jeweils pro Einzelprodukt abgelegt. Insgesamt besteht ein Eintrag aus bis zu 50 Attributen, wobei meist lediglich die Transaktionsschlüssel und Produktbezeichner zur Assoziationsregelanalyse benötigt werden.

Die Datenanalyse erfolgt dreistufig:

1. Ermittlung und Selektion von jeweils zu untersuchenden Produkten oder Transaktionen.

2. Bereinigung unerwünschter oder unvollständiger Einträge.

3. Durchführung einer Assoziationsregelanalyse auf den verbleibenden Daten.

Das Ziel derartiger Analysen kann eine sogenannte *Cross-Sales-Berechnung* oder deren Spezialform der *Promotionsanalyse* sein. Hierbei wird ausgehend von extrahierten Regeln untersucht, wie die Verkäufe von Produkten von einem anderen Produkt abhängen oder wie sich diese in Abhängigkeit eines Produktes verändern. Zusätzlich zu den Interessantheitsmaßen einer Assoziationsregel werden geschäftsrelevante, meist aggregierte Kennzahlen, wie beispielsweise der Umsatz eines Produktes, benötigt. Außerdem müssen Einschränkungen auf den Elementen von Bedingung und Schlussfolgerung einer erzeugten Assoziationsregel möglich sein, sowie eine sinnvolle Gruppierung der erzeugten Regeln nach den Elementen der Regel, Interessantheitsmaßen und Hierarchien.

Die *Promotionsanalyse* untersucht die Auswirkungen von gutscheinbasierten Werbeaktionen. Dabei wird ermittelt, welche Produkte in Korrelation zu dem virtuellen Produkt „Gutschein“ existieren und wie sich diese positiv oder negativ auf den Gesamtumsatz auswirken. Auf diese Weise kann der Gewinn einer Werbeaktion spezifiziert werden. Beide Ansätze betrachten lediglich mögliche Verbindungen zwischen einigen wenigen Produkten, d.h. sie benötigen kurze Assoziationsregeln mit wenigen Elementen.

Eine weitere Anwendung der EHK besteht in der *ABC-Analyse*. Dabei wird mit Hilfe von Kundenkarten das Kaufverhalten der Kunden untersucht und diese durch jeweils ermittelte häufig gekaufte Produktkombination in Segmente geteilt. Jedes Segment kann

darauf mit speziell angepassten Angeboten versorgt werden. Relevant sind hierbei lediglich häufige, prägnante Gruppen.

Derartige Analysen sind bisher aufgrund des hohen Aufwandes lediglich einem eingeschränkten Nutzerkreis zugänglich. Um diesen zu erweitern sind qualitative Änderungen der entsprechenden Verfahren erforderlich. Angestrebt werden beispielsweise Laufzeiten von unter zehn Minuten pro Durchlauf auch bei großen Datenmengen. Außerdem soll die Benutzung auf bereits vorhandenen Datenstrukturen erfolgen, Datenaufbereitung auf ein Minimum reduziert und Replikationen vermieden werden. Dies vereinfacht die Aufrechterhaltung der Datenkonsistenz und damit die Datenverwaltung. Wichtig ist ebenfalls, dass alle nötigen Vorverarbeitungsschritte, wie beispielsweise das Entfernen von Duplikaten oder die Auswahl relevanter Datenbereiche, zur Laufzeit erfolgen und flexibel wählbar sind. Nur damit kann eine hohe Aktualität der Daten erreicht werden.

Um neue Anwender für derartige Verfahren zu gewinnen, muss außerdem die Handhabung stark vereinfacht werden. In aktuellen Lösungen werden die Parameter einer Analyse durch einen Spezialisten für die jeweils zu untersuchenden Daten durch manuelle Annäherungen an sinnvolle Parameterwerte bestimmt und fest definiert oder als Vorschlag dem Anwender präsentiert.

Bei einer deutlich erhöhten Nutzeranzahl und frei wählbaren Datenbereichen ist dies durch die Vielfalt der möglichen Szenarien nicht tragbar. Einem unerfahrenen Anwender fehlen hingegen oft die Intuition und das Fachwissen, um sinnvolle Parameterwerte zu schätzen. Er besitzt meist weder die Zeit noch die Ressourcen, um die Parameter durch Probieren zu erraten. Diese sollen somit entfernt, vereinfacht oder mit sinnvollen, robusten Standardwerten belegt werden. Eine solche Funktionalität kann unnötige Kosten durch extreme Laufzeiten und Fehleingaben vermeiden.

2.5.4. Bestehende Datencharakteristiken

In dieser Arbeit werden realistische Daten verwendet, welche von Kunden der SAP AG zur Verfügung gestellt wurden. Deren Eigenschaften entsprechen weitestgehend den synthetischen Daten vergleichbarer Arbeiten [Sri; Ill]. Im Detail bestehen aber wichtige Besonderheiten der realistischen Daten, welche deutliche Auswirkungen auf das Verhalten einer Assoziationsregelsuche zeigen können. Im Anhang werden einige Eigenschaften dieser Daten dargestellt, öffentlich verfügbar sind sie jedoch nicht.

In diesen Verkaufsszenarien treten oft sehr kurze Transaktionen von drei bis vier Elementen auf. Vereinzelte Transaktionen können trotzdem mehr als 50 Elemente aufweisen. Damit sind Optimierungen für kurze Transaktionen sinnvoll, können jedoch nicht zwangsläufig verwendet werden. Ein Großteil der Transaktionen enthalten Elemente mehrfach. Da lediglich auf Vorhandensein von Elementen geprüft wird, können Duplikate entfernt

werden. Nach einer derartigen Bereinigung des Datenbestandes bestehen ein Großteil der Transaktionen lediglich aus einem Element und ist damit zum Erzeugen von Assoziationsregeln nicht fähig. Die in dieser Arbeit verwendeten Daten weisen nach der Datenbereinigung bis zu 30% einelementige Transaktionen auf. Assoziationsregeln benötigen aber mehrelementige Kombinationen als Basis. Um Interessantheitsmaße zu berechnen, welche auch die Häufigkeit der Einzelelemente benötigen, müssen diese unter Beachtung der einelementigen Transaktionen gegeben sein. Aus dem Datenbestand zur Bestimmung von l-Kombinationen mit $l > 1$ können sie aber entfernt werden, um Speicher und Rechenaufwand zu sparen. Dieser Punkt wird in synthetischen Daten meist nicht beachtet, wo diese Eigenschaft weniger deutlich auftritt.

Eine weitere Besonderheit der untersuchten Daten besteht darin, dass etwa 15% der Transaktionen ausschließlich aus seltenen Elementen bestehen. Außerdem treten sehr häufige Elemente überproportional oft allein in einer Transaktion auf. Synthetische Daten gehen hier von einer gleichmäßigen Verteilung aus. In Kombination mit dem Entfernen von einelementigen Transaktionen verschiebt sich die Rangfolge der Elemente zur Erzeugung von häufigen Kombinationen teilweise deutlich. Die Effizienz von Algorithmen zur EHK ist oft direkt von der Reihenfolge der zu untersuchenden Elemente abhängig. Da hierfür meist eine Sortierung nach Häufigkeit der Elemente vorgenommen wird, kann die Beachtung der bereinigten Häufigkeiten bei realistischen Daten deutliche Leistungssteigerungen erbringen. Da synthetische Daten diese Eigenheiten nicht zeigen, ist eine entsprechende Optimierung nicht nötig.

2.5.5. Konsolidierung der Anforderungen

Aus den Erkenntnissen der vorhergegangenen Abschnitte lassen sich die folgenden Anforderungen an die Assoziationsregelanalyse ableiten.

Flexible Szenariospezifikation
Eine wichtige Forderung an Assoziationsregelsuchen besteht in der Möglichkeit einer flexiblen Szenariobeschreibung. Diese Freiheit soll in mehreren Richtungen gegeben sein. Beispielsweise soll die Wahl der zu untersuchenden Attribute frei wählbar und eine Suche auf verschiedenen Teilbereichen der Daten möglich sein.

Für Warenkorbanalysen sind Szenarien mit frei wählbaren Transaktionsschlüsseln wünschenswert, wodurch entsprechende Optimierungen und Vorberechnungen verhindert werden. Transaktionen bestehen oft aus mehreren Attributen. Dies kann die Bildung einer Datenstruktur zur effizienten Suche häufiger Kombinationen erschweren. Damit ist beispielsweise die Bestimmung der Anzahl der gewählten Transaktionen oder die Zuordnung eines Elementes zu einer Transaktion nur mit signifikantem Berechnungsaufwand möglich und meist nicht aus vorhandenen Statistikdaten ableitbar. Bei großen Datenmengen kann diese Berechnung unter Umständen nicht praktikabel durchgeführt werden.

Der Aufbau der Datenstrukturen zum Durchführen einer Assoziationsregelanalyse kann im Zusammenhang mit komplexen Selektionen und Verbundoperationen mehr Berechnungsaufwand als die darauf erfolgende Assoziationsregelbestimmung erzeugen. Damit kann sich der Fokus einer Optimierung auf die Erzeugung der initialen Datenstrukturen der EHK verlagern. Insbesondere in Verbindung mit der Einschränkung auf kurze Kombinationen steigt der Laufzeitanteil der Vorverarbeitung gegenüber der eigentlichen Suche nach häufigen Kombinationen deutlich.

Aussagekräftige Regeln
Diese Arbeit betrachtet die Analyse realistischer Verkaufsdaten. Untersuchungen von beispielsweise Gensequenzen, welche ebenfalls mit einer Assoziationsregelanalyse möglich sind, sollen nicht betrachtet werden. Im Gespräch mit Kunden der SAP AG hat sich gezeigt, dass bei Verkaufsanalysen meist wenig Interesse an möglichst komplexen Regeln besteht. Viele vergleichbare Arbeiten nehmen dies jedoch an. Eine Assoziationsregel mit mehr als fünf beteiligten Elementen kann einen hohen Informationsgehalt besitzen. Eine effiziente Nutzung dieser Information ist jedoch schwierig. Wirtschaftlich interessanter sind daher kurze, häufige Regeln. Eine Regel mit nur einem oder wenigen Elementen in der Bedingung erlaubt klare Aktionen und eine gezielte Beobachtung von deren Auswirkungen.

Eine Begrenzung der Regellänge bewirkt außerdem oft eine deutliche Verkleinerung des Suchraumes. Dies beschleunigt die eigentliche Analyse signifikant. Die Vorverarbeitung der gewählten Daten wird somit in Relation zur Gesamtlaufzeit ein zeitintensiver Schritt. Außerdem wird oft eine Beschränkung auf bestimmte Produkte in den Bedingungen oder Schlussfolgerungen einer Regel verwendet. Dies kann den zu durchsuchenden Raum relevanter Kombinationen einschränken und erlaubt entsprechende Optimierungen.

Erweiterter Nutzerkreis
Das Anwenden von Assoziationsregelanalysen ist derzeit oft nur einer kleinen Nutzergruppe vorbehalten. Komplexe Parameter verhindern eine einfache Anwendung. Dadurch entgehen potenziellen Anwendern wichtige Informationen. Es liegt somit im Interesse von Unternehmen, solche Prozesse möglichst vielen Nutzern zur Verfügung stellen zu können. Die Probleme betreffen hierbei sowohl den Prozess der Analyse als auch deren korrekte Auswertung.

Hierfür werden verständliche Aufbereitungen von technischen Parametern und Kennzahlen benötigt. Möglich ist hierbei beispielsweise der Umsatz, der durch die Elemente einer Regel erzeugt wird. Diese Rückverfolgung von einer häufigen Kombination auf Eigenschaften der sie erzeugenden Transaktionen benötigt weitere kostenintensive Berechnungen, welche die Strukturen einer EHK nutzen können. Die Datenstrukturen einiger Verfahren, z.B. der Kombinationsbaum von FP-GROWTH, erlauben diese Rückschlüsse nicht.

Robustes Laufzeit- und Systemverhalten
Da die Technologie der Assoziationsregelextraktion für möglichst viele Anwender verfügbar sein soll, kann nur bedingt vorausgesetzt werden, dass diese aus der Wahl ihrer Eingabedaten und -parameter auf das daraus resultierende Systemverhalten schließen können. Damit besteht die Gefahr, die Systemlandschaft eines Unternehmens durch fehlerhafte Bedienung einer Assoziationsregelsuche zu stören. Um dies zu vermeiden, müssen Parameter leicht verständlich oder gar nicht vorhanden sein und Besonderheiten der untersuchten Daten und Rechnersysteme automatisch behandelt werden.

Da reguläre Anwender in erster Linie an dem informativen Charakter von Assoziationsregeln interessiert sind, können zu Gunsten einer stabilen Analyselaufzeit approximierte Ergebnisse verwendet werden. Deren Aussagekraft muss dabei erhalten bleiben. Sind approximierte Ergebnisse unzureichend, können diese von fachlich versierten Anwendern nachträglich gezielt verbessert werden.

Moderne Systemarchitekturen
Dynamisch erzeugte Szenarien sind ein wichtiger Bestandteil moderner Datenanalysen. Um eine aufwändige Vorverarbeitung und Umstrukturierung der untersuchten Daten zu vermeiden, sollten diese nicht repliziert werden. Derartige Daten sind stets aktuell und können keine Inkonsistenzen aufweisen. Aber sie nutzen die gleiche Infrastruktur wie andere Nutzer der Daten. Veränderungen wirken sich somit direkt auf alle Beteiligten aus. Um dies zu vermeiden, sollten Assoziationsregelanalysen die zugrunde liegenden Daten nicht modifizieren.

Ein Trend moderner Systemarchitekturen entfernt sich von zentralen Großrechnern und bewegt sich hin zu verteilten, parallelen Rechenclustern mit einer großen Anzahl an Rechenknoten mit jeweils vergleichsweise wenigen verfügbaren Ressourcen. Ein Vertreter einer solchen Rechnerarchitektur ist der Suchmaschinenbetreiber Google, die im Claremont Report [AAB+08] erwähnten Rechnerwolken (engl. cloud-computing) oder P2P-Systeme. Die Knoten sind untereinander zu Netzwerken verbunden, aber teilen sonst keine Systemkomponenten. Man spricht in diesem Fall von *Shared-Nothing*-Architekturen. Auf derartigen Rechenlandschaften wird eine verteilte Datenverarbeitung für viele Anwendungen zwingend erforderlich, da ein Knoten nur eine begrenzte Datenmenge effizient verarbeiten kann.

Datenverteilung
In Datawarehouse-Szenarien wird oft eine Unterteilung der Geschäftsdaten nach Jahren oder Filiale vorgenommen. Die Daten einer solchen Partition werden jeweils in identischen Datenschemas abgelegt. Eine solche Vorgehensweise erlaubt eine Fokussierung der Optimierung auf aktuelle Daten. Ältere Partitionen werden beispielsweise nicht im Hauptspeicher des Datenbanksystems vorgehalten. Aktuelle Partitionen werden zusätzlich durch Optimierungen wie Cache-Verfahren, Datenbankindizes und materialisierte Sichten beschleunigt. Nicht benötigte Daten können hierbei trotz einer großen Datenmenge sehr effizient gelöscht werden, indem die komplette Partition entfernt wird. Die

Aufteilung der Daten wird somit durch semantisches Wissen des Anwenders geprägt und ist nicht beliebig automatisch definierbar.

Für viele Anfragen werden Daten aus mehreren dieser Partitionen benötigt, beispielsweise für Mehrjahresbilanzen oder Verkaufs- oder Warenkorbanalysen. In diesen Fällen muss eine Vereinigung der Daten oder eine verteilte Anfrageverarbeitung erfolgen. Eine Vereinigung bedeutet eine Replikation der Daten und die Veränderung der Partitionierung. Die ist oft aus Datensicherheitsgründen nicht möglich oder die dabei entstehende Datenmenge kann mit einem Einzelknoten nicht sinnvoll verarbeitet werden. Eine Datenpartitionierung kann nach verschiedenen Verfahren erfolgen:

1. Daten können kontext- und inhaltsunabhängig gleichmäßig auf eine Menge von Partitionen verteilt werden.

2. Daten können kontextabhängig partitioniert werden, beispielsweise nach Land, Jahr, Produktgruppe oder Filiale.

3. Daten können nach ihrem Inhalt partitioniert werden. Dies entspricht oft einer kontextabhängigen Unterteilung aber kann auch eine Verteilung mittels *Hash*-Wert oder anderer spezifischer Eigenschaften darstellen, welche den Kontext unberücksichtigt lassen.

Vor allem kontextbasierte Partitionierungen können sehr unterschiedliche Partitionsgrößen und Datencharakteristiken erzeugen. Dies bewirkt ein sehr unterschiedliches Verhalten der Analyseverfahren pro Partition und verletzt nach Zaki [Zak99] eine wichtige Voraussetzung für eine effiziente Suche von Assoziationsregeln.

In dieser Arbeit werden damit folgende konkrete Anforderungen an die Assoziationsregelanalyse gestellt:

- Die Suche von häufigen Kombinationen erfolgt über Top-N-Verfahren.

- Die Suche, inklusive Top-N-Verfahren, muss verteilt durchgeführt werden können.

- Statische Vorberechnungen, Datenreplikationen oder Modifikationen der Ausgangsdaten sind nicht erlaubt.

- Die Partitionseigenschaften und Dateneigenschaften dürfen die einfache Verwendung der Verfahren nicht verhindern.

- Die zu untersuchenden Daten und damit deren Partitionierung erfolgt dynamisch, wodurch sämtliche Aufbereitung zur Anfragelaufzeit erfolgen müssen.

- Die EHK erfolgt auf auch anderweitig im SAP BWA verwendeten Daten und darf deren Verwendung in keiner Weise stören.

- Die Architektur des SAP BWA (mehrere Blade-Server, hauptspeicherbasiert, spaltenweise Datenablage) muss für eine effiziente Implementierung vollständig ausgenutzt werden.

- Die Bewertung der Regeln muss automatisiert erfolgen und dem Nutzer Hinweise und Navigationshilfen auf interessante Regeln bieten, ohne von ihm Fachkenntnisse oder die aufwändige Beschreibung seiner Suchziele zu fordern.

- Die einfache, robuste Anwendung der Verfahren durch unerfahrene Nutzer hat Vorrang gegenüber der möglichst effizienten Ausführung der Regelsuche oder der professionellen Auswertung von deren Regeln.

Bisherige Arbeiten nutzen die schwer zu wählende Mindesthäufigkeit als Steuerparameter und vermeiden partitionierte Daten. Verteilte Berechnungen werden oft durch (teilweise) Replikation oder die Umverteilung der Daten ermöglicht oder setzen besondere Datencharakteristiken voraus. All dies ist eine sinnvolle Strategie, wenn die Assoziationsregelsuche von einem Spezialisten ausgeführt wird. Dieser kann die Daten korrekt bereitstellen und ist sich der Auswirkungen einer Wahl falscher Parameter bewusst. Für die in dieser Arbeit adressierten unerfahrenen Nutzer gilt das nicht, was den höheren Aufwand bei leichter zu nutzenden Verfahren rechtfertigt.

Ähnliche Einschränkungen gelten bei der Auswertung der Assoziationsregeln. Hierfür sind eine Vielzahl von Maßen bekannt, welche jeweils wichtige Aspekte der Regeln betonen, deren korrekte Deutung jedoch nur durch Fachleute möglich ist. Die gezielte Wahl eines Maßes kann bessere Wertungen der Regeln ermöglicht, als es ein allgemeines Maß leisten kann. Letzteres erlaubt jedoch bereits einen allgemeinen Eindruck über den Wert einer Regel und kann oft einfacher gedeutet werden.

2.6. Zusammenfassung

In diesem Kapitel wurde ein Überblick über die in der Literatur bereits bekannten Verfahren zur Bestimmung von Assoziationsregeln vorgestellt. Dabei erfolgten in je einem Abschnitt Erläuterungen zu den drei dafür nötigen Arbeitsschritten.

Begonnen wurde mit der Einführung der grundlegenden Algorithmen APRIORI, FP-GROWTH, ECLAT, BUC und PARTITION zur Datenanalyse. Mit diesen lassen sich häufige Kombinationen bestimmen, welche als Grundlage für Assoziationsregeln dienen. Im Hinblick auf eine vereinfachte Anwendung dieser Verfahren folgte ein Überblick über

verwandte Problemstellungen und Verfahren zur Top-N-Berechnung. Anschließend wurden Möglichkeiten zur Optimierung der Datenanalyse aufgezeigt, welche die Ergebnismenge auf sinnvolle Einträge begrenzen. Dies umfasst neben der Beschränkung auf häufige, geschlossene Kombinationen auch die Verwendung von Regelvorlagen. Mit beiden Ansätzen können einige Ergebnisse bereits während der Berechnung der Kombinationen verworfen werden, wodurch eine nachfolgende Weiterverarbeitung vereinfacht wird.

Im Anschluss an die Erläuterungen zur Datenaufbereitung erfolgte die Einführung über die Aufbereitung der aus den häufigen Kombinationen gebildeten Assoziationsregeln. Vorgestellt wurden verschiedene Möglichkeiten zur Bewertung und Einteilung der Regeln. Diese Arbeit nutzt hauptsächliche objektive Interessantheitsmaße zur Bewertung von Regeln. Diese stellen mathematische Zusammenhänge oder statistische Auffälligkeiten von Regeln dar und sind somit nicht vom Nutzer und dessen Kontextwissen abhängig. Sie eignen sich damit zur automatischen Beurteilung von Regeln. Einige wichtige objektive Interessantheitsmaße, wie z.B. Lift, Konfidenz oder Überzeugung, wurden näher erläutert und der Abschnitt mit einem Überblick über eine Auswahl von bekannten Maßen abgeschlossen.

In Abschnitt 2.4 wurde der dritte Schritt der Assoziationsregelanalyse, die manuelle Auswertung der Regeln durch den Nutzer, betrachtet. Dieser Schritt ist für eine erfolgreiche Auswertung von Regeln sehr wichtig, wird aber hauptsächlich durch geschickte, manuelle Navigation des Anwenders bestimmt. Entsprechend beschränken sich die dafür in dieser Arbeit gezeigten Lösungen auf die grafische Aufbereitung der Assoziationsregeln.

Abschließend wurden in Abschnitt 2.5 die Aspekte realistischer, praxisnaher Verkaufsszenarien am Beispiel von vier Kunden der SAP AG betrachtet. Insbesondere im Hinblick auf den in dieser Arbeit adressierten geschäftlichen Bereich der Assoziationsregelanalyse können deutliche Konflikte zwischen den Ansprüchen der Verfahren und den gewünschten Zielen bestehen. Die daraus folgenden Anforderungen an die Verfahren zur Assoziationsregelbestimmung wurden zusammengefasst und bilden die grundlegenden Problemstellungen dieser Arbeit: die schwierig durchzuführende Extraktion häufiger Kombinationen und die Auswertung der daraus erzeugten Assoziationsregeln. Beide Probleme werden in je einem der folgenden Kapitel untersucht.

3. Anwenderfreundliche Bestimmung häufiger Kombinationen

Immer schneller wachsende Datenbestände stellen für moderne Unternehmen gleichzeitig eine große Chance und eine Herausforderung dar. Sie enthalten wichtige Informationen, welche richtig gedeutet eine effiziente Anpassung an die Bedürfnisse und Wünsche von Kunden oder die Optimierung interner Abläufe erlauben. Sie können somit einen wichtigen Wettbewerbsvorteil darstellen. Die interessanten Informationen sind jedoch oft nur schwer zu erkennen oder werden von einem Übermaß unwichtigen Informationen verdeckt. Das Erlangen wichtiger Erkenntnisse aus den jeweils eigenen Datenbeständen wird somit in vielerlei Hinsicht erschwert. Zusätzlich wachsen die Datenbestände von Unternehmen oft deutlich schneller als die Anzahl verfügbarer Fachleute zur Analyse dieser Daten. Selbst mit genügend vorhandener Rechenleistung kann durch die begrenzten personellen Ressourcen nicht beliebig skaliert werden. Somit erscheint eine einfach nutzbare Implementierung zur Extraktion von Wissen aus Informationen ökonomisch sinnvoll.

Wenn die Durchführung solcher Analyseverfahren vereinfacht wird, indem weniger Fachkenntnisse gefordert werden, kann die Anzahl der möglichen Anwender erhöht werden. Dies steigert den praktischen Wert der Datenbestände von Unternehmen. Dies gilt auch für die in dieser Arbeit untersuchte Bestimmung von Assoziationsregeln und die dafür erforderliche Extraktion häufiger Kombinationen. Die hierfür benötigten technischen Parameter der Algorithmen und Besonderheiten der verwendeten System- und Datenbankarchitekturen müssen vor dem Nutzer verborgen werden, um eine einfache Anwendung zu erlauben. In Betracht gezogen werden sollen sie nur bei Bedarf und entsprechendem Expertenwissen des Anwenders. In diesem Kapitel wird die Ersetzung technischer Parameter untersucht und darauf folgend eine Anpassung von Algorithmen und Parametern an praxisnahe SAP-Verkaufsszenarien im SAP BW Accelerator diskutiert.

Begonnen wird mit der Erweiterung des Algorithmus ECLAT um die Fähigkeit zur effizienten Suche der Top-N-Kombinationen. Dieser Algorithmus wird im darauf folgenden Abschnitt auf partitionierte Datenbestände übertragen. Mit Hilfe der Top-N-Verfahren wird die Suche häufiger Kombinationen erleichtert. Im nächsten Kapitel werden zusätzlich zu diesen Parametervereinfachungen Möglichkeiten präsentiert, die Menge der häufigen Kombinationen im Hinblick auf die nachfolgende Aufbereitung zu Assoziationsregeln zu bereinigen. Hierfür werden Ansätze für die Verwendung von geschlossenen Kombinationen gezeigt und Alternativen für die Verwendung von Top-N-Strategien präsentiert. Im Zen-

trum der Untersuchung steht jedoch die Suche nach den N häufigsten Kombinationen in partitionierten Datenbeständen.

3.1. Effiziente Bestimmung von Top-N-Kombinationen

Der wichtigste technische Parameter einer Assoziationsregelanalyse ist die Mindesthäufigkeit der ihr zugrunde liegenden Kombinationen. Dieser steuert die Abwägung zwischen Ressourcenverbrauch, Ergebnismenge und Ergebnisqualität. Dabei entstehen mehrere Probleme.

1. Der Zusammenhang zwischen den gewünschten Assoziationsregeln und den dafür benötigten häufigen Kombinationen ist für den unerfahrenen Anwender schwer verständlich.

2. Der Bedeutung der Mindesthäufigkeit und deren Auswirkungen bei einer Assoziationsregelsuche sind dem Anwender oft unklar.

3. Die korrekte Wahl dieses Parameters ist anwendungs- und datenabhängig. Das macht sie schwierig. Die gezielte Auswahl ist aber sehr wichtig.

Die Mindesthäufigkeit kann sowohl absolut angegeben werden als auch relativ zur Anzahl der betrachteten Transaktionen. Beide Angaben können je nach Anwendungsfall intuitiv durch einen Anwender gewählt werden. Selbst wenn dieser die Bedeutung des Parameters kennt, bleibt diese Wahl oft schwierig. Je nach Art der untersuchten Daten kann der Begriff *häufige Kombination* sehr unterschiedlich gedeutet werden. Einige Datensätze weisen bereits bei einer Mindesthäufigkeit von über 90% sehr viele Kombinationen auf, während die häufigste Kombination anderer Datenbestände nur in $0,1\%$ aller Transaktionen vorkommt. Erschwerend ist, dass schon leichte Abweichungen des gewählten Wertes signifikante Auswirkungen auf die Ergebnismenge und die Algorithmenlaufzeit haben können.

Um dem Anwender diese schwierige Wahl zu erleichtern, wird in dieser Arbeit die Verwendung von Top-N-Strategien vorgeschlagen. Der Nutzer wählt eine gewünschte Ergebniskardinalität aus und überlässt dem Algorithmus die Suche nach den passenden Parametern.

Die grundlegende Vorgehensweise zur Bestimmung der N häufigsten Kombinationen wurde in Abschnitt 2.2.6 bereits vorgestellt. Für viele der gängigen Algorithmen zur EHK sind entsprechende Adaptionen bekannt. Von Cong werden beispielsweise Möglichkeiten für Top-N-Lösungen mit einer definierten Mindestlänge beschrieben [Con01]. He schlägt eine Möglichkeit zu einer approximierten Top-N-Lösung mittels ECLAT vor [He05]. Au-

ßerdem bestehen Lösungen für FP-GROWTH [Che04], APRIORI [FKT00] und geschlossene Kombinationen [HWLT02].

Die Wahl des verwendeten spezifischen Algorithmus ist bei Assoziationsregelsuchen in praxisnahen Verkaufsdaten wenig relevant. Zheng [ZKM01] zeigt, dass für diese Anwendungen die bekannten Algorithmen weitestgehend gleich effizient arbeiten. Deutliche Unterschiede entstehen lediglich bei synthetischen Daten oder unrealistischen Parameterbelegungen.

Die Algorithmenwahl kann somit auf die Zielplattform, den SAP BW Accelerator (BWA), optimiert werden. Das betrifft insbesondere dessen begrenzte Ressource Hauptspeicher und die attributweise, spaltenorientierte Speicherung der Daten. Eine effiziente Integration der Algorithmen in die bestehende Infrastruktur kann die Durchführung der Assoziationsregelsuche vereinfachen und beschleunigen. Um Hauptspeicher zu sparen, ist die Verwendung bereits vorhandener Datenstrukturen sinnvoll. Diese entsprechen zu großen Teilen bereits denen von ECLAT, was sowohl die Vorverarbeitung der Daten als auch die EHK vereinfacht. Eine Verwendung dieser Daten durch ECLAT ist somit effizient realisierbar. Die Datenstrukturen eines FP-GROWTH, die Kombinationsbäume, sind vor allem durch ihre Zeigerstrukturen deutlich anders organisiert als die Daten des BWA. Die bestehenden Strukturen können nur im geringen Maße verwendet werden.

Zusätzlich ermöglicht die spaltenweise Ablage der Daten den direkten, schnellen Datenzugriff auf benötigte Informationen. Insbesondere bei einer sehr nah an den Daten arbeitenden Implementierung sind Datenzugriffe effizient durchführbar. Die Optimierungen des APRIORI-Algorithmus auf möglichst effiziente Datenzugriffe wirken an dieser Stelle nur bedingt. Zudem ist der APRIORI-Algorithmus den Algorithmen ECLAT und FP-GROWTH oft an Effizienz unterlegen [ZPOL97; HPY00]. Der Algorithmus ECLAT stellt somit insgesamt die beste Alternative zur Integration in den SAP BW Accelerator dar.

Die Anpassung von ECLAT zur Extraktion von Top-N-Ergebnissen ist anhand der Beschreibungen im Abschnitt 2.2.6 leicht nachzuvollziehen. Das Vorgehen folgt dem Branch-and-Bound-Verfahren [LD60] und soll daher lediglich in schematischer Form und mit Hilfe eines Beispiels erläutert werden.

Der Algorithmus 3.1 verdeutlicht die im Rahmen dieser Arbeit erstellte Top-N-Implementierung für ECLAT. Im Folgenden soll der Algorithmus mit der Bezeichnung *TOP-N-ECLAT* referenziert werden. Dessen Funktionsweise beruht darauf, dass die jeweilig gültige Mindesthäufigkeit während der Laufzeit ausgehend vom minimalen Startwert schrittweise erhöht wird. Sie gleicht sich somit stetig an die Häufigkeit der abschließend N-ten Kombination an. Dieses Vorgehen ist effizient, wenn die Anpassung schnell erfolgt. Da ab N vorhandenen Kandidaten die erforderliche Ergebniskardinalität erreicht ist, sind seltenere als die bereits bestimmten Kombinationen garantiert nicht Teil des Endergebnisses. Damit können diese übersprungen werden. Damit kann eine Grenze gesetzt werden, mit deren Hilfe Teilbereiche des Suchraumes ausgeschlossen werden.

Algorithmus 3.1 Algorithmus TOP-N-ECLAT

Require: Transaktionsdatenbestand DB
Require: Menge häufiger Produkte E
Require: Anzahl gesuchter Kombinationen N

```
1:  minSupport = 1
2:  procedure eclat_topn(set, last, RESULT)
3:  pos = 0
4:  for all i ∈ E do
5:    if pos < last then
6:      zaehler = count(DB, set ∪ i) /* bestimme Häufigkeit */
7:      if zaehler ≥ minSupport then
8:        RESULT ← (set ∪ i, zaehler)
9:        grenze = min(N, |RESULT|)
10:       minSupport = RESULT[grenze].zaehler
11:       eclat_topn(set ∪ i, pos, RESULT)
12:     end if
13:   end if
14:   pos++
15: end for
16: end procedure
17: RESULT = ∅
18: eclat_topn(∅, |E|, RESULT)
19: filter_topn(RESULT)
20: return RESULT
```

Die Erweiterungen von ECLAT auf Top-N-Suchen erfolgen hierbei in den Zeilen 1, 9 und 10. In Zeile 1 wird die Mindesthäufigkeit initialisiert. Die Zeilen 9 und 10 bestimmen die zu nutzende Mindesthäufigkeit in Abhängigkeit von der aktuellen Ergebniskardinalität und passen den Parameter dynamisch an. Das restliche Vorgehen entspricht dem von ECLAT. Die große Ähnlichkeit zwischen ECLAT und TOP-N-ECLAT zeigt, dass beide Verfahren je nach Anforderungen des Anwenders gleichermaßen verwendet werden können, ohne jeweils eigene, redundante Implementierungen zu besitzen.

Für die EHK sind viele Szenarien denkbar, welche keine vollständigen Ergebnisse erfordern. Die seltensten Kombinationen eines korrekten Top-N-Ergebnisses können beispielsweise als wenig wichtig erachtet werden. Relevant ist dieser Umstand vor allem dann, wenn mehrere Ergebnisse die N-te Position belegen und die gleiche Häufigkeit aufweisen. Auf diese Kombinationen kann verzichtet werden, wenn sie durch ihre Seltenheit vernachlässigbar sind. Das Ergebnis ist in diesem Fall nicht länger korrekt und nicht deterministisch. Mehrere gleiche Suchen können verschiedene Ergebnisse liefern, je nachdem welche Kombination für die N-te Position ausgewählt wurde. Für viele Anwendung ist dies jedoch ein tragbarer Umstand. Um dieses Vorgehen mit Hilfe von TOP-N-ECLAT

TID	Elemente
T01	a b c d
T02	c
T03	d
T04	d
T05	d
T06	d
T07	b c d
T08	d
T09	a b c
T10	a b c
T11	a b c d
T12	c

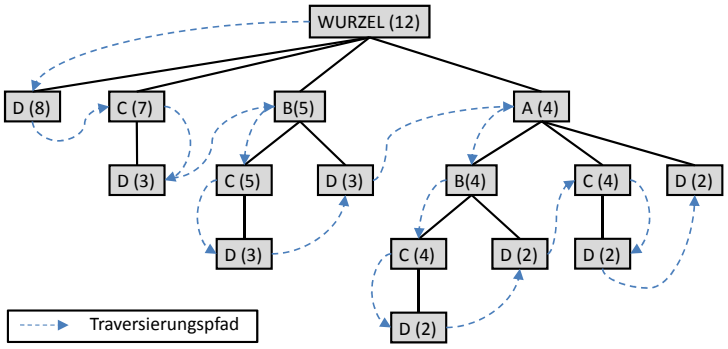


Abbildung 3.1.: Beispiel und Ablauf für ECLAT

zu erreichen, kann die Mindesthäufigkeit in Zeile 10 jeweils um eins erhöht werden, sobald N Kandidaten verfügbar sind. Alle Kombinationen, welche häufiger als das N-te Ergebnis sind, werden weiterhin garantiert bestimmt. Die Kombinationen, deren Häufigkeit derjenigen der N-ten Kombination entsprechen, sind jedoch zufällig aus der Menge aller entsprechenden Kombinationen ausgewählt und können variieren.

Durch eine solche Vereinfachung kann eine unerwartet lange Laufzeit, insbesondere bei extrem niedrigen Mindesthäufigkeiten, vermieden werden. Begründet wird dies dadurch, dass ab der Bestimmung von N Ergebniskandidaten jeder nachfolgend betrachtete Kandidat entweder verworfen werden kann oder zur Anhebung der Mindesthäufigkeit beiträgt. Dies wird erreicht, indem er einen der seltensten Ergebniskandidaten verdrängt. Damit wird die Mindesthäufigkeit erhöht oder zumindest die Wahrscheinlichkeit einer zukünftigen Anhebung vergrößert.

Bis zum Erreichen der finalen Mindesthäufigkeit werden bei diesem Verfahren einige Ergebniskandidaten zum Zeitpunkt ihrer Bestimmung mit einer im Endeffekt zu geringen Häufigkeit ermittelt, einsortiert und gespeichert. Der Mehraufwand vergrößert sich somit, je langsamer die Mindesthäufigkeit ansteigt. Dieser Kostenfaktor erhöht die Laufzeit des Algorithmus deutlich und sollte minimiert werden. Sowohl der Ablauf von TOP-N-ECLAT als auch die dabei auftretenden Probleme werden im nachfolgenden Beispiel verdeutlicht.

Beispiel:

Das Ziel dieses Beispiels ist die Bestimmung der Top-5-Kombinationen. Als Datenbestand dient die Tabelle und deren äquivalente Baumdarstellung in Abbildung 3.1. Die

gepunktete Linie symbolisiert die Traversierung durch die Baumstruktur bei der Anwendung von ECLAT. Die Zahl in Klammern zeigt die jeweilige Anzahl entsprechender Transaktionen einer Kombination. Dabei ist zu beachten, dass es unter Zuhilfenahme der Zwischenergebnisse jedes Knotens einfach und effizient möglich ist, die Transaktionen der jeweiligen Kindknoten zu ermitteln. Diese sind jeweils eine Teilmenge der Transaktionen des Elternknotens. Ohne dieses Vorwissen ist eine direkte Bestimmung der Häufigkeit einer Kombination deutlich aufwändiger und muss gegebenenfalls durch eine Suche im vollständigen Datenbestand erfolgen.

In der Repräsentation als Baumstruktur wird der Baum ausgehend vom Wurzelknoten von links nach rechts der gepunkteten Linie folgend untersucht. Der vollständige Pfad wird durchlaufen, wenn keine Mindesthäufigkeit gefordert ist. Erreicht ein Knoten nicht die jeweils gültige Mindesthäufigkeit, werden alle seine Kindknoten gemäß der Downward-Closure-Eigenschaft diese auch nicht erfüllen. Entsprechend ist keine Betrachtung notwendig und die Suche wird mit dem Nachbarknoten fortgesetzt.

Kombinationen mit der gleichen Häufigkeit wie die N-te Kombination erreichen in diesem Beispiel ebenfalls eine Aufnahme in die Ergebnismenge. Die fettgedruckten Einträge in Tabelle 3.1 symbolisieren die zu diesem Schritt jeweils gültigen Top-5 Kombinationen. Die restlichen Einträge zeigen ehemalige Top-5 Kandidaten, welche aus der Ergebnismenge verdrängt wurden. Derartige Kombinationen können zur Optimierung des Speicherverbrauches des Algorithmus in regelmäßigen Abständen entfernt werden, sollen in diesem Beispiel aber zur Verdeutlichung des auftretenden Problems bestehen bleiben.

Nr.	**Top-Liste**	**sup**	**Nr.**	**Top-Liste**	**sup**
1	[**D:8**]	1	7	[**D:8, C:7, B:5, BC:5, CD:3, BCD:3, BD:3**]	3
2	[**D:8, C:7**]	1	8	[**D:8, C:7, B:5, BC:5, A:4**, CD:3, BCD:3]	4
3	[**D:8, C:7, CD:3**]	1	9	[**D:8, C:7, B:5, BC:5, A:4, AB:4**, CD:3, BCD:3]	4
4	[**D:8, C:7, B:5, CD:3**]	1	10	[**D:8, C:7, B:5, BC:5, A:4, AB:4, ABC:4**, CD:3, BCD:3]	4
5	[**D:8, C:7, B:5, BC:5 CD:3**]	3	11	[**D:8, C:7, B:5, BC:5, A:4, AB:4, ABC:4, AC:4**, CD:3, BCD:3]	4
6	[**D:8, C:7, B:5, BC:5, CD:3, BCD:3**]	3			

Tabelle 3.1.: Beispiel: Ablauf TOP-N-ECLAT

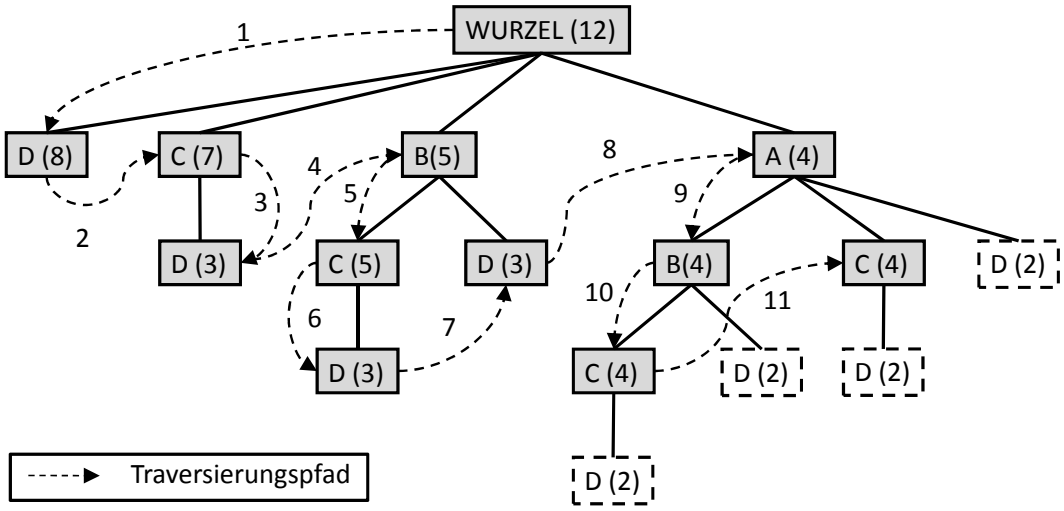


Abbildung 3.2.: Beispiel: TOP-N-ECLAT

Der Verlauf des Algorithmus in Tabelle 3.1 entspricht den Nummerierungen in Abbildung 3.2. In der Spalte *Top-Liste* wird die bis zu dem Zeitpunkt erzeugte Ergebnismenge dargestellt. Die Spalte *sup* zeigt die jeweils gültige absolute Mindesthäufigkeit.

Wie in Tabelle 3.1 leicht ersichtlich ist, werden zwei Kombinationen, (CD:3, BCD:3), als Ergebniskandidaten gespeichert, obwohl sie das endgültige Top-N-Ergebnis nicht erreichen. Auf komplexeren Datenbeständen können an dieser Stelle deutlich mehr unnötige Zwischenergebnisse erzeugt werden. Diese werden zumindest zeitweise gespeichert und müssen verwaltet werden. Damit kann ein signifikant erhöhter Ressourcenverbrauch entstehen.

Im Anhang A, Seite 187, wird der Algorithmus für das gleiche Beispiel aufgezeigt, wobei neue Kandidaten häufiger sein müssen als der aktuell seltenste Kandidat. Dabei kann die Bestimmung der Top-5 Kombinationen nach bereits sieben Schritten mit einem oft sinnvollen, aber nicht vollständigen Ergebnis beendet werden.

Im Vorfeld der Analyse ist dabei nicht abschätzbar, wie schnell sich die Mindesthäufigkeit an ihren Endwert angleicht. Die im Rahmen der Evaluation ermittelte Laufzeiten sind bei großem N meist etwa $30\% - 50\%$ über der Laufzeit mit vordefinierter Mindesthäufigkeit. Das entspricht im Wesentlichen dem Verhalten anderer Top-N-EHK-Implementierungen [ZH99; PHM00]. Ein solcher Mehraufwand kann durch die vereinfachte Bedienung der EHK gerechtfertigt werden.

In vielen Szenarien ist die Anwendung von Top-N-Strategien unproblematisch und einfach durchzuführen. In einigen Konstellationen können jedoch für den Nutzer unerwartete Verzögerungen auftreten, sollte die Mindesthäufigkeit nur langsam steigen. Dies widerspricht den Zielen dieser Arbeit, nach denen derartige Top-N-Suchen robust und einfach verwendbar sein sollen. Zur Behebung dieses Problems werden im nachfolgenden Abschnitt Lösungsmöglichkeiten vorgestellt.

3.1.1. Heuristische Kandidatenpriorisierung mit FASTINC

Eine Methode zur Reduktion von unnötig betrachteten Top-N-Kandidaten wurde von Han, Wang, Lu und Tzvetkov [HWLT02] unter dem Namen *TFP* vorgestellt. Deren Arbeit befasst sich mit der Bestimmung der häufigsten N geschlossenen Kombinationen. Die grundlegende Idee der Arbeit beruht darauf, die Traversierung des Suchraumes jeweils bei dem Knoten durchzuführen, dessen Häufigkeit von allen bisher bekannten Kandidaten maximal ist. Als Kandidat gelten Knoten, dessen Kinder bisher nicht untersucht wurden. Der Algorithmus endet, sobald die Kindknoten aller Top-N-Kandidaten untersucht sind. TFP ist effizient für kleinere Datenmengen oder kleine N. Ein Top-N-Kandidat muss hierbei nicht zwangsläufig Teil des Endergebnisses werden. Ein Knoten, welcher zur Erzeugung weiterer Kandidaten herangezogen wird, weist aber die größte Häufigkeit aller bisher nicht untersuchten Knoten auf. Damit besteht für diesen Knoten die höchste Chance, weitere Top-N-Kandidaten zu erzeugen. Garantiert ist dies jedoch nicht.

Das vollständig korrekte Ergebnis wird ermittelt, da alle Top-N-Ergebnisse um die jeweils möglichen Element erweitert wurden. Diese Kombinationen sind entweder selbst Teil des Ergebnisses oder haben die geforderte Mindesthäufigkeit nicht erreicht. Initialisiert werden die möglichen Kandidaten mit den einelementigen Kombinationen. Nach der Downward-Closure-Eigenschaft kann keine Kombination häufiger sein als seine Teilkombinationen. Durch Induktion kann die Richtigkeit und Vollständigkeit des Ergebnisses gezeigt werden.

Der Algorithmus benötigt für große N oder umfangreiche Datenbestände jedoch viel Speicher, da jeder noch nicht näher untersuchte Ergebniskandidat mindestens die Liste seiner Transaktionen vorhalten muss. Nur damit kann ein effizienter Abstieg zu den jeweiligen Kindknoten durchgeführt werden. Unter Ausnutzung von Kompressionsstrategien, wie Bitvektoren kann dieser Speicherverbrauch reduziert werden. Er ist für Szenarien mit mehr als 100 Millionen Transaktionen und $N > 10.000$ jedoch nicht praktikabel.

Außerdem werden Suchen nach häufigen Kombinationen mit zusätzlichen Einschränkungen, z.B. einer Mindestlänge, aufwändig. Hier müssen neben den N Kandidaten auch alle kürzeren Kombinationen als die mit Mindestlänge betrachtet werden, sollten sie die Mindesthäufigkeit erreichen. Das kann den benötigten Speicherplatz deutlich erhöhen.

Ein Durchlaufen des Suchraumes mit einer möglichst geringen Anzahl betrachteter Ergebniskandidaten ist jedoch wünschenswert. Im Folgenden werden zur Lösung Vorverarbeitungsschritte diskutiert, welche eine untere Grenze lb für die Häufigkeit der N-ten Kombination ermitteln und eine abschließende reguläre Top-N-EHK mit einer Mindesthäufigkeit $minSupport = lb$ durchführen (siehe Algorithmus 3.1, Zeile 1). Die erhöhte initiale Mindesthäufigkeit verhindert die Untersuchung von Teilbäumen, welche nach den Erkenntnissen aus dem Vorverarbeitungsschritt keine Chance auf Aufnahme in das endgültige Ergebnis haben.

Der Vorverarbeitungsschritt arbeitet approximativ. Es wird hierbei versucht, die aussichtsreichsten Top-N-Kandidaten innerhalb des Suchraumes bevorzugt zu prüfen. Damit soll erreicht werde, dass die Liste mit Top-N-Kandidaten schnell mit häufigen Kombinationen gefüllt wird und damit seltene Kombinationen möglichst nicht Top-N-Kandidat werden. Ein derartig approximiertes Top-N-Ergebnis kann effizienter bestimmt werden als durch eine vollständige Analyse des Suchraumes. Die Güte des Ergebnisses ist allerdings unklar. In dieser Arbeit wird die Kombination dieses Vorverarbeitungsschrittes und dem im vorherigen Abschnitt vorgestellten TOP-N-ECLAT als *FASTINC*-Algorithmus bezeichnet.

Für einige Anwendungen ist eine direkte Verwendung des durch FASTINC ermittelten approximierten Ergebnisses möglich. Die Qualität des Ergebnisses ist oft deutlich reduziert, da bei einem solchen Ergebnis in den meisten Fällen einige korrekte Kombinationen nicht ermittelt werden. Außerdem können bei einer derartigen Approximation nur wenige Aussagen über die erreichte Güte angegeben werden. Die Rückgabe eines solchen Ergebnisses an den Nutzer ist damit nur selten tragbar. Die Häufigkeit der approximiert N-ten Kombination weicht meist nur gering von der Häufigkeit der korrekten N-ten Kombination ab, die Ergebnismengen können sich jedoch deutlich unterscheiden. Das Wissen über einen unteren Grenzwert der Häufigkeit der N-ten Kombination kann aber genutzt werden, um in einer weiteren regulären EHK das vollständige, korrekte Ergebnis effizient zu ermitteln. Die Vorverarbeitung und die darauf aufbauende exakte Suche weisen gemeinsam meist eine geringere Laufzeit als eine reguläre Top-N-Suche ohne Vorverarbeitungsschritt auf. Der folgende Abschnitt beschäftigt sich mit den Möglichkeiten zur Bestimmung der Grenze lb.

3.1.2. Eingrenzung des möglichen Suchraumes

Bei der Traversierung des Suchraumes von TOP-N-ECLAT (vgl. Abbildung 3.2) kann beobachtet werden, dass die Häufigkeit einer Kombination mit wachsender Pfadlänge verringert wird. Dies ist eine direkte Folge aus der Downward-Closure-Eigenschaft, nach der alle kürzeren Teilkombinationen von X' (niedrige Ebene) gleich häufig oder häufiger sind als eine betrachtete Kombination X (höhere Ebene). Daraus lässt sich der Schluss ziehen, dass sich die gesuchten Top-N-Ergebnisse mit erhöhter Wahrscheinlich-

keit nahe dem Wurzelknoten befinden. Sie enthalten somit entsprechend wenige Elemente. Wird folglich ein tiefer Abstieg im Suchraum vermieden und lediglich die Umgebung des Wurzelknotens betrachtet, können wahrscheinlich ein Großteil der endgültigen Top-N-Ergebnisse ermittelt werden. Ein solches Vorgehen kann die unnötige Analyse langer Kombinationen vermeiden, da diese mit großer Wahrscheinlichkeit durch nachfolgende kürzere Kombinationen aus der Top-N-Ergebnismenge verdrängt werden.

Eine Heuristik zur Entscheidung über einen tieferen Abstieg im Kombinationsbaum muss beachten, dass nur wenige Zusagen über die nachfolgenden Knoten möglich sind. Die Kombinationen einer niedrigeren Ebene können sowohl häufiger als auch seltener als die Kombinationen höherer Ebenen sein. Dieser Fall kann eintreten, wenn beim Abstieg auf längere Kombination in zwei verschiedenen Ästen die jeweilige Transaktionsmenge pro Schritt verschieden stark reduziert wird. Entsprechend können im Laufe der Verarbeitung auch häufigere als die bereits ermittelten Kombinationen gefunden werden. Ein Beispiel hierfür sind die Kombinationen *BC:5* und *A:4* in Tabelle 3.1. Beides gilt sowohl für Breiten- als auch Tiefensuchen einer EHK.

Das Problem kann somit auf die Entscheidung vereinfacht werden, ob die Suche nach der Betrachtung eines Knotens bei dessen Kind- oder Nachbarknoten fortgesetzt werden soll. Dies ist im Vorfeld nicht eindeutig entscheidbar, ohne die jeweils folgenden Kombinationen und deren Häufigkeiten bereits zu kennen. Außerdem kann im Falle einer falschen Wahl nicht effizient zu vorherigen, bereits verarbeiteten Knoten zurückgesprungen werden. Deren zuletzt bekannten Zustände müssten hierfür vorgehalten werden. Bei sehr vielen Knoten und einer großen Anzahl jeweils zu speichernder Transaktionen ist das nicht möglich. Für einen Rücksprung ist andernfalls eine wiederholte Evaluation dieses Astes oder eine direkte Suche der Kombination im vollständigen Datenbestand nötig, was die Laufzeit der EHK deutlich erhöht.

Die Strategie von FASTINC besteht darin, die Suche mittels TOP-N-ECLAT durchzuführen und dabei unnötig tiefe Betrachtung von Ästen zu vermeiden, sollten diese wahrscheinlich keine weiteren Top-N-Kombinationen enthalten. Kombinationen, welche durch nachfolgend ermittelte, kürzere und häufigere Kombinationen aus dem endgültigen Ergebnis verdrängt werden, sollen nicht betrachtet werden. Die Entscheidung über einen weiteren Abstieg entlang eines Pfades wird heuristisch entschieden. Dadurch können korrekte Ergebnisse im Falle einer falschen Entscheidung übersprungen werden. Entsprechend werden auch Ergebnisse als Kandidaten deklariert, welche das finale Top-N-Ergebnis nicht erreichen. Dieser Arbeitsschritt nutzt die Konzepte von Branch-and-Bound-Algorithmen, liefert jedoch durch eine optimistische Einschränkung des Suchraumes nicht garantiert das vollständige Ergebnis.

Die Häufigkeiten der in dieser Vorverarbeitung gefundenen Kombinationen ist bereits korrekt bestimmt. Sie entspricht den Häufigkeiten der entsprechenden Kombinationen bei einer vollständigen Suche. Da die richtigen Häufigkeiten ermittelt werden, aber nicht alle Ergebnisse bestimmt wurden, muss die N-te Kombination der Vorverarbeitung seltener

oder gleich häufig der endgültig N-ten Kombination sein. Sie kann somit als unterer Grenzwert lb der Häufigkeit der korrekten N-ten Kombination verwendet werden.

Da ein solcher approximierter Schritt keine häufigeren Kombinationen findet, als sie bei einer vollständigen Suche ermittelt werden, sind alle Kombinationen X des Datenbestandes als Vorauswahl möglich. Die erreichbare untere Grenze ist jedoch signifikant von der gewählten Kandidatenmenge abhängig. Im Rahmen dieser Arbeit werden drei Heuristiken untersucht, nach welchen die Entscheidung über eine weitere Evaluation eines Pfades und damit die Wahl der Ergebniskandidatenmenge entschieden werden kann. Als Abbruchkriterium einer Suche werden folgende Ansätze vorgeschlagen:

1. Die aktuelle Kombinationslänge erreicht eine vordefinierte Maximallänge mpl.

2. Die Häufigkeit der aktuell betrachteten Kombination X erreicht nicht einen Wert $support(X) \geq \delta \cdot support(e)$ für ein vordefiniertes $0 \leq \delta \leq 1$ und der nächst selteneren 1-Kombination e in E.

3. Der Rang der aktuellen Kombination erreicht nicht mindestens den Rang $N \cdot \gamma$ für ein vordefiniertes $0 \leq \gamma \leq 1$.

Die vorgestellten Möglichkeiten nutzen die Eigenschaft eines Kombinationsbaumes, dass häufige Kombinationen nicht gleichmäßig verteilt im Suchraum vorliegen. Die Häufigkeit einer Kombination sinkt meist mit steigender Ebene innerhalb des Baumes und mit abnehmender Häufigkeit der 1-Kombinationen (oder im Falle von ECLAT von links nach rechts im Kombinationsbaum). Die drei vorgeschlagenen Strategien sollen im Folgenden näher betrachtet werden.

1. Die erste Möglichkeit definiert eine statische Grenze, um einen Großteil der korrekten Top-N-Kombinationen zu ermitteln. Eine solche Implementierung beendet die Suche nach häufigen Kombinationen beim Erreichen einer nutzerspezifischen Maximallänge der Kombination. Diese Art der Begrenzung ist unabhängig von den untersuchten Datenbeständen. Sie ist einfach zu implementieren und benötigt keine komplexen Berechnungen zur Entscheidung über den weiteren Abstieg.

   Die Definition der maximalen Kombinationslänge mpl ist für jede Suche nach häufigen Kombinationen dynamisch wählbar. Beispielsweise kann mpl in Abhängigkeit von der Anzahl der betrachteten Elemente oder der durchschnittlichen Transaktionsbreite gewählt werden. In vielen Fällen ist ein sinnvoller Wert jedoch nur schwer bestimmbar, da beispielsweise Wissen über Abhängigkeiten zwischen den untersuchten Elementen benötigt wird. Außerdem können sich ungünstige Häufigkeitsverteilungen der Einzelelemente signifikante auf das Algorithmenverhalten auswirken. Die Erlangung von Wissen über Datenbesonderheiten und Abhängigkeiten ist jedoch ein wichtiges Ziel der EHK. Die Optimierung der EHK ist somit

von den durch sie zu bestimmenden Ergebnissen abhängig, welche vor der Suche zwangsläufig noch unbekannt sind.

Eine Schätzung der Abhängigkeiten ist möglich, aber ungenau. Zusätzlich bewirkt sie instabile und schwer vorhersagbare Laufzeiten. Das in dieser Arbeit betrachtete Ziel einer robusten Anwendung kann nur bedingt erreicht werden, auch wenn im Idealfall eine Verbesserung möglich ist. Wie die Evaluation dieses Verfahrens zeigen wird, stellt $mpl = 2$ bei Verkaufsdaten jedoch bereits einen sinnvollen Wert dar. Bei diesen Daten sind Abhängigkeiten wenig ausgeprägt und die daraus entstehenden Kombinationen enthalten nur wenige Elemente. Die Top-N-Kombinationen sind somit meist kurz. Außerdem werden mit der Beschränkung von $mpl = 2$ bereits $\binom{|E|}{2}$ wahrscheinliche Kandidaten überprüft. Dies kann bei ausreichend großer Kardinalität von E die Top-N-Kandidaten füllen und eine sinnvolle Grenze lb bilden.

2. Die zweite Bedingung arbeitet dynamischer als die erste Optimierungsmöglichkeit und passt sich an die Besonderheiten der jeweilig untersuchten Datenbestände an. Hierbei wird jeder Ast traversiert, bis die Häufigkeit der aktuell betrachteten Kombination unter einem nutzerdefinierten Faktor δ der Häufigkeit der nächsten 1-Kombination fällt. Die Häufigkeiten aller Einzelelemente sind bei den meisten Verfahren zur EHK aus einem Vorverarbeitungsschritt bekannt und können somit ohne Zusatzkosten in die Optimierung der Suche einbezogen werden.

   Aufgrund der Tatsache, dass alle Kombinationen unterhalb dieses betrachteten Knotens seltener sind als der Elternknoten, erfolgt eine Abschätzung. Diese ermittelt, wie wahrscheinlich der aktuelle Ast weitere endgültige Top-N-Ergebnisse erzeugen kann. Ein derartiges grobes Beschneiden des Suchraumes führt allerdings schnell zu vielen verworfenen Top-N-Kandidaten und verschlechtert im Endeffekt die Güte des Grenzwertes lb. Außerdem wird die Kandidatenliste bei dieser Optimierung vergleichsweise langsam gefüllt, wodurch lb nur langsam wächst.

   Der Faktor δ stellt ein zusätzliches Problem dar, da dieser deutlich auf das Verhalten des Algorithmus einwirkt. Ein Anwender muss diesen explizit wählen oder auf möglicherweise unpassende Standardvorgaben zurückgreifen. Der Algorithmus ist insgesamt in seiner Anwendung wenig robust, wenn die Häufigkeiten der Einzelelemente ungünstig verteilt sind. Für diese Arbeit ist diese Optimierung somit nicht vertretbar, da trotz einer potenziell guten Ergebnismengenschätzung die Gefahr deutlicher negativer Auswirkungen im Falle einer Fehleinschätzung besteht.

3. Die dritte Heuristik geht davon aus, dass eine finale Top-N-Kombination bereits beim Eintritt in die Ergebniskandidatenmenge eine gute Platzierung erreicht. Wird eine Kombination schon bei der Aufnahme in die Kandidatenmenge schlecht platziert, wird sie sehr wahrscheinlich im Laufe der noch folgenden Kombinationen wieder aus dem Endergebnis verdrängt. Aus solchen Kombinationen abgeleitete, weitere Top-N-Kandidaten werden noch unwahrscheinlicher Teil des Endergebnis-

ses. Diese sind immer nur gleich häufig oder seltener als der aktuell betrachtete Kandidat. Sie erreichen somit das Top-N-Ergebnis nicht, wenn die sie erzeugende Kombination bereits zu selten ist.

Sowohl die Güte des Ergebnisses als auch das Laufzeitverhalten des Verfahrens hängt signifikant von der Wahl des Parameters γ ab. Sind die Anforderungen an eine benötigte Platzierung zu hoch, werden viele Knoten und deren Kinder verworfen. Entsprechend wenige Kombinationen werden in die Liste der Ergebniskandidaten übernommen. Somit ändert sich die Güte der Ergebniskandidaten nur geringfügig und der jeweils verwendete *minSupport* wächst nur langsam. Entsprechend negativ wirkt sich dies auf die Laufzeit des Algorithmus aus.

Im Falle einer zu geringen Anforderung an die Ergebniskandidaten verhält sich der Algorithmus ähnlich einer nicht optimierten Variante. Es werden nur wenige unnötige Kombinationen vermieden und das Verfahren erbringt geringen Laufzeitgewinn. Insgesamt kann eine verwendete Optimierung bei geschickter Parametrisierung schnell einen guten Wert für *lb* erzeugen. Sie neigt im Falle einer ungünstigen Wahl von γ aber zu einem schlechten Laufzeitverhalten. Das Ziel einer robusten, einfachen Verarbeitung kann daher nicht erreicht werden.

Diese drei Möglichkeiten einer Vorverarbeitung können ebenfalls in Kombination durch mehrfache Iteration zur Anhebung von *lb* beitragen. Dies ermöglicht die Nutzung von verschiedenen Vorteilen der einzelnen Heuristiken. Da pro Schritt der jeweils aktuell höchste Wert von *lb* verwendet wird, kann sich dieser Grenzwert lediglich erhöhen und sich damit näher an den korrekten Wert angleichen. Der Berechnungsaufwand steigt damit an. Bei einer lediglich geringen Anhebung von *lb* pro Schritt überschreiten die Kosten leicht den möglichen Gewinn. Zur Verbesserung der vollständigen Algorithmenlaufzeiten ist dies somit nur sinnvoll, wenn ein gering erhöhtes *lb* die Algorithmenlaufzeit signifikant reduziert. Für die Analyse von Verkaufsdaten ist dies nicht der Fall. Hier hat sich Versuchen mit realistischen Datenbeständen und Anfragen die Beschränkung auf eine vordefinierte Maximallänge *mpl* als effizienteste und damit sinnvollste Lösung herausgestellt.

3.2. Verteilte Bestimmung von Top-N-Kombinationen

Moderne Systemarchitekturen bestehen oft aus einem Verbund vergleichsweise schwacher Einzelrechner. Die Leistungsfähigkeit dieser Verbünde kann aber der von Großrechnern entsprechen. Die Kosten sind hingegen durch den Verzicht auf besonders leistungsfähige Einzelkomponenten deutlich günstiger, als dies bei Großrechnern möglich ist. Diese Rechnerverbünde können gemeinsam Problemstellungen lösen, welche für einen einzelnen Rechner nicht möglich sind. Entsprechend werden Daten oft ebenfalls verteilt gespeichert. Außerdem werden Datenbestände zur besseren Verwaltung der Daten oder zur effiziente-

ren Verarbeitung durch Datenlokalität verteilte abgelegt. Um diese Bestände verarbeiten zu können, ist eine parallele Anfrageverarbeitung eine sinnvolle und notwendige Anforderung an einen Algorithmus zur Datenverarbeitung. Dieser Abschnitt soll sich mit Verfahren zur Suche von häufigen Kombinationen in diesen Datenbeständen befassen. Im allgemeinen Fall können nur wenige oder keine Annahmen über die Art und Weise der Aufteilung der betrachteten Daten getroffen werden. Außerdem kann die Partitionierung durch vielfältige Gründe des Anwenders verändert werden, ohne die Anforderungen der EHK zu berücksichtigen. Die Behandlung dieser Problemstellung ist Inhalt der folgenden Abschnitte. Vorerst soll jedoch ein Szenario mit gleichmäßig aufgeteilten Daten betrachtet werden. In den späteren Abschnitten wird der nachfolgende Ansatz auf Szenarien mit ungleichmäßigen Partitionsgrößen erweitert.

Eine grundsätzliche Methode zur Verarbeitung partitionierter Datenbestände besteht in einer Umverteilung oder Vereinigung der Daten. Auf diese Weise können beispielsweise die Aufgabe in disjunkte, parallel zu berechnende Teilbereiche zerlegt werden. Diese Möglichkeit besteht in der Praxis oft nicht, da dies durch semantische oder syntaktische Partitionierungen der Daten durch den Nutzer verhindert wird. Das kann beispielsweise eine Unterteilung nach spezifischen Attributen wie Jahr oder Filiale sein. Eine Umverteilung zerstört diese vom Anwender bewusst gewählten Datencharakteristiken, ohne diesen darüber zu informieren. Um Unstimmigkeiten zu vermeiden, sollten entsprechend sowohl die physische Datenablage als auch das Datenschema nach Möglichkeit unverändert bleiben. Des Weiterem können Datensicherheitsgründe eine Umverteilung oder Replikation verhindern [LZCZ07]. Beispielsweise sind durch Anforderungen der Datensicherheit nicht alle Daten filial- oder länderübergreifend kopierbar. Analysiert werden müssen sie jedoch gemeinsam. Entsprechend kann auch ein Algorithmus zur Bestimmung der N häufigsten Kombinationen nicht auf die Fähigkeit einer verteilten Suche verzichten. Nur durch Anpassung des Algorithmus an die zu verarbeitenden Daten kann dieser in einer Vielzahl von Szenarien verwendet werden.

In dieser Arbeit soll als einzige Anforderung bestehen, dass die Daten im Falle der Warenkorbanalyse nach Transaktionsschlüssel verteilt vorliegen. Das heißt, alle Einträge eines Warenkorbes müssen in der gleichen Partition gespeichert werden, um gemeinsam analysiert werden zu können. Die Verletzung dieser Bedingung würde vor jeder Analyse eine Umverteilung der Daten erfordern, wodurch ein deutlicher Mehraufwand entsteht. Für die Top-N-EHK werden in dieser Arbeit keine weiteren Forderungen gestellt.

Ein naheliegender Ansatz zur Lokalisierung von globalen Top-N-Ergebnissen auf partitionierten Daten ist die Suche nach allen jeweils lokalen Top-N-Kombinationen pro Partition und deren Vereinigung zu einem globalen Top-N-Ergebnis. Dieser Ansatz scheitert jedoch, wenn keine Angaben über die Daten und deren Eigenschaften vorhanden sind. Weisen alle Partitionen sehr verschiedene lokale Ergebnisse auf, können diese Kandidaten insgesamt selten sein. Eine globale Top-N-Kombination muss hingegen jeweils nicht ein lokales Top-N-Ergebnis erreichen. Entsprechend wird sie kein Kandidat und nicht Teil des Endergebnisses.

Als Beispiel soll ein Kleidungsproduzent dienen, der seine Verkaufsdaten nach Winter und Sommer partitioniert speichert. Bei einer Suche nach den Top-2 seiner wichtigsten Produkte ermittelt er für die Sommermonate die Produkte *T-Shirts* und *Badehose*, für die Wintermonate *Mützen* und *Mäntel*. Eine Vereinigung dieser beiden Teilergebnisse würde nur diese vier Produkte als Kandidaten für die Top-2 des Gesamtjahres vorschlagen. Da jedoch sowohl im Sommer als auch im Winter beispielsweise *Jacken* verkauft wurden, können für diese insgesamt mehr Verkäufe als für die vier ermittelten Top-2 Kandidaten vorliegen. Die *Jacken* als häufigstes Element werden nicht korrekt ermittelt. Dieser Umstand ist in der Praxis wahrscheinlich selten anzutreffen. Das Verfahren kann also in vielen Fällen bereits ein sinnvolles Ergebnis liefern. Der jeweilige Anwender muss sich des potenziell unzureichenden Ergebnisses aber bewusst sein und die Qualität einschätzen können.

Im Rahmen dieser Arbeit werden zur Lösung dieses Problems die Algorithmen PARTITION [SON95] und TPUT [CW04] kombiniert und auf die Besonderheiten einer Top-N-EHK angepasst. Die Algorithmen wurden in den Abschnitten 2.2.5 und 2.2.6 bereits detailliert vorgestellt.

Zusammengefasst arbeitet der Algorithmus PARTITION zur Bestimmung verteilter häufiger Kombinationen wie folgt:

1. Bestimmung alle lokalen Ergebnisse pro Partition. Nach Savasere [SON95] kommt jedes globale Ergebnis in mindestens einer Partition vor, wodurch sich als globale Ergebniskandidatenmenge die Vereinigung aller lokalen Ergebnisse eignet. Die Häufigkeiten der einzelnen Kombinationen können bis zu diesem Punkt unvollständig sein.

2. Ermittlung fehlender lokaler Häufigkeiten.

3. Vereinigung der Ergebnisse von Schritt 1 und 2 zur Bildung und Ausgabe des Endergebnisses.

Der TPUT-Algorithmus wendet dieses Prinzip auf eine verteilte Aggregation an. Allerdings liegt das Augenmerk bei TPUT verstärkt auf einer minimalen Datenübertragung. Die eigentliche Aggregation wird als einfache und sehr schnelle Operation vorausgesetzt. Deren benötigte Laufzeit wird gegenüber der Laufzeit zur Datenkommunikation als unbedeutend angenommen. Da TPUT die zu versendende Datenmengen optimiert, muss der Großteil der benötigten Laufzeit zwingend durch die Datenübertragung zwischen den einzelnen Knoten erzeugt werden. Ist dies nicht gegeben, ist TPUT nicht effizient.

TPUT bestimmt ähnlich wie PARTITION lokale Ergebnisse pro Partition. Der Algorithmus ermittelt jedoch anders als PARTITION die lokalen Top-N-Ergebnisse. Diese können noch unzureichend sein, ermöglichen aber vereinigt eine Aussage über eine globale unte-

re Grenze der vollständigen, globalen Top-N-Aggregation. Durch einen zweiten Suchlauf wird mit Hilfe dieses Grenzwertes das korrekte endgültige Top-N-Ergebnis ermittelt.

Die Vorgehensweise der beiden Verfahren ist zueinander sehr ähnlich. Eine einfache Vereinigung zu einer verteilten Top-N-EHK ist jedoch problematisch. Die ersten Suchschritte von TPUT sind einfach, wenn die lokalen Aggregationen als sehr schnell definiert sind. Auch die zweite Aggregation zur Bestimmung des vollständigen globalen Ergebnisses ist bei TPUT schnell. Die entstehenden Kosten sind somit im Vergleich zu einer unverteilten Aggregation vertretbar, wenn der Gewinn durch eine gesteigerte Rechenleistung bei mehreren Rechenknoten beachtet wird. Übertragen auf eine verteilte Top-N-EHK müssen lokale Suchen ebenfalls pro Partition zweifach ausgeführt werden. Einmal als lokale Top-N-Suche und im zweiten Durchlauf als Suche mit einer aus dem ersten Durchlauf ermittelten Untergrenze u für die Mindesthäufigkeit.

Da für TPUT die Kosten der lokalen Suchen als unbedeutend angenommen werden, ist einer wiederholten Suche unproblematisch. Die gesenkten Kosten einer optimierten Datenübertragung rechtfertigen den entstehenden Mehraufwand. Sind, wie im Falle der EHK, lokale Suchen sehr aufwändig, werden mehrfache lokale Suchen nicht kompensiert und die Gesamtlaufzeit des Verfahrens steigt an. Der Laufzeitanteil der lokalen Suchen dominiert damit anderes als bei TPUT die Berechnung einer verteilten EHK. Die Kosten der Datenübertragung sind hingegen im Vergleich meist unbedeutend. Eine Verdopplung dieser Suchen wirkt sich somit signifikant auf die Gesamtlaufzeit aus.

Der zweite Arbeitsschritt von TPUT, eine wiederholte Suche mit Grenzwert, entspricht bei der EHK einer Suche mittels PARTITION. Als global definierte Mindesthäufigkeit wird der Grenzwert u genutzt. Fällt dieser im ersten Schritt niedrig aus, ergeben sich extrem große Laufzeiten der EHK. Dies ist in der Praxis ein wahrscheinliches Szenario, da hier Partitionen mit sehr unterschiedlichen Eigenschaften auftreten können, wodurch u sinkt. Nach Zaki [Zak99] wächst die Anzahl möglicher Kombinationen exponentiell mit sinkender Mindesthäufigkeit. Die bei niedriger Mindesthäufigkeit erzeugten großen lokalen Ergebnismengen erhöhen den Ressourcenverbrauch der EHK deutlich. Im gleichen Maße wird die Algorithmenlaufzeiten der EHK bei geringen Mindesthäufigkeiten gesteigert.

Die verteilte Aggregation mittels TPUT besitzt vergleichbare Probleme, wobei diese durch die Annahme einer jederzeit schnellen lokalen Aggregation auch bei geringen Grenzwerten durchführbar ist. Eine direkte Verwendung von TPUT für eine verteilte EHK ist somit nicht sinnvoll, wenn die Verfahren einfach verwendbar sein sollen.

Zur Behebung dieses Umstandes soll in dieser Arbeit angenommen werden, dass bei einer EHK meist keine vollständig korrekten Top-N-Ergebnisse benötigt werden. Die Suche nach neuen Zusammenhängen erfolgt oft ohne genaue Vorstellungen über das gewünschte Ergebnis und ist geprägt durch ein prinzipielles Interesse an dem in den Daten enthaltenem Wissen. Wenn ein unvollständiges Ergebnis verwendbares, neues Wissen erbringt,

bleibt der Wert einer Suche erhalten. Die einfache Anwendung erhöht jedoch die Anzahl der möglichen Nutzer. Insgesamt kann auf diese Weise die Menge an Informationen, und damit der Wert der vorhandenen Daten, gesteigert werden. Die Eigenschaft von TPUT, ein vollständig korrektes Ergebnis zu bestimmen, kann somit zu Gunsten eines robusten Laufzeitverhaltens gelockert werden.

Mit Hilfe dieser Annahme wird in dieser Arbeit ein Algorithmus namens *Save-Threshold (STH)* eingeführt, welcher ein laufzeitstabiles und effizientes Suchen von Top-N-Kombinationen auf verteilten Daten ermöglicht. STH erzeugt eine möglicherweise unvollständige, jedoch in ihrer Qualität bewertbare Ergebnismenge. Zum Vergleich soll ein Algorithmus mit dem Namen *TPARTITION* vorgestellt werden, welcher die direkte Übertragung des Konzeptes von TPUT auf PARTITION darstellt und ein vollständiges verteiltes Top-N-Ergebnis ermittelt. Wie bereits erwähnt, kann TPARTITION dabei eine für den Nutzer unerwartet lange Laufzeit aufweisen und ist entsprechend schwierig anzuwenden.

3.2.1. Approximierte Top-N-Ergebnisse mit STH

Der Ablauf von STH, dargestellt in Abbildung 3.3, entspricht im Wesentlichen dem Vorgehen einer verteilten EHK mit PARTITION. Der Unterschied besteht vor allem darin, dass statt Kombinationen über einer lokalen Mindesthäufigkeit die lokalen Top-N-Ergebnisse pro Partition gesucht werden. Die für das globale Top-N-Ergebnis ermittelten lokalen Kandidaten werden anschließend vereinigt. Von Kandidaten, welche nicht auf allen Partitionen Teil des Top-N-Ergebnisses waren, sind einige Teilhäufigkeiten noch nicht vorhanden. Diese müssen nachträglich durch eine Suche auf den fehlenden Partitionen bestimmt werden.

STH wird mit einer Mindesthäufigkeit $minSupportAbs = 1$ initialisiert und liefert ein lokal vollständiges, aber global approximiertes Ergebnis. Es wird nicht garantiert, dass alle globalen Top-N-Ergebnisse gefunden werden. Unter Verwendung von STH können jedoch ausgehend von den lokalen Top-N-Ergebnissen meist ein Großteil der gewünschten globalen Top-N-Ergebnisse gefunden werden. Es besteht eine hohe Wahrscheinlichkeit, dass ein globales Top-N-Ergebnis auf mindestens einer Partition ebenfalls Teil des lokalen Top-N-Ergebnisses ist. Damit wird für diese Kombinationen die globale Häufigkeit auf allen Partitionen bestimmt und sie erreichen das globale Top-N-Ergebnis. Die Häufigkeiten der gefundenen Kombinationen entsprechen den korrekten Werten. Einige globale Top-N-Kombinationen können möglicherweise nicht ermittelt werden, da sie keines der lokalen Top-N-Ergebnisse erreichen. Solche global nicht ermittelten Kombinationen werden im Endergebnis von anderen Kombinationen ersetzt, die nicht Teil der korrekten globalen Top-N-Kombinationen sind. Diese fälschlich bestimmten Ergebnisse treten jedoch im Vergleich zu den korrekten Ergebnissen meist ebenfalls oft auf und repräsentieren daher potenziell ähnlich wichtige Informationen.

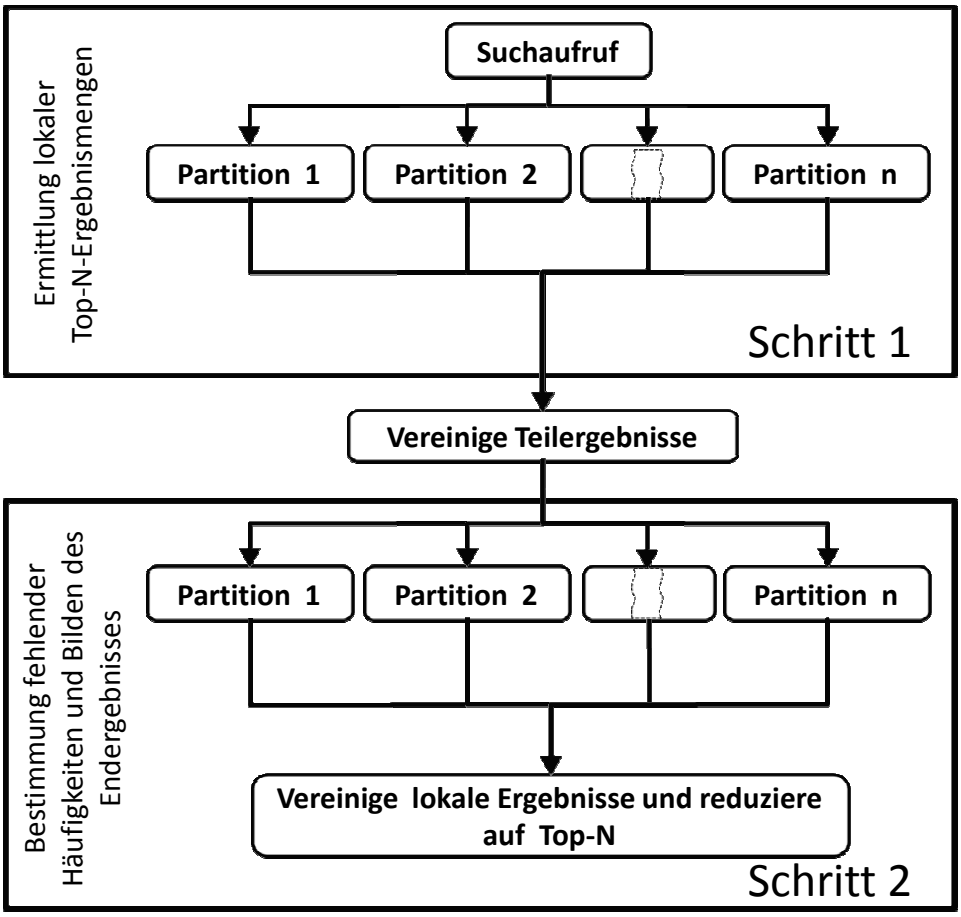


Abbildung 3.3.: Verteilte Suche nach Top-N häufigen Kombinationen mit STH

Abbildung 3.4 veranschaulicht ein durch STH ermitteltes Ergebnis. Die lokalen Top-N-Ergebnisse aller Partitionen werden konsolidiert und als globale Top-N-Kandidaten betrachtet. Alle fehlenden Häufigkeiten eines solchen Kandidaten werden äquivalent zum Vorgehen von PARTITION auf den entsprechenden Partitionen nachträglich ermittelt. Die vollständigen, globalen Häufigkeiten aller vorher ermittelten Kandidaten können nachfolgend bestimmt werden. Nach Häufigkeit sortiert bilden die häufigsten N Kombinationen das Endergebnis.

Durch Bestimmung eines Grenzwertes für möglicherweise nicht erfasste Kombinationen kann das globale Top-N-Ergebnis in einen garantierten und einen unsicheren Bereich geteilt werden. Bei welcher Häufigkeit diese Aufspaltung erfolgt, ist aus den lokalen Ergebnissen bestimmbar. Der folgende Abschnitt widmet sich der Bestimmung dieses Grenzwertes für garantierte Kombinationen bei der Verwendung von STH.

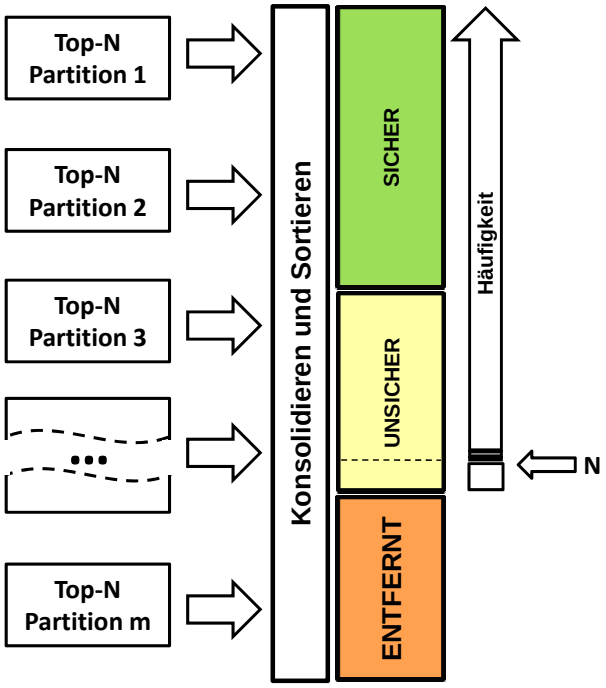


Abbildung 3.4.: Visualisierung des Save-Threshold

3.2.2. Qualitätsaussagen bei Verwendung von STH

Wie im vorherigen Abschnitt bereits erläutert wurde, sind die ermittelten Häufigkeiten aller durch STH gefundenen Kombinationen korrekt. Eine Top-N-Kombination wird somit entweder mit den richtigen Häufigkeiten ermittelt oder nicht gefunden. Unter diesen Bedingungen ist die Bestimmung einer Häufigkeit s möglich, welche eine nicht gefundene Kombination maximal erreichen kann. Im Umkehrschluss werden alle Kombinationen X mit $support(X) > s$ vollständig und mit korrekten Häufigkeiten ermittelt. Die Ergebnisse oberhalb dieses Schwellwertes können somit garantiert werden. Eine nicht ermittelte Kombination kann nicht häufiger und damit besser platziert sein. Aus dem Argument der Vollständigkeit ist ebenfalls ersichtlich, dass alle Kombinationen X mit der Häufigkeit $support(X) > s$ durch STH ermittelt werden.

Sind die per STH ermittelten Ergebnisse ausreichend gut, können sie bereits eine Vielzahl der vom Anwender erwarteten Erkenntnisse erbringen. Die potenziell nicht zu den häufigsten N Kombinationen gehörenden Kombinationen X mit $support(X) \leq s$ sind außerdem ebenfalls mit korrekter Häufigkeit bestimmt. Sie erlauben eine Weiterverarbeitung zu Assoziationsregeln, ohne falsche Erkenntnisse zu beschreiben. Solche Regeln sind lediglich nicht Teil des exakten Ergebnisses. Wird der Wert einer Kombination durch deren Häufigkeit repräsentiert, sind fälschlicherweise nicht ermittelten Kombinationen nur geringfügig wertvoller als die sie ersetzenden Kombinationen. Um Aussagen treffen zu

können, wie viele Kombinationen durch STH maximal nicht gefunden wurden, besteht die Herausforderung in der Bestimmung der Grenze s für garantierte Kombinationen.

Sei W die häufigste Kombination, welche nicht ermittelt wird. W kann nur im globalen Ergebnis verfehlt werden, wenn es auf keiner Partition $P_{1...p}$ Teil der lokalen Top-N-Ergebnisse war. $Q_{1...p}$ seien die jeweils seltensten gefundenen Kombinationen pro Partition. Eine von keiner Partition als Kandidat gelieferte Kombination kann nicht häufiger sein als die Summe aller $supportAbs(Q)$. Wäre sie doch häufiger, müsste die Kombination mindestens auf einer Partition i das dortige $supportAbs(Q_i)$ überschreiten. Damit wäre diese Kombination Teil des Ergebnisses der Partition i und somit Teil der globalen Ergebniskandidaten. Die maximal mögliche globale Häufigkeit von W entspricht daher dem Grenzwert s. Für einen partitionierten Datenbestand DDB gilt:

$$support(W) = \frac{\sum_{i=1}^{p} (supportAbs(Q_i) - 1)}{|DDB|} = s.$$

Damit können global gefundenen Kandidaten X mit einer Häufigkeit $support(X) > s$ garantiert werden. Durch Nachsuchen von fehlenden Häufigkeiten auf einzelnen Partitionen sind ebenfalls die korrekten globalen Häufigkeiten der Kombinationen bekannt. Da keine nicht ermittelte Kombination häufiger sein kann, besitzen sie ebenfalls die richtige Platzierung innerhalb der Ergebnismenge.

Die Suchen auf jeder Partition können zusätzlich auf die lokalen Top-$(N \cdot \alpha)$-Kombinationen (für $\alpha > 1$) erweitert werden, um das Ergebnis zu verbessern. Dadurch kann auf einigen Partitionen i die Häufigkeit $supportAbs(Q_i)$ gesenkt werden, wodurch die Grenze s insgesamt gesenkt wird. Durch diese Ausweitung auf neue Kandidaten können potenziell zusätzliche Ergebnisse garantiert werden. Die Menge der lokalen Kandidaten wird außerdem bei vergleichsweise geringem Mehraufwand erweitert. Dadurch werden weitere globale Top-N-Kandidaten betrachtet und ausgewertet. Diese Vergrößerung der Kandidatenmenge und die damit verbundene Absenkung von s können Ergebnisse garantieren, welche im regulären Fall von $\alpha = 1$ nicht sicher sind.

Der Mehraufwand einer solchen Berechnung wächst, wie in der nachfolgenden Evaluation gezeigt, etwa linear mit α und ist somit für kleine Faktoren gering. Die Funktion von α ähnelt dem gleichnamigen Parameter von TPUT. Wie die nachfolgende Evaluation ebenfalls zeigt, sind oft bereits Werte von $1.05 \leq \alpha \leq 1.20$ (dies entspricht $5\% - 20\%$ zusätzliche lokale Ergebnisse) für Verkaufsdaten ausreichend. Wie in der Evaluation dieses Kapitels erkennbar wird, sind die Ergebnisse von STH bei vielen Szenarien ab $\alpha \approx 1.05$ bereits vollständig garantiert. Die optimale Wahl von α ist jedoch abhängig von Charakteristiken der untersuchten Daten und den Suchparametern der EHK. Hierfür ist Fachwissen und Nutzerinteraktion nötig und die Anwendung wird erschwert. Garantien in Abhängigkeit von α können nicht angegeben werden, wobei die Qualität des Ergebnisses aber stets mit α ansteigt.

Partition 1	
TID	**Elemente**
T01	a b c d
T02	c
T03	d
T04	d
T05	d
T06	d

Partition 2	
TID	**Elemente**
T07	b c d
T08	d
T09	a b c
T10	a b c
T11	a b c d
T12	c

Tabelle 3.2.: Partitionierter Transaktionsdatenbestand DDB

Beispiel:

Als Ziel sollen in dem verteilten Datenbestand aus Tabelle 3.2 die Top-2 häufigsten Kombinationen mit Hilfe von STH ermittelt werden. Zum einfacheren Verständnis soll an dieser Stelle eine absolute Häufigkeit für den Wert von s verwendet werden. Dabei wird der folgende Ablauf durchgeführt:

1. Auf allen Partitionen wird eine Suche nach den lokalen Top-2 Kombinationen durchgeführt, welche $I_1 = [D:5, C:2]$ und $I_2 = [C:5, B:4]$ zurückliefern.

2. Vereinigt ergeben sich als Ergebniskandidaten $I' = I_1 \cup I_2 = [C:7, D:5, B:4]$. Auf der ersten Partition fehlt die Häufigkeit von B und auf der zweiten Partition von D. Die jeweils kleinsten lokalen Häufigkeiten sind 2 und 4, womit sich eine obere Grenze nicht gefundener Kombinationen von $s = (2-1) + (4-1) = 4$ errechnet.

3. Durch Nachsuchen von B auf der ersten und D auf der zweiten Partition werden $[D:3]$ und $[B:1]$ ermittelt.

4. Die Vereinigung der Ergebnisse vom zweiten und fünften Schritt erzeugen die global häufigen Kombinationen I

$$\begin{aligned} I &= [C:7, D:5, B:4] \cup [D:3] \cup [B:1] \\ &= [\mathbf{D{:}8,\ C{:}7}, B:5] \end{aligned}$$

5. Reduziert auf die Top-2 Kombinationen wird $I = [D:8, C:7]$ gebildet.

6. Da $s = 4$ und $supportAbs(X) > s$ für alle $X \in I$ gilt, ist das ermittelte Ergebnis vollständig und garantiert.

Der ungünstigste Anwendungsfall für diesen Algorithmus ist ein verteilter Datenbestand, bei welchem jede Partition gegenüber allen anderen Partitionen ein disjunktes Teilergebnis erzeugt. Die gefundenen Kombinationen sind dabei lokal jeweils häufig, können global

jedoch selten auftreten. In diesem Fall kann ein großer Anteil der globalen Top-N nicht sicher ermittelt werden. Die verfehlten Kombinationen können lokal dünn besetzt über jeweils mehrere Partitionen verteilt sein, wodurch sie kein lokales Top-N-Ergebnis erreichen. Entsprechend werden sie auf keiner Partition ein Ergebniskandidat und somit nicht Teil des Endergebnisses. Das korrekte globale Top-N-Ergebnis wird in diesem Fall nicht ermittelt.

In der Praxis kann davon ausgegangen werden, dass interessante Kombinationen eine Allgemeingültigkeit aufweisen und zumindest auf einigen Partitionen gefunden werden. Von diesem Standpunkt können nahezu disjunkte Teilergebnisse sogar ein Hinweis auf einen zur Suche nach Assoziationsregeln ungeeigneten Datenbestand sein. Eine separate Suche nach Assoziationsregeln pro Partition erbringt in derartigen Szenarien möglicherweise wichtige Erkenntnisse, welche bei einer globalen Analyse nicht gefunden werden.

3.2.3. Mögliche Verbesserungen der Qualitätsaussagen

Im Rahmen dieser Arbeit wurden noch weitere Ansätze zum Verringern von s für STH untersucht. Ein erfolgversprechender Ansatz soll in diesem Abschnitt erläutert werden. Dabei erfolgt der Nachweis, dass dieser Ansatz nicht besser sein kann, als die im vorhergehenden Abschnitt eingeführte Schranke s. Der Ansatz basiert auf den von Savasere gezeigten Eigenschaften von PARTITION [SON95].

1. Wenn eine Kombination global eine relative Mindesthäufigkeit aufweist, erreicht sie auf mindestens einer Partition deren lokale relative Mindesthäufigkeit.

2. Daraus folgt, dass alle Kombinationen mit einer globalen relativen Häufigkeit gs oberhalb der maximalen lokalen Mindesthäufigkeit als Kandidat gefunden werden. Dies wird damit begründet, dass alle Partitionen die lokalen Kombinationen mit mindestens dieser relativen Häufigkeit ermittelt haben. Das entspricht einem PARTITION-Algorithmus mit $minSupport = gs$. Da PARTITION ein vollständiges Ergebnis liefert, werden auch alle Ergebnisse oberhalb von gs korrekt bestimmt.

Nach dem ersten Schritt von STH kann für alle Partitionen festgestellt werden, welche minimalen lokalen Häufigkeiten für $Q_{1...p}$ relativ zur jeweiligen Partitionsgröße gefunden wurden. Das Maximum aller relativen Häufigkeiten von $Q_{1..p}$ bildet die globale relative Grenze gs. Kombinationen mit dieser relativen Häufigkeit wurden auf allen Partitionen gesucht. Eine Kombination, welche eine globale Häufigkeit über dieser maximalen lokalen relativen Häufigkeit aufweist, muss somit gefunden werden.

$$gs = \max\left(\frac{supportAbs(Q_1) - 1}{|P_1|}, \ldots, \frac{supportAbs(Q_p) - 1}{|P_p|}\right)$$

Hierbei wird ein verteilter Datenbestand DDB mit den Partitionen $P_{1\ldots p}$ betrachtet. Alle Kombinationen X mit $support(X) > gs$ werden gefunden, da übertragen auf eine reguläre Anwendung von PARTITION auf jeder Partition alle Kombinationen mit der lokalen, relativen Häufigkeit über gs gesucht wurden. Damit werden mindestens die Ergebnisse von PARTITION mit $minSupport = gs$ geliefert. Wie Savasere für PARTITION zeigt, müssen die Ergebnisse oberhalb von gs vollständig und korrekt sein. Gegenüber dem in Abschnitt 3.2.2 vorgestellten absoluten Grenzwert s gilt jedoch stets $s \leq gs$. Somit können mit dessen Hilfe mindestens die von gs garantierten Kombinationen gesichert werden. Hierfür muss gezeigt werden:

$$\frac{\sum_{i=1}^{p}(supportAbs(Q_i)-1)}{|DDB|} \leq \max\left(\frac{supportAbs(Q_1)-1}{|P_1|},\ldots,\frac{supportAbs(Q_p)-1}{|P_p|}\right)$$

Im folgenden Nachweis soll neben den in den vorherigen Abschnitten für s eingeführten Gleichungen zusätzlich $support(Q_i) \geq 0$, $|P_i| > 0$ für alle Partitionen P_i von DDB und $i = 1\ldots p$, $|DDB| = \sum_{i=1}^{p}|P_i|$, und die lokalen, relativen Häufigkeiten $b_i = \frac{supportAbs(Q_i)-1}{P_i}$ gelten. Zum Beweis wird angenommen, dass $s \leq gs$ nicht allgemein gilt. Damit muss der folgende Fall existieren:

$$\sum_{i=1}^{p}(supportAbs(Q_i)-1) > gs \cdot |DDB|$$

Damit gilt

$$\begin{aligned}
\sum_{i=1}^{p}(supportAbs(Q_i)-1) &> gs \cdot |DDB| \\
\sum_{i=1}^{p}(supportAbs(Q_i)-1) &> gs \cdot \sum_{i=1}^{p}|P_i| \\
\sum_{i=1}^{p}(supportAbs(Q_i)-1) &> \sum_{i=1}^{p}(gs \cdot |P_i|) \\
\sum_{i=1}^{p}\frac{supportAbs(Q_i)-1}{|P_i|} \cdot |P_i| &> \sum_{i=1}^{p}(gs \cdot |P_i|) \\
\sum_{i=1}^{p}(b_i \cdot |P_i|) &> \sum_{i=1}^{p}(gs \cdot |P_i|)
\end{aligned}$$

Da $gs = \max(b_1 \ldots b_p)$ gilt, muss gs entsprechend gleich oder größer als jedes b_i sein. Durch einen Vergleich der Faktoren folgt, dass $b_i \cdot |P_i|$ stets gleich oder kleiner $gs \cdot |P_i|$ ist.

Die Summe über alle $b_i \cdot |P_i|$ ist somit ebenfalls stets kleiner oder gleich der Summe über alle $gs \cdot |P_i|$. Die Ungleichung ist somit nicht erfüllbar und die Annahme widerlegt. Damit wird gezeigt, dass die absolute Untergrenze s stets niedriger oder gleich der relativen unteren Grenze gs sein muss. Entsprechend ist eine Überprüfung der absoluten Grenze s stets ausreichend.

Sind die Häufigkeiten von nur wenigen der durch STH ermittelten Kombinationen nicht garantiert, ist das Endergebnis bereits für viele Szenarien verwendbar. In Abhängigkeit von den Anforderungen der jeweiligen Anwendung kann das Ergebnis jedoch bei zu wenigen garantierten Ergebnissen auch als unzureichend erachtet werden. Mit dem im nachfolgenden Abschnitt vorgestellten Algorithmus TPARTITION können die vollständig korrekten Ergebnisse ermittelt werden. Die durch STH ermittelten Informationen können zur Ausführung von TPARTITION verwendet werden. Eine einfache und laufzeitrobuste Anwendung ist dabei nicht sichergestellt. Sowohl die zu erwartende Ergebnismenge, als auch die Laufzeit sind für TPARTITION nur unter Annahme einer gleichmäßigen und für den Algorithmus günstigen Datenverteilung abschätzbar. Die intuitive Erwartungshaltung des Anwenders über das Verhalten des Algorithmus kann somit unerwartet deutlich verfehlt werden und die Verwendung des Verfahrens erschweren.

3.2.4. Vollständige Ergebnisse mit TPARTITION

Aus den vorherigen Abschnitten ist bekannt, dass mit Hilfe von STH das Ergebnis einer verteilten EHK approximiert ermittelbar ist. Für einige Anwendungen kann es jedoch zwingend notwendig sein, das vollständig korrekte Ergebnis zu bestimmen. Zur Lösung ist hierbei ein iterativer Prozess möglich. Damit können unter Nutzung von klassischen Verfahren wie PARTITION mit gezielter Parameterwahl bei einer verteilten EHK mindestens N Kombinationen ermittelt werden. Regelkreislaufverfahren können dies iterativ lösen, wobei die Anzahl der nötigen Suchen, und damit die Gesamtlaufzeit, nicht im Vorfeld abschätzbar ist und Über- bzw. Unterschätzungen möglich sind. In dieser Arbeit wird zur Bestimmung einer vollständig korrekten Top-N-EHK eine Erweiterung von STH vorgeschlagen. Mit Hilfe von STH soll eine Mindesthäufigkeit u bestimmt werden. Mit dieser werden, wenn vorhanden, garantiert N global häufige Kombinationen ermittelt. Abbildung 3.5 zeigt den vorgeschlagenen Ablauf im Überblick. Nachfolgend soll die Kombination von STH und PARTITION als *TPARTITION* bezeichnet werden.

Die Vorgehensweise von TPARTITION erfordert die Durchführung einer approximierten und einer vollständigen Suche und benötigt daher insgesamt mehr Ressourcen als STH. Außerdem kann der durch STH definierte Parameter $minSupport = u$ sehr niedrig bestimmt werden, wodurch der nachfolgende PARTITION-Algorithmus für einige Datenbestände extreme Laufzeiten aufweist. Die Anwendbarkeit des Verfahrens wird insgesamt deutlich eingeschränkt. Dem in dieser Arbeit adressierten Anspruch einer einfachen Be-

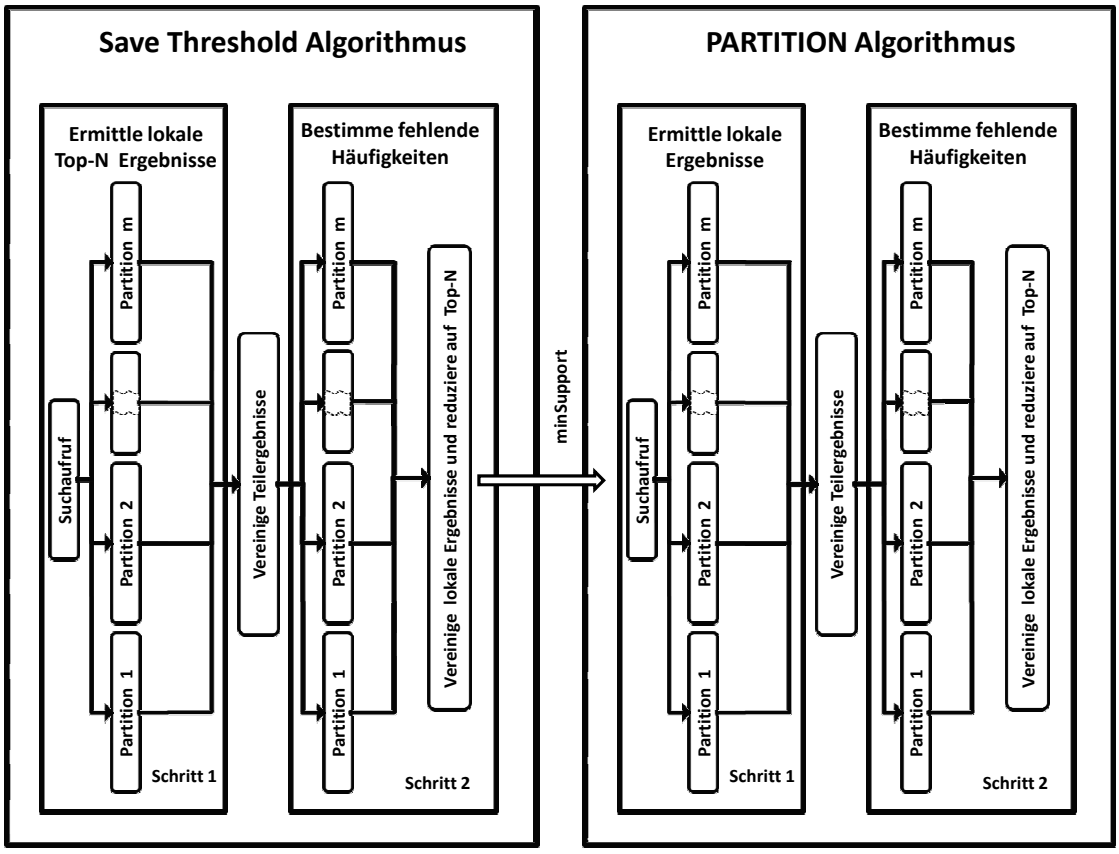


Abbildung 3.5.: Verteilte Suche nach Top-N häufigen Kombinationen mit TPARTITION

dienung wird nicht entsprochen, im Gegenzug wird jedoch das korrekte, der Anfrage entsprechende Ergebnis ermittelt.

Die Mindesthäufigkeit u für PARTITION ist aus den durch STH approximiert bestimmten Ergebnissen ableitbar. Aus den Eigenschaften von STH ist bekannt, dass einige Kombinationen nicht gefunden werden können. Betrachtet man die Häufigkeit des N-ten Ergebnisses X von STH, wird klar, dass alle Top-N-Kombinationen des vollständigen Ergebnisses häufiger oder gleich häufig sein müssen als X. Eine Suche mittels PARTITION und $u = support(X) = minSupport$ liefert somit garantiert mindestens alle korrekten Top-N-Ergebnisse. Eine nachfolgende Einschränkung auf die häufigsten N Kombinationen bildet das Endergebnis.

Besteht die Gesamtlaufzeit von PARTITION und STH maßgeblich aus dem initialen Schritt der lokalen EHK, entsprechen die Kosten für PARTITION - bei Annahme von ca. 50% Mehraufwand von TOP-N-ECLAT gegenüber ECLAT - zusätzlich etwa 50% der Kosten von STH sowie einer doppelte Vorverarbeitung der Daten. Der schnellere Ablauf

von PARTITION gegenüber STH ist eine Folge daraus, dass statt einer Top-N-Suche direkt die passende Mindesthäufigkeit gewählt wird. Liegen die einzelnen Einträge der untersuchten Daten ungünstig verteilt vor, können die Kosten von PARTITION auch deutlich steigen. Im Extremfall wird die Bestimmung der Häufigkeiten aller möglichen Kombinationen nötig. Die Laufzeiten für ein vollständiges TPARTITION sind jedoch in jedem Fall deutlich höher als bei dem alleinigen STH.

Ist bereits im Vorfeld klar, dass die Ergebnisse von STH nicht von ausreichender Qualität sind, kann nach der Vereinigung der lokalen Top-N-Ergebnisse eine Mindesthäufigkeit für das nachfolgende PARTITION ermittelt werden. Auf die Nachsuche und die daraus folgende Bestimmung der korrekten Häufigkeiten kann verzichtet werden. Die Vereinigung aller lokalen Top-N zu einer Liste globaler Top-N-Kandidaten enthält nicht die korrekten Häufigkeiten der Kandidaten. Die Angaben sind aber garantiert kleiner als die korrekten Werte. Äquivalent zu einer vollständigen STH zur Bestimmung der Parameter von PARTITION kann die Häufigkeit des N-ten Top-N-Kandidaten als Mindesthäufigkeit u genutzt werden.

Da die Laufzeit von STH maßgeblich von den lokalen EHK dominiert wird, sind die möglichen Einsparungen durch einen Abbruch nach der ersten Konsolidierung gering. Die Häufigkeit der N-ten Kombination als Mindesthäufigkeit für PARTITION kann aber deutlich geringer sein als nach der Vollendung von STH. Aufgrund der exponentiellen Laufzeitsteigerung mit sinkender Mindesthäufigkeit bei einer EHK werden die Gewinne durch eine verkürzte STH meist durch eine Steigerung der Laufzeit von PARTITION kompensiert. Die Bestimmung von u durch Verwendung des kompletten STH ist somit meist die bessere Alternative.

Zusammenfassend bleibt die Erkenntnis, dass STH lediglich ein approximiertes Ergebnis liefert, dies aber robust und effizient bestimmt. Eine Anwendung sollte entsprechend auf die Nutzung von TPARTITION verzichten, wenn die Ergebnisqualität von STH die gewünschten Informationen bereits ausreichend widerspiegelt. Der folgende Abschnitt soll die bisher gezeigten Verfahren auf ihre Tauglichkeit für praxisnahe Szenarien untersuchen und um fehlende Eigenschaften erweitern.

3.3. Verteilte Top-N-Kombinationen in der Praxis

Für die meisten Szenarien einer verteilten EHK werden die Partitionen der zu untersuchenden Daten als zueinander ähnlich angenommen. Dies entspricht nicht der Praxis, da hier die Daten oft ungleichmäßig über mehrere Rechenknoten verteilt vorliegen. Dies kann beispielsweise durch eine Partitionierung nach einer bestimmten geschäftsbezogenen Semantik begründet sein, wie eine Unterteilung nach Land, Monat oder Filiale. Die einzelnen Partitionen können sich dabei im Umfang signifikant unterscheiden. Als weitere

Möglichkeit können die zu untersuchenden Daten auch durch eine Selektion aus gleichmäßig verteilten Gesamtdaten entstehen. Das kann unter Beachtung der Selektion ebenfalls sehr unterschiedliche Partitionsgrößen hervorrufen.

Leere Partitionen sind hierbei unkritisch, da sie keine Ergebnisse erzeugen können und somit nicht betrachtet werden müssen. Aber kleine, nichtleere Partitionen sind möglich und widersprechen der zur verteilten EHK meist geforderten Annahme einer gleichmäßigen Partitionierung der Daten. Ähnliche Partitionsgrößen sind nach Zaki [Zak99] eine wichtige Voraussetzung für viele Verfahren zur EHK und eine Verletzung dieser Bedingung kann das Laufzeitverhalten dieser Algorithmen signifikant verschlechtern.

Befragungen von Kunden der SAP AG ergaben, dass in den meisten Szenarien mindestens einer der beiden Problemfälle zutrifft. Oft sind sowohl unterschiedliche Partitionsgrößen als auch ungleichmäßige Selektionen möglich. Die in dieser Arbeit angestrebten Ziele der einfachen Anwendbarkeit der EHK und einer stabilen, robusten Laufzeit von Assoziationsregelsuchverfahren werden somit schwer umsetzbar. Entweder wird eine Vorverarbeitung der Daten benötigt, um die Datenpartitionierung zu kontrollieren oder aufzulösen oder die Laufzeit der EHK kann in manchen Fällen extrem ansteigen.

Eine Lösung dieser Problematik ist die Vereinigung der vollständigen verteilten Daten oder der kleineren Partitionen. Die einzelnen Partitionen der resultierenden Datenbestände können damit groß genug werden, um eine stabile Verarbeitung der EHK zu ermöglichen. Eine Konsolidierung zu einem unpartitionierten Datenbestand würde eine verteilte Datenverarbeitung bei Bedarf sogar vollständig vermeiden.

Dies bedeutet jedoch einen Transport, Modifikation oder eine Replikation der betrachteten Daten mit all den damit verbundenen Konsequenzen und Kosten. Eine vom Nutzer willentlich angelegte und potenziell notwendige Semantik der Partitionierung kann zerstört werden. Eine weitere Möglichkeit besteht in der Verwendung einer Stichprobe über die vollständigen gewählten Daten, welche unpartitioniert abgelegt wird. Die nachfolgende Analyse betrachtet jeweils diese unpartitionierte Stichprobe. Dies mindert das Problem einer kostspieligen Replikation und vermeidet eine Modifikation der Originaldaten. Mit Hilfe von Stichproben kann jedoch lediglich ein Ergebnis mit geschätzten Häufigkeiten erreicht werden. Außerdem müssen die Ausgangsdaten möglicherweise entgegen bestehender Datenschutzrichtlinien oder ähnlicher Beschränkungen teilweise repliziert werden. Hinzu kommen gegebenenfalls Kosten zur Pflege oder Neuaufbau der Stichprobe, um diese an Änderungen der Ausgangsdaten anzupassen. Ähnliches gilt für eine Vereinigung und die damit nötige Replikation kleiner Partitionen, um deren Verarbeitung zu ermöglichen. Dabei muss der Aufwand pro untersuchtem Datenbestand erfolgen, inklusive der jeweils verwendeten Selektion.

Eine anwenderfreundliche Lösung soll daher zum Ziel haben, die vorhandenen semantischen, syntaktischen und systembedingten Gegebenheiten möglichst direkt verarbeiten zu können. Ein Verfahren zur Behandlung unterschiedlicher Partitionsgrößen mit stabilen

Laufzeiten wird im Folgenden erläutert. Begonnen wird am Beispiel des PARTITION-Algorithmus zur Verdeutlichung der durch unterschiedliche Partitionsgrößen entstehenden Probleme. Im Anschluss wird eine Lösung sowohl für die auf einer Mindesthäufigkeit basierenden Analysen als auch für die Top-N-EHK vorgeschlagen.

Der PARTITION-Algorithmus nutzt zur Bestimmung der verteilten EHK für jede betrachtete Partition eine lokale EHK mit der gleichen relativen Mindesthäufigkeit, welche auch für den gesamten Datenbestand gewählt wurde. Beispielsweise würde eine EHK nach allen Kombinationen X mit einer globalen Häufigkeit von $support(X) \geq 20\%$ auf allen Partitionen des Datenbestandes nach Kombinationen mit den lokalen Häufigkeiten von mindestens 20% suchen. Weiterführende Informationen wurden hierfür bereits im Abschnitt 2.2.5 vorgestellt oder können in [SON95] nachgelesen werden.

Ausgehend von einem exponentiell wachsenden Suchraum der EHK-Algorithmen bei schwindender Mindesthäufigkeit wächst die Laufzeit deutlich für geringe absolute Mindesthäufigkeiten [Zak99]. Zur Implementierung der Verfahren werden meist absolute statt relative Mindesthäufigkeiten verwendet. Damit ergibt sich z.B. für $minSupport = 10\%$ und $|DB| = 50$ eine absolute Mindesthäufigkeit $minSupportAbs = 5$. Ein Problem entsteht, wenn die absolute Mindesthäufigkeit klein wird. Für $minSupport = 1\%$ ergibt sich in dem Beispiel $minSupportAbs = 1$ und der Algorithmus sucht, speichert und übergibt dem Nutzer alle in dem Datenbestand vorhandenen Kombinationen. Das Erzeugen dieses sehr großen Ergebnisses ist selbst für kleine Datenmengen aufwändig. Die Laufzeit wächst exponentiell mit sinkendem $minSupportAbs$, fällt aber nur linear mit verringertem Datenumfang [Zak99]. Zum Vergleich bewirken dieselben Parameter für eine Partition mit $|DB| = 10.000$ ein $minSupportAbs = 100$. Diesen Wert würden nur wenige Ergebnisse erreichen. Eine EHK mit einer Mindesthäufigkeit von $minSupport = 1\%$ wäre entsprechend effizient durchzuführen.

Daraus folgt die für einen Nutzer im Regelfall nicht erwartete Eigenschaft einer verteilten EHK, dass kleine Partitionen bei gleicher relativer Häufigkeit um Größenordnungen längere Laufzeiten aufweisen können als deutlich größere Partitionen. Bei sehr ungleichen Datenaufteilungen können entsprechend kleine Partitionen der bremsende Faktor der gesamten EHK werden, welcher die Datenanalyse erschweren oder sogar praktisch verhindern kann. In Experimenten konnte gezeigt werden, dass an einem realistischen, nach Datum partitionierten Datenbestand mit sehr ungleichen Datenmengen pro Partition für kleine Teile Laufzeiten von mehreren Stunden benötigt wurden, während um Größenordnungen umfangreicheren Partitionen ihrerseits nur einige Sekunden zur Berechnung der EHK brauchten.

Um eine gute Nutzerfreundlichkeit zu erreichen, ist eine Lösung dieses Problems nötig, welche außerdem für den Nutzer transparent und automatisch erfolgen soll. Eine Möglichkeit besteht in der Einführung einer Mindestgröße für eine zu untersuchende Partition. Alle Partitionen, welche weniger als die geforderten Transaktionen aufweisen, verhalten sich wie eine leere Partition und liefern eine leere Ergebnismenge zurück. Mit der

Annahme, dass damit nicht lokal ermittelte Kombinationen auf den großen Partitionen gefunden werden, ist eine solche Vereinfachung möglich. Lediglich ausschließlich auf kleinen Partitionen auftretende Kombinationen werden nicht ermittelt. Die Erstellung der endgültigen Ergebnisse erfolgt, wie im Abschnitt 2.2.5 und 3.2.4 vorgestellt, unverändert.

Bei dem Nachsuchen fehlender Häufigkeiten von Ergebniskandidaten können die im ersten Schritt ignorierten Partitionen wieder eingebunden werden. Bei einer direkten Bestimmung von Häufigkeiten besteht die Gefahr des großen Ansteigens der Laufzeiten nicht, da die Menge der zu bestimmenden Kombinationshäufigkeiten vordefiniert ist. Das Endergebnis einer solchen Analyse besteht wie bei STH aus Kombinationen mit jeweils korrekten Häufigkeiten. Lediglich Kombinationen, welche ausschließlich auf den ignorierten Partitionen häufig sind und ignoriert wurden, werden nicht gefunden. Nach Untersuchungen an praxisnahen Szenarien mit Verkaufsdaten ist ersichtlich, dass die finale Ergebnismenge von den großen Partitionen dominiert wird. Sicher ist dies jedoch nicht. Die im Rahmen der Evaluation durchgeführten Experimente weisen größtenteils korrekte Ergebnisse auf. Konkrete Garantien sind jedoch nur für einen kleinen Teil des Ergebnisses möglich.

Als alternativer Ansatz im Umgang mit kleinen Partitionen kann ein minimaler Schwellwert für die absolute Mindesthäufigkeit einer EHK definiert werden, welcher auf einer Partition erreicht werden muss. Der Großteil der auf kleinen Partitionen erzeugten Ergebnisse nahe der definierten Mindesthäufigkeit trägt nicht signifikant zum Endergebnis bei, da die großen lokalen Ergebnismengen aus den jeweils niedrigen absoluten Mindesthäufigkeiten resultieren. Große Partitionen weisen dieses Problem nicht auf. Können deren lokale absolute Häufigkeiten das globale Endergebnis dominieren, werden die meisten der auf kleinen Partitionen erzeugten Resultate das globale Ergebnis nur geringfügig beeinflussen.

Daraus folgt, dass eine Begrenzung der lokalen absoluten Mindesthäufigkeiten für die meisten Anwendungen einen sinnvollen Kompromiss zwischen Laufzeit und Ergebnisqualität darstellt. Dafür wird ein Parameter *mams* (Minimum Absolute MinSupport) vorgeschlagen, welcher eine untere Grenze für die jeweils lokalen absoluten Mindesthäufigkeiten definiert. Dieser Wert muss auf jeder Partition von der lokalen absoluten Mindesthäufigkeit erreicht werden. Die absolute Mindesthäufigkeit ergibt sich für t lokal betrachtete Transaktionen durch

$$minSupportAbs = \max(\lceil minSupport \cdot t \rceil, mams)$$

Dieses Verfahren wurde im Rahmen dieser Arbeit als Erweiterung von PARTITION im *Minimum-Absolute-MinSupport-Threshold*-Algorithmus (*MAST*) implementiert. Der optimale Wert für *mams* ist abhängig von verschiedenen Datencharakteristiken, wie beispielsweise der globalen Transaktionsanzahl. Ebenso von der Anzahl der Partitionen und vorhandenen Abhängigkeiten zwischen den zu untersuchenden Daten. Datenabhängigkeiten sind dem Nutzer jedoch meist nicht bekannt und das Ziel der Assoziationsregelsuche. Dieser Einfluss kann somit meist nicht für eine Anpassung von *mams* verwendet werden.

Der Umfang der gewählten Transaktionen ist oft ebenfalls unbekannt, insbesondere wenn Selektionen auf den Daten verwendet werden. Ist $|DB|$ verfügbar, muss der Parameter $mams$ derart gewählt werden, dass keine sinnvollen Ergebnisse auf einer Partition seltener sind als $mams$. Somit können unerwartet große Laufzeiten vermieden werden, ohne die Ergebnisqualität deutlich zu verschlechtern. Als Abwandlung des TOP-N-ECLAT müssen Suchen beim MAST-Algorithmus statt der regulär verwendeten Mindesthäufigkeit $minSupportAbs = 1$ mit einer Mindesthäufigkeit $minSupportAbs = mams$ beginnen (vgl. Algorithmus 3.1, Zeile 1).

Experimente und Analysen realistischer Szenarien ergaben, dass für Verkaufsdaten bereits eine Wahl von $mams = 2$ die Gefahr für eine unerwartet lange Algorithmenlaufzeit und eine damit einhergehende lokale Ergebnismengenexplosion auf kleinen Partitionen signifikant reduziert wird. Die Qualität des entstehenden Ergebnisses ist aber noch nicht abschätzbar. Diese Anforderung kann durch eine Vereinigung von MAST und dem STH-Verfahren realisiert werden.

Unter Nutzung der Argumente aus Abschnitt 3.2.2 ergibt sich als obere Grenze für die Häufigkeit der häufigsten nicht ermittelten Kombination W auf den Partitionen $P_{1...p}$ der Wert

$$supportAbs(W) \leq \sum_{i=1}^{p} (supportAbs(Q_i) - 1) = si$$

wobei Q_i der seltensten lokal gefundenen Kombination der Partition P_i entspricht. Für den Fall, dass für einige Partitionen keine lokalen Ergebnisse vorhanden sind, gilt das jeweilige $minSupportAbs - 1$ als lokale Grenze. Wie bei STH kann sowohl für reguläre, mindesthäufigkeitsbasierte Verfahren als auch Top-N-Algorithmen eine obere Grenze si für die häufigste nicht erfasste Kombination angegeben werden. Damit kann für alle Kombinationen $X \in I$ mit $support(X) > si$ äquivalent zu STH die Korrektheit garantiert werden. Außerdem wird die Ermittlung alle Kombinationen mit einer Häufigkeit größer als si sichergestellt.

Für Datenbestände mit einer Vielzahl kleiner Partitionen kann si stark anwachsen und bewirkt eine große Anzahl unsicherer Top-N-Ergebnisse. Das liegt darin begründet, dass jede lokale Erhöhung der absoluten Mindesthäufigkeit auf einer kleinen Partition zu si addiert wird. Entsprechend wächst si mit der Anzahl der Partitionen, welche mit ihrer regulären Mindesthäufigkeit den Schwellwert $mams$ nicht erreichen. Praktische Evaluationen an realistischen Daten zeigen, dass die finale Ergebnismenge trotz fehlender Garantien zu einem großen Teil korrekt ist. Qualitätszusagen für nicht garantierte Ergebnisse sind in diesem Fall wie bei STH auch hier nicht möglich. Lediglich deren korrekt bestimmte Häufigkeit kann zugesichert werden.

Nachdem die bisherigen Abschnitte gezeigt haben, wie eine EHK auch auf ungünstig verteilten Daten durchgeführt werden kann, betrachtet der folgende Abschnitt Möglichkeiten, um die Menge der extrahierten Kombinationen einzuschränken. Dabei wird neben

der Mindesthäufigkeit oder der Zugehörigkeit zu einer Top-N-Ergebnismenge auch der potenzielle Wert einer Kombination beachtet.

3.4. Heuristische Reduktion der Ergebnismenge

Gemäß der Zielsetzung dieser Arbeit, die Suche nach Assoziationsregeln und die dafür benötigten häufigen Kombinationen für einen Nutzer so einfach wie möglich zu gestalten, ist ein kompaktes Ergebnis sinnvoll. Neben den in bisherigen Abschnitten vorgestellten Ansätzen zur einfachen EHK stellen die folgenden Verfahren weitere Möglichkeiten dar, dem Anwender die Extraktion von interessanten Regeln zu erleichtern. Zum Erreichen dieses Zieles kann bereits während der EHK die nachfolgend aus den Ergebnissen erzeugte Regelmenge um wenig hilfreiche Einträge bereinigt werden. Wenn diese mit sehr hoher Wahrscheinlichkeit keine wichtigen Regeln erzeugen können, kann deren frühzeitige Entfernung sowohl eine effiziente EHK und eine einfachere Auswertung der Assoziationsregeln ermöglichen. Auch in diesen Algorithmen wird der einfachen Anwendung und stabilen Laufzeit der Algorithmen auf Verkaufsdaten untersucht. Die Anpassungen sollen für den Nutzer weitestgehend transparent erscheinen und auch im schlimmsten Fall entstehende Zusatzkosten minimieren.

3.4.1. Approximative geschlossene Kombinationen mit ACHARM

Wie im Abschnitt 2.2.7 bereits einführend erläutert wurde, kann eine Möglichkeit zur Reduktion extrahierter Daten darin bestehen, das Ergebnis um redundante, reproduzierbare Ergebnisse zu bereinigen. Aus Abschnitt 2.2.7 ist bekannt, dass durch eine Suche nach häufigen geschlossenen Kombinationen eine deutliche Verringerung der Ergebnismenge erzielt werden kann. Indem die Ergebnismenge nur Kombinationen enthält, welche nicht Teil einer gleichhäufigen Kombination sind, kann in einigen Szenarien der Umfang des Ergebnisses deutlich reduziert werden. Dies kann dessen Auswertung erleichtern.

Bei der Suche nach geschlossenen Kombinationen wird stets ein gleichwertiges Ergebnis gegenüber einer vollständigen Suche erreicht. Das komplette Ergebnis lässt sich aus den geschlossenen Kombinationen eindeutig rekonstruieren. Dies ist jedoch oft nicht erforderlich. Die alleinige Auswertung der geschlossenen Kombinationen kann ausreichend sein, da geschlossene Kombinationen die Information und den Wert der durch sie repräsentierten Kombinationen weitestgehend widerspiegeln. Sie enthalten jedoch zusätzlich weitere Einzelelemente und damit für viele Interessantheitsbetrachtungen eine Aufwertung gegenüber den nicht geschlossenen Teilkombinationen. Die Beschränkung der Ergebnisauswertung auf geschlossene Kombinationen ermöglicht somit einen Fokus auf die wahrscheinlich für den Anwender relevanten Kombinationen.

Der Algorithmus CHARM, eine Abwandlung von ECLAT, ermittelt die Menge der geschlossenen häufigen Kombinationen [ZH99]. Unter der Annahme, dass die Definition der Menge geschlossener häufiger Kombinationen nicht immer vollständig erfüllt sein muss, um unnötige Kombinationen zu vermeiden, wird in diesem Abschnitt eine Abwandlung von CHARM vorgeschlagen. Diese soll lediglich einen Großteil aller geschlossenen Kombinationen erkennen und einige nicht geschlossene Kombinationen X' mit $support(X') = support(X), X' \subset X$ im Ergebnis erlauben. Dabei gilt die Annahme, dass ein geringfügig erhöhter Umfang des Ergebnisses zu Gunsten einer beschleunigten Auswertung akzeptabel ist. Wenn erforderlich, können außerdem nach der Ergebnisberechnung nicht geschlossene Kombinationen nachträglich entfernt werden.

Untersucht wird hierbei die Auswirkung von nicht vollständig durchgeführten Tests auf Geschlossenheit für alle Ergebniskandidaten. Wie untere anderem in [ZH99; PHM00] gezeigt wird, erweist sich der Test auf Geschlossenheit für die Ergebniskandidaten als kostspielig. Für dünn besetzte Datenbestände können die Tests auf Geschlossenheit sogar den Großteil der Ressourcen einer EHK beanspruchen [HGN00; TCRB06]. Während in [ZH99; PHM00] diese Kosten durch geschickte Zugriffsstrukturen minimiert werden, wird in dieser Arbeit vorgeschlagen, diese Tests vereinfacht durchzuführen und einen Großteil der entstehenden Kosten zu vermeiden. Optimiert wird hierbei der Algorithmus ECLAT bzw. dessen Variante CHARM. Dieser Algorithmus soll als *Approximierter CHARM (ACHARM)* bezeichnet werden.

Abbildung 3.6 zeigt ACHARM in Anlehnung an das Beispiel aus Abbildung 3.1. Die obere Darstellung repräsentiert den vollständigen Suchraum, wobei leicht identifizierbare, zueinander geschlossene Kombinationen grau unterlegt sind. Erkannt werden kann dies durch die jeweils gleichbleibende Häufigkeit in Klammern trotz einem Abstieg im Kombinationsbaum. Wird in einem Arbeitsschritt beispielsweise die Häufigkeit der Kombination B bestimmt, werden deren entsprechende Transaktionen auf weitere häufigere Kombinationen BC und BD mit der Teilkombination B untersucht.

Sind diese Häufigkeiten bestimmt, kann ein Vergleich mit der Häufigkeit der Ausgangskombination B durchgeführt werden. Wurde die Menge der für B relevanten Transaktionen durch den Schritt auf BC eingeschränkt, kann B durch BC repräsentiert werden. Indem C zu dem Knoten von B hinzugefügt wird und einen neuen Knoten BC bildet, kann die Betrachtung der Kindknoten des ursprünglichen BC vermieden werden. Deren Kombinationen werden nun ebenfalls unter dem neuen Knoten BC ermittelt. Der durch Anwendung von ACHARM entstehende Kombinationsbaum wird in der unteren Abbildung gezeigt. Der Suchraum wird auf diese Weise mit ACHARM reduziert.

Bisherige Implementierungen für eine vollständige Bestimmung geschlossener Kombination überprüfen für jeden Ergebniskandidaten, ob dieser bestehende Kandidaten ersetzen kann oder bereits durch diese repräsentiert wird. Die dafür nötigen Analysen werden bei ACHARM durch eine einfache Prüfung auf konstante Häufigkeit trotz Abstieg im Kombinationsbaum ersetzt, was deutlich geringeren Aufwand bedeutet.

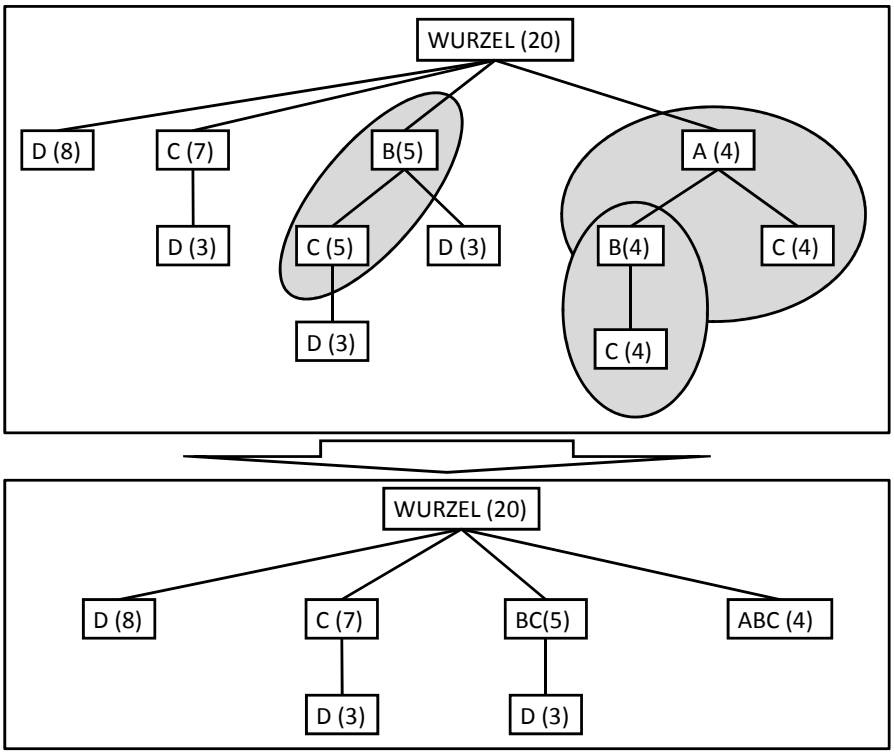


Abbildung 3.6.: Approximativ ermittelte geschlossenen Kombinationen mit ACHARM

Diese Strategie kann jedoch einen Teil der nicht geschlossenen Kombinationen als solche nicht erkennen. In Abbildung 3.6 kann dies am Beispiel der Kombination *CD:3* und der Kombination *BCD:3* gezeigt werden. Obwohl *CD* durch *BCD* repräsentiert werden kann und die gleichen Häufigkeiten gelten, wird *BC* als geschlossen erkannt. Die wesentlichen Vorteile von geschlossenen Kombinationen gegenüber der vollständigen Ergebnismenge bleiben jedoch erhalten. Die Wiederherstellung des kompletten Ergebnisses ist beispielsweise äquivalent den geschlossenen Kombinationen weiterhin vollständig möglich und der Ergebnisumfang wird gegenüber nicht geschlossenen Kombinationen reduziert.

Geschlossene Kombinationen auf verteilten Datenbeständen

Eine negative Eigenschaft der Suche nach geschlossenen Kombinationen ist deren nur bedingte Umsetzbarkeit für verteilte Berechnungen. Hierfür besteht bisher wenige Forschungsarbeit. Von Lucchese [LOP07; LOPS05] wird eine Lösung vorgestellt, welche sich auf eine parallele Verarbeitung der Daten bei gemeinsam verwendbarem Speicher beschränkt. Damit kann beispielsweise eine verteilte Suche auf einem Rechner mit mehreren Recheneinheiten erfolgen. Die Analyse über Systemgrenzen hinweg ist aber nicht möglich. Deren Algorithmus *MT-CLOSED* arbeitet mit einer Aufteilung des Suchraumes.

Dies erlaubt eine parallele Verarbeitung der Daten auf mehreren Rechenknoten, wobei jedem Knoten die Daten zur Berechnung seiner Kombinationen vollständig zur Verfügung stehen müssen. Jeder der Teilsuchräume benötigt Zugriff auf alle von ihm benötigten Daten. Im Normalfall sind die Transaktionen eines Datenbestandes Grundlage mehrerer Kombinationen und Teilsuchräume. Daraus resultiert für MT-CLOSED meist eine nicht disjunkte Partitionierung der Gesamtdaten. Ein Großteil der einzelnen Datensätze ist in mehrere Partitionen vorhanden. Die im Regelfall nicht mögliche disjunkte horizontale Partitionierung der Daten verhindert eine Verwendung in Kombination mit den Lösungen dieser Arbeit, da sich der Algorithmus nur bedingt an die zu untersuchenden Daten anpassen kann.

Ein Verfahren um diese Lücke zu schließen und eine parallelisiert Suche geschlossener Kombinationen auf beliebig partitionierten Daten zu ermöglichen, wird in [LZCZ07] unter dem Namen *DFCIM* vorgestellt. Die einzelnen Rechnerknoten arbeiten hierfür unabhängig voneinander und teilen mit Ausnahme des Netzwerkes keine weiteren Komponenten. Die Funktionsweise ist hierbei analog der Verwendung von PARTITION zur Durchführung der EHK. Der Ablauf ist wie folgt:

1. Lokale Suche geschlossener Kombinationen pro Partition.

2. Vereinigung der jeweiligen Teilergebnisse.

3. Nachsuche fehlender Teilhäufigkeiten.

4. Bildung des finalen Ergebnisses.

Das Problem ist hierbei, dass nach der Vereinigung der lokalen geschlossenen Kombinationen nur selten eine Aussage über den Status der globalen Kombinationen zu treffen ist. Lediglich auf allen Partitionen als geschlossen erkannte Kombinationen sind bereits als global geschlossen erkennbar. Alle anderen Kandidaten müssen entweder durch Extraktion aus den geschlossenen Kombinationen oder durch Nachsuche im Datenbestand evaluiert werden. Zusätzlich muss das Endergebnis bei erfolgter Nachsuche um nicht geschlossene Kombinationen bereinigt werden, da erst hier der globale Status einer Kombination feststellbar ist. Damit kann als endgültiges Ergebnis die Menge der global geschlossenen Kombinationen ermittelt werden. Im Schritt der Vereinigung muss das Zwischenergebnis jedoch stark erweitert werden, was die Reduktion der geschlossenen Kombinationen weitestgehend aufhebt.

Eine Suche nach geschlossenen Kombinationen ist somit nur bedingt auf verteilte Datenbestände übertragbar. Durch die aufwändige Konsolidierung der Teilergebnisse und die dabei durchgeführte Überprüfung auf globale Geschlossenheit für gefundene Kombinationen verringert sich ein zu erwartender Gewinn durch eine verteilte Berechnung deutlich. Der Vorteil eines kompakteren Resultats bei geschlossenen Kombinationen bleibt vorhanden, kann aber in vielen Fällen ebenso nach der Ermittlung des vollständigen,

nicht geschlossenen Ergebnisses vergleichbar effizient erreicht werden. Die Verwendung dieses Verfahrens ist daher nur sinnvoll, wenn selbst bei geringer Reduktion der auf einer Partition gefundenen Ergebnismenge deutliche Laufzeitgewinne erreicht werden, was bei den in dieser Arbeit betrachteten Verkaufsdaten nicht zu erwarten ist. Im Falle von nicht verteilten Datenbeständen ist die Bestimmung von geschlossenen Kombinationen jedoch weiterhin sinnvoll. Der nachfolgende Abschnitt zeigt, wie deren Wirkungsweise noch verstärkt werden kann.

3.4.2. Bestimmung unscharf-geschlossener Kombinationen

Als geschlossen gilt eine Kombination X', wenn keine Kombination X existiert, deren Häufigkeit mit der von X' übereinstimmt und bei der $X' \subset X$ ist. Nur damit ist eine vollständige und korrekte Rekonstruktion aller Kombinationen aus der Teilmenge der geschlossenen Kombinationen möglich. Wird auf die Eigenschaft einer vollständigen Wiederherstellung der Ergebnismenge verzichtet, kann die Definition geschlossener Kombinationen gelockert werden. Objektive Interessantheitsmaße für Assoziationsregeln und häufige Kombinationen sind meist stetig, wodurch geringe Änderungen der zugrunde liegenden Häufigkeiten nur wenig das sich daraus ergebende Interessantheitsmaß verändern. Leichte Abweichungen bei der Rekonstruktion der Häufigkeiten einzelner Kombinationen können somit akzeptabel sein, da der Wert der Regel nur gering verändert wird.

Daraus folgt, dass die gewonnenen Informationen aus zwei Kombinationen X' und X mit $X' \subset X$ und $support(X') \approx support(X)$ nur geringfügig die zu ermittelnde Information aus X übersteigt. Der Mehrwert einer separaten Betrachtung von X' ist gering. Die Kombination X' kann daher für einige Betrachtungen von Assoziationsregeln trotz leicht unterschiedlicher Häufigkeit als durch X repräsentiert betrachtet werden. Ein ähnlicher Ansatz wurde bereits unter der Bezeichnung *Condensed* oder *δ-tolerant Itemsets* untersucht [JYP03; XHYC05; CKN06; PDZH04]. Im Rahmen dieser Arbeit soll dieses Konzept mit dem Algorithmus ACHARM kombiniert werden und wird als *Fuzzy-Closed Frequent Itemset Mining (FCFIM)* bezeichnet. Anders als in den bekannten Verfahren soll der erlaubte Fehler jedoch unabhängig von DB gelten und darf in keinem Fall überschritten werden. Die bisherigen Implementierungen unterscheiden sich in der Art der Definition der erlaubten Unschärfe:

- [PDZH04] toleriert bei dem Algorithmus CFP eine Abweichung zwischen den absoluten Häufigkeiten zweier Kombinationen.

- [JYP03; PDZH04] erlauben bei dem Algorithmus TCFI pro unterschiedlichem Element zweier Kombinationen eine relative Abweichung δ zwischen den Häufigkeiten.

Da CFP Abweichungen der absoluten Häufigkeiten toleriert, können sich die für eine Kombination bestimmten Häufigkeiten relativ betrachtet sehr deutlich vom korrekten

Wert unterscheiden. Während beispielsweise eine Abweichung von 100 bei einer häufigen Kombination unbedeutend sein kann, ist sie bei einer selteneren Kombination möglicherweise sehr groß. Dies verfälscht sehr stark den Wert der seltenen Kombinationen.

Bei TCFI wird ein relativer Fehler δ zwischen den Häufigkeiten der Kombinationen X und X' (bei $X' \subset X, |X| = |X'| + 1$) toleriert. Werden jedoch mehrere Kombinationen X' durch X repräsentiert, wächst dieser Fehler an. Laut [CKN06] vergrößert sich der hierbei entstehende Fehler exponentiell mit der Anzahl gemeinsam repräsentierten Kombinationen. Werden beispielsweise die Kombinationen AB und BC beide durch ABC repräsentiert, da beide innerhalb der δ-Toleranz von ABC liegen, wächst der maximale Fehler nach [CKN06] auf δ^2. Dies ist zumindest bei der Übertragung auf ACHARM jedoch nicht der Fall. Der maximal mögliche Fehler summiert sich stattdessen über alle vereinigten Ergebnisse. Gilt beispielsweise die mindestens erforderliche Übereinstimmung von $\delta = 90\%$ der beteiligten Transaktionen, können je 10% verschiedene Transaktionen für AB und BC jeweils disjunkt sein. Die Häufigkeit von ABC wird also potenziell um $2 \cdot 10\%$ Transaktionen überschritten. Laut [CKN06] ist dieser Fehler nur bei $90\%^2 = 81\%$. Während diese Abweichung bei nur wenigen gemeinsam vereinigten Kombinationen unbedeutet ist, kann der maximale Fehler bei einigen Datenbeständen das gewählte δ deutlich übersteigen. Die Qualität der Rekonstruktion des vollständigen Ergebnisses ist daher reduziert.

Um die Güte eines rekonstruierten Ergebnisses beurteilen zu können, bleibt der maximal mögliche Fehler in dieser Arbeit konstant. Außerdem soll die Häufigkeit einer Kombination um einen maximalen relativen Faktor von der korrekten Häufigkeit abweichen, um Änderungen seltenerer Kombinationen nur im entsprechend kleineren Rahmen zu erlauben. Dafür wird ein Parameter β eingeführt, welcher den maximalen relativen Abstand der Häufigkeiten zweier Kombinationen X und X' definiert, bis zu dem diese als zueinander unscharf-geschlossen gelten. Als Ähnlichkeitsfaktor gilt hierbei $supportAbs(X)/supportAbs(X')$. Wird $\beta = 1 = 100\%$ gewählt, ergibt sich der Spezialfall der geschlossenen Kombinationen. Die Toleranz β gilt dabei unabhängig von der Anzahl der Elemente in X und X'. Sie muss entsprechend für alle durch X repräsentierten Kombinationen garantiert werden.

Die Unschärfe β gilt wie auch bei [PDZH04; JYP03] nicht transitiv. Wenn sowohl X_1 und X_2 als auch X_2 und X_3 zueinander unscharf-geschlossen sind, muss X_1 nicht unscharf-geschlossen zu X_3 sein. Somit ist die Menge der unscharf-geschlossenen Kombinationen nicht eindeutig. Mehrere Mengen können die erforderlichen Bedingungen erfüllen. Insbesondere ist die resultierende Menge bei FCFIM abhängig von der Reihenfolge der Vereinigung der unscharf-geschlossenen Kombinationen. Im Falle von FCFIM erfolgt die Vereinigung gemäß der Traversierung des Suchraumes.

FCFIM arbeitet äquivalent zu ACHARM, deklariert eine Kombination aber als unscharf-geschlossen, wenn der entstehende relative Fehler zwischen den absoluten Häufigkeiten zweier Kombinationen X und X' inklusive der bereits im Vorfeld entstandenen Abwei-

chungen geringer als der erlaubte Unschärfefaktor ist. Um für FCFIM einen maximalen Fehler nicht zu überschreiten, wird in jedem Traversierungschritt durch den Kombinationsbaum der bis zu dieser Stelle vorhandene absolute Fehler mitgeführt. Der aus den vorherigen Kombinationen bekannte maximale absolute Fehler wird als ϵ bezeichnet. Ist die folgende Ungleichung erfüllt, gelten X' und X zueinander als unscharf-geschlossen und werden gemeinsam durch X repräsentiert.

$$supportAbs(X) \geq (supportAbs(X') + \epsilon) \cdot \beta$$

Wie schon bei der regulären Bestimmung geschlossener Kombinationen, sind die erreichbaren Ergebnisreduktionen stark von den betrachteten Daten abhängig. Für einige Datenbestände kann die Menge der geschlossenen Kombinationen um Größenordnungen geringer sein als die vollständige Ergebnismenge. Verkaufsdaten weisen hier oft nur geringe Reduktionen auf. Diese kann durch eine Erweiterung auf unscharf-geschlossenen Kombinationen verbessert werden. Wie bereits für ACHARM erläutert, ermittelt auch FCFIM ein approximiertes Ergebnis. Auch hier können nicht unscharf-geschlossene Kombinationen Teil des Ergebnisses werden. Diese nutzen lediglich den möglichen Fehler nicht soweit aus, wie es möglich wäre. Im Gegenzug wird aber wie bei ACHARM für jede Kombination eine Validierung gegenüber den bereits ermittelten Ergebnissen gespart. Die somit beschleunigte Verarbeitung kann die nicht erreichte, maximal mögliche Verringerung der Ergebnismenge rechtfertigen.

Während dem Abstieg im Suchraum werden die gefundenen Kombinationen seltener. Da der absolute Fehler ϵ jedoch jeweils mitgeführt wird, kann er in Relation zu den Häufigkeiten der Kombination eine Überschreitung der Fehlertoleranz β bewirken, auch wenn durch den Abstieg der absolute Fehler nicht erhöht wurde. Wird das erkannt, muss ϵ wieder auf 0 zurückgesetzt werden, indem die relevante Transaktionsmenge auf Vorhandensein der entsprechenden Kombination überprüft wird. Sie ist stets Obermenge der korrekten Transaktionsmenge, wodurch diese mit vertretbarem Aufwand leicht extrahierbar ist. Beispielsweise können zwei Kombinationen $AB : 11$ und $AC : 11$ bei $\beta = 20\%$ durch $ABC : 10$ repräsentiert werden. Dabei entsteht der absolute Fehler $\epsilon = 2$. Wird basierend auf ABC die Kombination $ABCD : 8$ bestimmt, gilt auch hier $\epsilon = 2$. Das entspricht einem möglichen Fehler von $2/8 = 25\%$. Da dies bereits β überschreitet, was bei FCFIM nicht erlaubt ist, müssen die acht Transaktionen von $ABCD$ geprüft werden. Die Häufigkeit von $ABCD$ wird validiert und der Fehler somit $\epsilon = 0$.

Während eine Implementierung für eine unverteilte Suche mittels FCFIM durch eine Abwandlung der Definition einer geschlossenen Kombination erfolgen kann, ist eine Übertragung dieses Konzeptes auf partitionierte Daten nicht möglich. Übergreifend kann aus lokalen Ergebnissen nicht entschieden werden, ob eine Kombination global unscharf-geschlossen ist. Hier ist eine Lösung denkbar, welche verteilt die vollständigen häufigen (geschlossenen) Kombinationen bestimmt und beim Vereinigen und Bilden des globalen vollständigen Ergebnisses dieses auf die unscharf-geschlossenen Kombinationen reduziert. Auf diese Art bleibt der Vorteil einer Ergebnisbereinigung erhalten. Allerdings kann der Berechnungsaufwand nicht reduziert werden, er kann sogar steigen. Die Nutzung der

unscharf-geschlossenen Kombinationen bei einer verteilten EHK ist daher meist nicht sinnvoll. Wenn die entstehende Ergebnismenge jedoch sehr groß ist, kann deren Vereinfachung zur Menge der unscharf-geschlossenen Kombinationen die Weiterverarbeitung zu Assoziationsregeln vereinfachen.

Von Cheng, Yu und Han [CYH06] wird eine weitere Form der approximierten Berechnung vorgeschlagen. Im Gegensatz zu FCFIM werden durch deren Algorithmus *AC-CLOSE* keine Unterschiede in den Häufigkeiten erlaubt, sondern Abweichungen der enthaltenen Elementen toleriert. Die Unschärfe besteht somit nicht in semantischen Eigenschaften wie der Häufigkeit einer Kombination, sondern auf syntaktischer Ebene, der Kombination selbst. Während das Ziel auch bei AC-CLOSE eine Vereinfachung des Ergebnisses und eine Beschleunigung von dessen Berechnung ist, sind dessen Ergebnisse deutlich verschieden zu denen von FCFIM.

Eine Anwendung von AC-CLOSE bei einer Warenkorbanalyse ist nur bedingt möglich. Da hier meist nur wenige Elemente in den ermittelten Kombinationen enthalten sind, muss eine erlaubte Abweichung sehr groß gewählt werden, um eine relevante Optimierung zu erreichen. Für die in dieser Arbeit hauptsächlich adressierte Analyse von Geschäftsdaten ist AC-CLOSE somit ungeeignet. Bei der Suche von häufigen Kombinationen mit vielen Elementen oder auf verunreinigten Datenbeständen mit vielen Störungen kann AC-CLOSE jedoch eine Qualitätsverbesserung des Ergebnisses bewirken [CYH06].

3.4.3. Schätzung von Suchparametern

Eine Alternative zur Verwendung von Top-N-Strategien stellt die Schätzung von Parametern einer EHK dar. Insbesondere wurden in dieser Arbeit die Auswirkungen veränderter Eingabeparameter auf die zu erwartenden Ergebnisse und deren Kardinalität untersucht. Dabei wird mit Hilfe einiger effizient zu extrahierender Informationen über die Charakteristiken eines Datenbestandes eine Parameterschätzung durchgeführt. Mit dieser Parameterbelegung kann die Größenordnung des zu erwartenden Ergebnisses bestimmt werden.

Grundsätzlich ist eine Schätzung von Parametern einer EHK nur bedingt möglich, da die Annahme einer statistischen Unabhängigkeit und gleichmäßiger Verteilung der Daten erfolgen muss. Da jedoch genau das Gegenteil davon, die Suche nach Abhängigkeiten der Daten, durchgeführt wird, widersprechen sich diese Annahme und deren Ziel. Entweder wird die Annahme verletzt oder das Ziel der Suche als nicht vorhanden angenommen.

Wird aber davon ausgegangen, dass die in einem Datenbestand zu suchenden Besonderheiten selten sind, kann in einigen Szenarien von der Unabhängigkeit eines Großteils der Daten ausgegangen werden. Unter der Annahme, dass die zu untersuchenden Daten keinerlei Abhängigkeiten aufweisen, sind die Analyse und die dabei entstehenden häufi-

gen Kombinationen wenig sinnvoll. Aus den Häufigkeiten der Einzelelemente sind jedoch grobe Abschätzungen des zu erwartenden Ergebnisumfanges möglich. Zur Realisierung einer Schätzung von günstigen Parametern zur EHK werden im Folgenden Möglichkeiten zur Kardinalitätsschätzung eines zu erwartenden Ergebnisses betrachtet.

- Laststatistik: aus bekannten, vorherigen Suchläufen und den dabei erzeugten Ergebniseigenschaften kann über einen Vergleich der Konfigurationen eine Schätzung sinnvoller Parameterbelegungen erfolgen.

- Datencharakteristik: aus effizient zu ermittelnden Datencharakteristiken kann die Fähigkeit eines Datenbestandes zur Bildung von Kombinationen geschätzt werden.

Sind genügend Erfahrungswerten vorhanden und die Datencharakteristiken konstant geblieben, ist die Parametrisierung neuer Anfragen trivial. Diese Informationen sind jedoch nur selten ausreichend gegeben, um eine Parameterschätzung zu ermöglichen. Außerdem ist es sehr wahrscheinlich, dass ein Datenbestand mit umfangreichen Laststatistiken dem Nutzer bereits bekannt ist und dieser auch selbstständig sinnvolle Parameter aus seinem Erfahrungsschatz wählen kann. Eine automatische Unterstützung ist nicht nötig. Die Bestimmung von Parametern für eine Suche in einem unbekannten Datenbestand wird aus den Statistikinformationen von vorherigen Suchen auf anderen Daten nur unzureichend ermöglicht, da bereits geringe Veränderungen ungünstige Parameterbelegungen erzeugen können.

Dieses Vorgehen entspricht jedoch im Wesentlichen der derzeit gängigen Herangehensweise bei einer EHK in unbekannten Daten. Dabei wird meist mit einer groben Überschätzung der Mindesthäufigkeit begonnen und diese iterativ auf einen günstigen Wert abgesenkt. Dies erfordert Fachwissen und durch ein mehrfaches Durchführen der Analyse zusätzliche Systemressourcen. Unter Beachtung der für jeden Versuch entstehenden Kosten und deutlichen Unterschiede in den Charakteristiken von Datenbeständen wird das Problem deutlich.[1]

Für die Nutzung von Datencharakteristiken wird in dieser Arbeit eine Möglichkeit zur effizienteren iterativen Bestimmung der jeweils günstigen Parameter vorgeschlagen. Dabei soll eine EHK durch Wahl einer Mindesthäufigkeit durchgeführt werden. Statt die Häufigkeiten von Kombinationen durch die Betrachtung der kompletten Daten korrekt zu bestimmen, werden diese aus den Häufigkeiten ihrer Einzelelemente und der Annahme von Unabhängigkeit geschätzt. Die dabei ermittelten Kombinationen entsprechen den statistisch zu erwartenden Kombinationen und weisen somit keinerlei Besonderheit auf. Sie entsprechen somit weder der Realität, noch sind sie als Grundlage für Assoziationsregeln geeignet. Unter der Annahme, dass positive und negative Abhängigkeiten in den Originaldaten gleichermaßen auftreten, bildet eine Ergebnisschätzung bei Unabhängig-

[1] Beispieldaten siehe Anhang Seite 184

keit einen Anhaltspunkt für die zu erwartende Ergebnismenge mit gleichen Parametern aus den Originaldaten.

Beispielsweise kann aus den Werten $support(Bier) = 10\%$, $support(Windeln) = 20\%$ und $support(Milch) = 30\%$ die Häufigkeit der Kombination $(Bier, Windeln, Milch)$ durch Multiplikation der Einzelhäufigkeiten mit $10\% \cdot 20\% \cdot 30\% = 0,6\%$ geschätzt werden. Entsprechend müssen diese Wahrscheinlichkeiten der Einzelwerte gegeben sein. Die für eine Kombination bestimmte Häufigkeit entspricht dem statistisch erwarteten Wert. Dies widerspricht dem Ziel einer EHK, bei dem meist Abweichungen von statistisch erwarteten Eigenschaften gesucht werden. Das Suchziel wird somit als nicht vorhanden angenommen. Ein derartig ermitteltes Ergebnis kann dennoch sinnvoll sein, wenn die Anzahl der über- und unterschätzten Kombinationen ähnlich ist. In diesem Fall ist zwar das Ergebnis wenig sinnvoll, doch die resultierende Ergebniskardinalität bleibt ähnlich und kann als Anhaltspunkt für den Umfang der korrekten Ergebnismenge dienen.

Dieser Algorithmus soll als *EHKSIM* bezeichnet werden. Sein Ziel ist jedoch anders als bei anderen Verfahren zur EHK nicht die Suche nach häufigen Kombinationen, sondern die Unterstützung des Nutzers bei der Wahl der EHK-Parameter. Alternativ können die Ausgangsdaten und deren Abhängigkeiten auch in Form einer Stichprobe (siehe Abschnitt 2.2.9) dargestellt werden. Dies ist jedoch nur sinnvoll, wenn das Erstellen der Stichprobe wenige Kosten verursacht. Ist das nicht der Fall, beispielsweise aufgrund der aufwändigen Vorverarbeitung der Daten, ist die Verwendung der Stichprobe zur Bestimmung sinnvoller Parameter vergleichsweise kostspielig. Zudem muss die Stichprobe klein genug sein, um eine schnelle EHK zu gewährleisten, was deren Ergebnisqualität negativ beeinflusst. Daher ist die Verwendung von EHKSIM vor allem bei großen Datenbeständen mit aufwändiger Vorverarbeitung eine sinnvolle Alternative.

Als Algorithmen zur Durchführung der simulierten EHK eignen sich z.B. APRIORI und ECLAT. Beide erzeugen aus bereits bekannten Kombinationen neue, komplexere Kandidaten und bestimmen deren Häufigkeit. Die Schwierigkeit bei beiden Verfahren besteht dabei hauptsächlich beim Zugriff auf die Ausgangsdaten, um Kandidaten zu validieren. Je größer die Datenmenge, umso größer ist die Laufzeit des Verfahrens. Wird in diesem Schritt die tatsächliche Häufigkeit eines Kandidaten durch deren Erwartungswert ersetzt, kann der vergleichsweise langsame Datenzugriff durch einfache, schnelle Multiplikationen ersetzt werden. Da als gewünschte Information lediglich der zu erwartende Ergebnisumfang betrachtet wird, müssen gültige Kombinationen nicht gespeichert werden. Deren Zählung ist ausreichend und spart somit weitere Datenzugriffe. Insgesamt benötigt eine simulierte EHK gegenüber einer tatsächlichen EHK signifikant weniger Speicherzugriffe. Unter der Annahme, dass diese Zugriffe den Großteil der Laufzeit einer EHK beanspruchen, kann diese somit signifikant beschleunigt werden. Insbesondere besteht keine Abhängigkeit zwischen der Laufzeit des Verfahrens und dem Umfang der betrachteten Datenmenge.

Die Bestimmung einer sinnvollen Mindesthäufigkeiten erfolgt iterativ und benötigt somit mehrere Durchläufe der EHK, bis günstige Parameter bestimmt sind. Prinzipiell weist diese Methode die Probleme eines Regelkreislaufes auf. Sowohl Über- als auch Untersteuerungen sind möglich und die Anzahl nötiger Iterationen ist im Vorfeld nicht erkennbar. Anders als die Bestimmung passender Parameter durch eine mehrfache reguläre EHK werden die durch Häufigkeitsschätzung bestimmten Kombinationen jedoch signifikant effizienter - durch wenige Multiplikationen ihrer Elementhäufigkeiten - bestimmt. Da zur Berechnung der Häufigkeit einer Kombination nicht auf die Ausgangsdaten zugegriffen wird, ist die Laufzeit für deren Bestimmung unabhängig von der Anzahl der zugrunde liegenden Transaktionen. Vor allem bei umfangreichen Datenbeständen ist eine solche Berechnung deutlich effizienter als eine korrekte EHK. Das hierbei geschätzte, statistisch erwartete Ergebnis ist dabei nicht direkt verwendbar. Es enthält keine Besonderheiten der betrachteten Daten. Diese werden sogar explizit nicht beachtet. Es erlaubt jedoch eine Abschätzung der zu erwartenden Ergebniskardinalität einer EHK, was für eine iterative Annäherung an günstige Parameter ausreichend ist.

Die vorgestellten Verfahren arbeiten heuristisch. Eine fehlerhafte Schätzung kann schon bei geringem Abweichen signifikante Auswirkungen auf das entstehende Ergebnis haben. Daher wird die Verwendung nur bedingt empfohlen und soll nur als Vorschlag betrachtet werden, falls keine anderen Verfahren, wie z.B. Top-N-Algorithmen, möglich sind. Das in dieser Arbeit betrachtete Ziel der Parameterreduktion kann mit Hilfe von EHKSIM nur teilweise erreicht werden. Erfahrungen mit praxisnahen Szenarien bestätigen sinnvolle Parameter, doch deren Güte kann nicht garantiert werden. Die Nutzbarkeit dieses Verfahrens ist deutlich schwieriger, als es bei den in dieser Arbeit bereits erläuterten Top-N-Algorithmen zu erwarten ist.

3.5. Evaluation

Der folgende Abschnitt untersucht die Algorithmen und Vorschläge dieses Kapitels näher und bietet einen Einblick in das Verhalten der vorgestellten Verfahren und deren Verwendung in praktischen Szenarien. Dabei wird jedem Abschnitt dieses Kapitels in gleicher Abfolge ein Abschnitt in der Evaluation gewidmet.

3.5.1. Testumgebung

Im Anhang A werden die in dieser Arbeit zur Evaluation genutzten Datenbestände im Detail beschrieben. An dieser Stelle soll sich somit auf eine kurze Erläuterung beschränkt werden. Betrachtet werden in dieser Auswertung sowohl synthetische als auch reale Datenbestände.

Vertreter synthetischer Daten sind *SynthB* und *SynthC*, welche durch eine Modifikation des IBM Data Generator [Ill] erzeugt wurden. Dessen nötige Konfiguration ist im Anhang ersichtlich. Der *Connect-4*-Datenbestand enthält für das Spiel „Vier gewinnt“ Spielstände und deren Spielausgang. Er spiegelt Geschäftsdaten nur bedingt wider. Er dient jedoch in ähnlichen Arbeiten oft als Beispiel für extrem dichte Datenbestände und soll das Verhalten bei für Geschäftsdaten untypischen Datencharakteristiken zeigen.

Als Vertreter der in Arbeit adressierten Verkaufsdaten wird der öffentlich verfügbare *Retail*-Datenbestand [BSVW99] genutzt. Um das Verhalten für SAP-Verkaufsdaten zu untersuchen, werden die Datenbestände *KundeA* und *KundeB* in verschiedenen Datenverteilungen und Partitionierungen genutzt. Diese sind nicht öffentlich verfügbar, einige Charakteristiken der Daten werden jedoch im Anhang A beschrieben. Partitionierte Datenbestände wurden jeweils nach der Transaktionsnummer aufgeteilt. Alle betrachteten Daten sind im vertikalen Datenformat gespeichert, d.h. jeder Eintrag einer Transaktion wird als ein Tupel (Transaktionsnummer, Element) in einer Zeile der Basisrelation abgelegt.

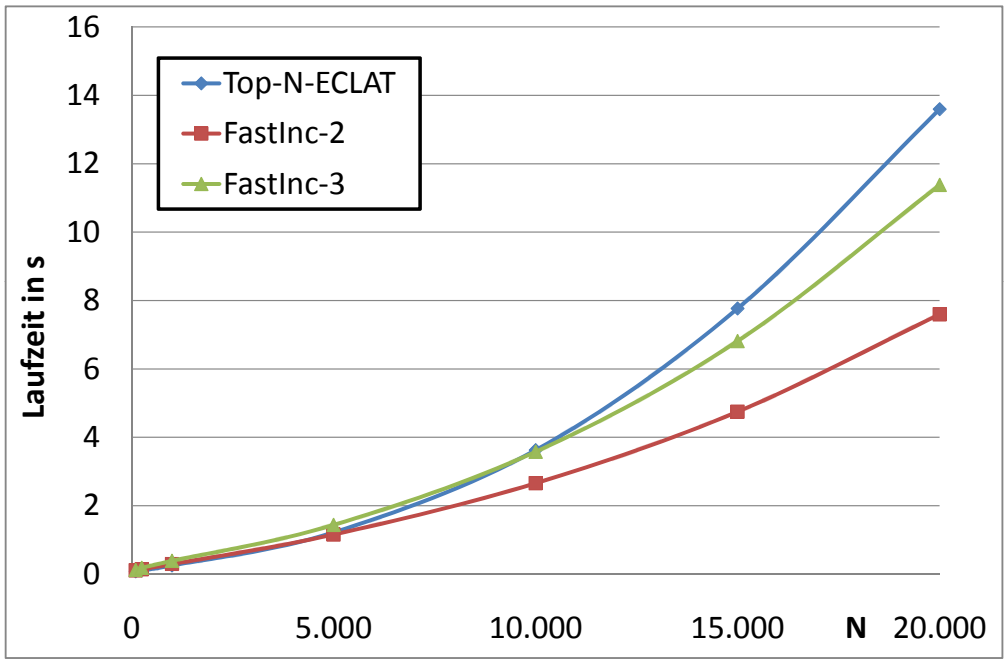
Für die Experimente wurde eine Blade-Serverarchitektur mit bis zu drei Blades verwendet. Pro Blade waren je vier Prozessoren mit 2,6 GHz Rechenleistung und 8 GB Hauptspeicher vorhanden. Als Betriebssystem diente *Microsoft Windows Server 2003 Enterprise x64*. Die Kommunikation zwischen den Blade-Servern erfolgte über Gigabit Ethernet. Sämtliche Algorithmen wurden innerhalb der Systemarchitektur des SAP BW Accelerator integriert. Entsprechend erfolgte die Implementierung in der Programmiersprache C++.

3.5.2. Heuristische Optimierung der Top-N-Suche

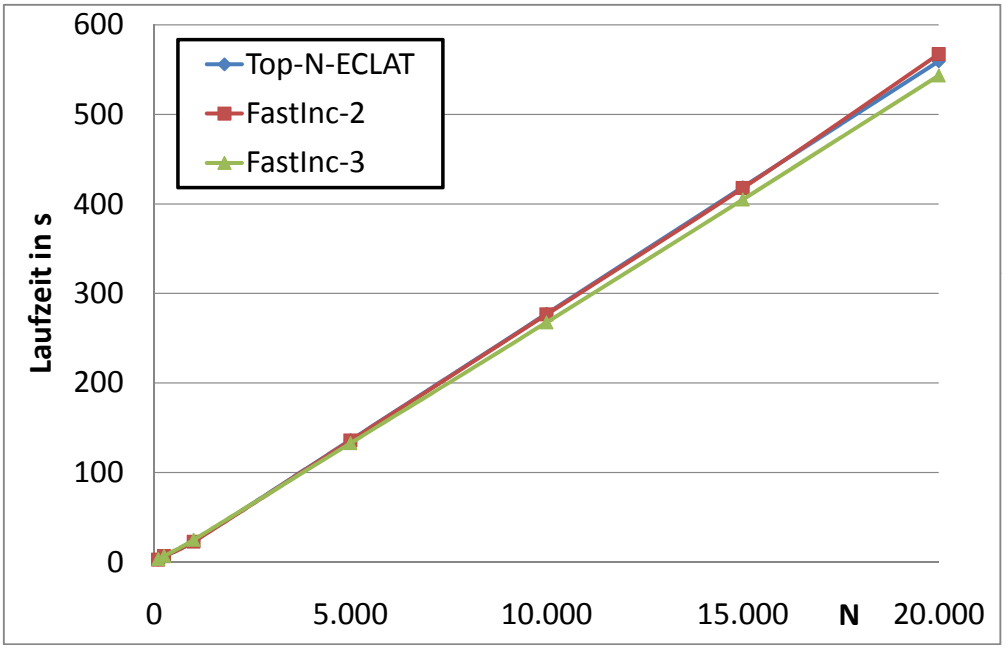
Zur Erzeugung von Assoziationsregeln müssen Kombinationen X mit mindestens zwei Elementen vorhanden sein. Nur diese sind in eine nichtleere Bedingung und Schlussfolgerung teilbar. 1-Kombinationen können selbst keine Assoziationsregel erzeugen, sondern lediglich Bedingung oder Schlussfolgerung einer Regel darstellen.

Unter Beachtung dieser Eigenschaft wird der Ausdruck *Top-N-Suche* im Rahmen dieser Evaluation als Suche nach den Top-N-Kombinationen mit mindestens zwei Elementen verstanden. Damit kann eine Top-N-Suche mehr als N Ergebnisse liefern. Einelementige Kombinationen mit größerer Häufigkeit als die N-te reguläre Kombination sind ebenfalls Teil des Ergebnisses. Hinzu kommen alle Kombinationen aus dem untersuchten Datenbestand mit zur N-ten ermittelte Kombination gleicher Häufigkeit.

Ziel dieser Evaluation ist, die Verringerung des Aufwandes für eine Top-N-EHK durch die Verwendung des zweistufigen Ansatzes mit FASTINC zu zeigen. Dabei soll die erwartete deutliche Steigerung der Algorithmeneffizienz an realen Szenarien geprüft werden. Die

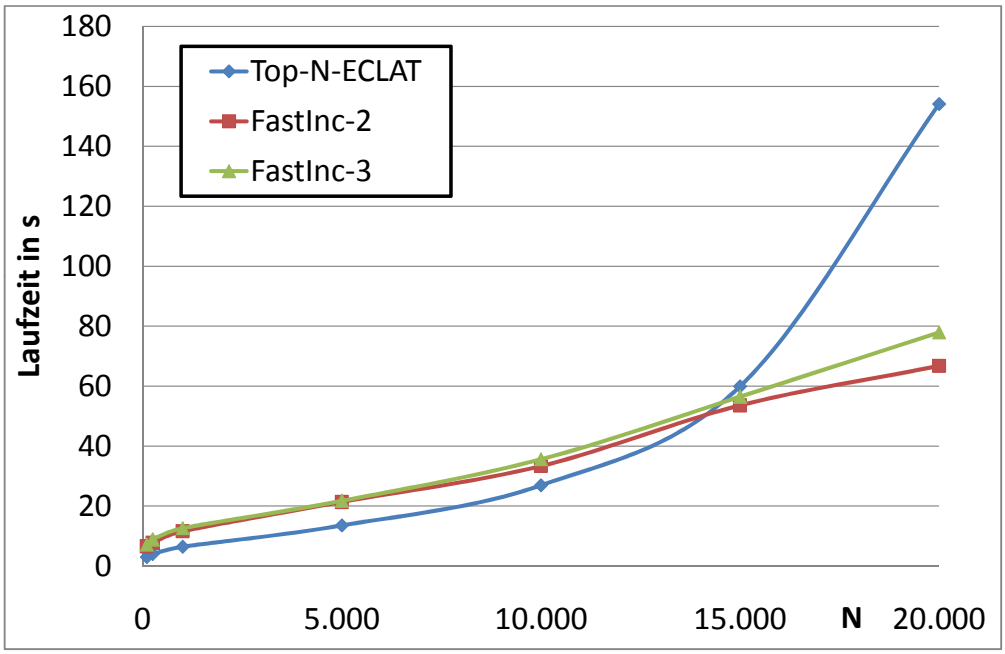


(a) Retail

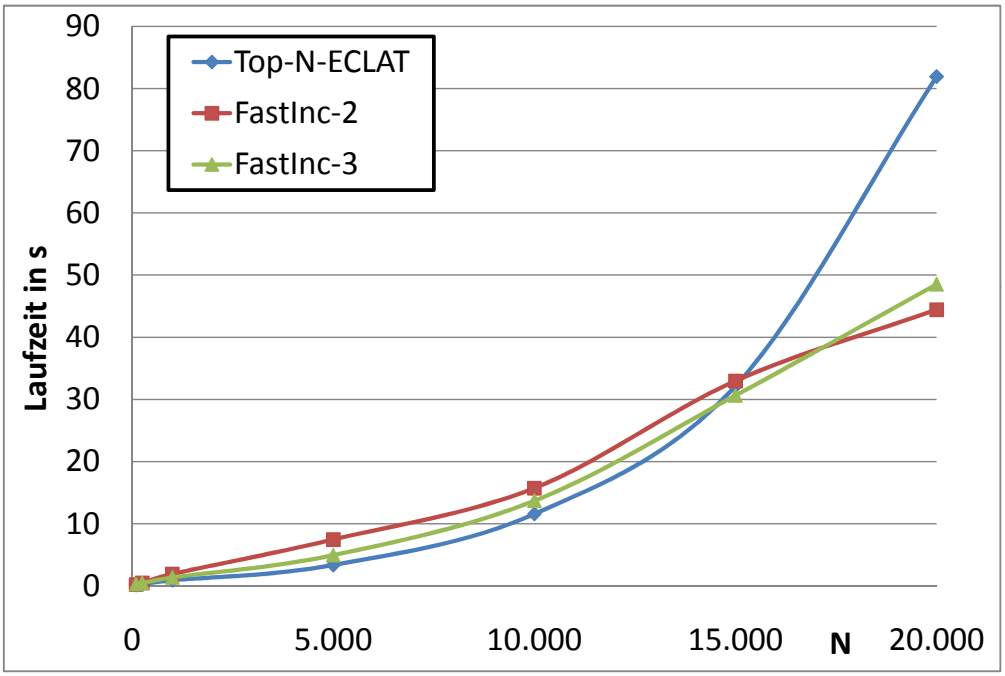


(b) Connect-4

Abbildung 3.7.: Top-N-Skalierung mit FASTINC (1)



(a) KundeB



(b) KundeA

Abbildung 3.8.: Top-N-Skalierung mit FASTINC (2)

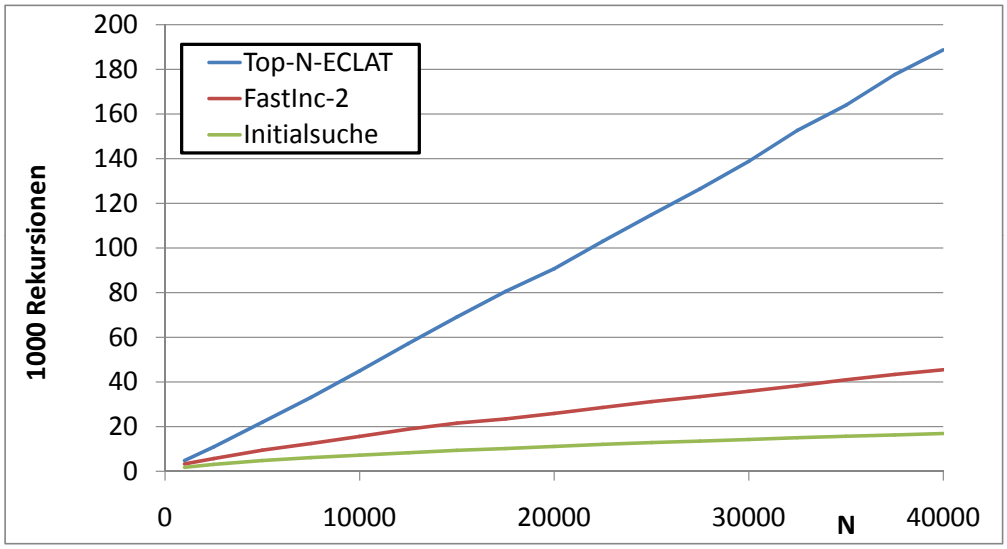
folgenden Abbildungen zeigen das Verhalten einer unverteilten Top-N-Suche und veranschaulichen die Auswirkung des FASTINC-Algorithmus. Die Abbildungen 3.7(a) bis 3.8(b) verdeutlichen den Effekt der Vorverarbeitung und die Auswirkungen verschiedener Initialsuchtiefen von FASTINC. Die Ergebnisse für *TOP-N-ECLAT* repräsentieren die Laufzeiten für eine Suche ohne Vorverarbeitung. TOP-N-ECLAT zeigt somit das Laufzeitverhalten ohne Optimierungen der Suchstrategie. Der Graph für *FASTINC-2* stellt die Zeit für eine maximale Initialsuchtiefe von $mpl = 2$ dar, *FASTINC-3* zeigt die Laufzeiten bei einer Suchtiefe von $mpl = 3$.

Die Auswirkungen des FASTINC-Parameters mpl sind abhängig von den untersuchten Daten. Sind viele Korrelationen in den Daten vorhanden und durchschnittlich sehr viele Elemente in einer Transaktion (siehe Beispiel Abbildung 3.7(b)), haben alle Top-N-Ergebnisse sehr ähnliche Häufigkeiten. Dadurch können auch Kombinationen mit vielen Elementen die Top-N-Ergebnismenge erreichen. Dies widerspricht den Annahmen über die positiven Auswirkungen der begrenzten Initialsuche von FASTINC. Die Häufigkeiten der Kombinationen im Kombinationsbaum nehmen nicht deutlich von links nach rechts und von oben nach unten ab. Wird diese Annahme verletzt, tritt keine signifikante Anhebung der unteren Schranke lb für die Häufigkeit der N-ten Kombination ein. Damit kann die Bestimmung des vollständigen Ergebnisses nur wenig beschleunigt werden. Trotz dieser ungünstigen Umstände können die Zusatzkosten für die Vorverarbeitung durch die nachfolgend effizientere Suche weitgehend kompensiert werden.

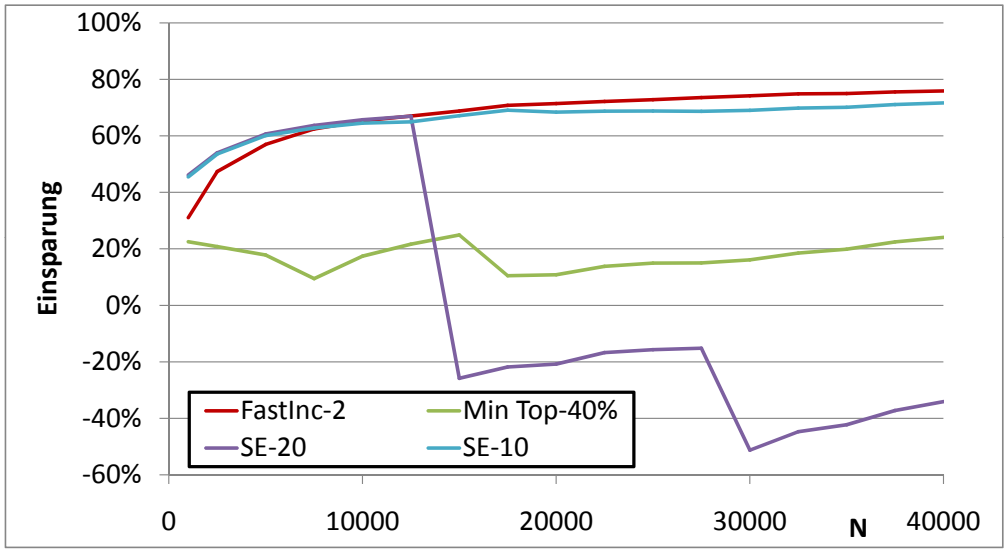
Für realistische Daten, dargestellt in Abbildung 3.7(a) und 3.8, wirkt sich die Initialsuche unterschiedlich stark aus. Für kleine N ist eine Initialsuche in diesen Beispielen nicht sinnvoll, da die Kosten dieser zusätzlichen Suche deren Nutzen teilweise bis um das Vierfache übersteigen. FASTINC weist jedoch mit wachsender Ergebnismenge eine bessere Skalierung auf als der Basisalgorithmus. Daraus resultiert eine Beschleunigung der Suche für große N. Aus Gesprächen mit SAP-Kunden ist eine Definition von $N > 10.000$ eine oft verwendete, praxisnahe Konfiguration. Die Ergebnisse werden oft nicht direkt verwendet, sondern gespeichert und in nachfolgenden Analysen weiterverarbeitet. Eine große Kombinationsmenge ist somit sinnvoll, um möglichst wenige potenziell wichtige Informationen zu verlieren. Für diese Szenarien stellt FASTINC-2 eine sinnvolle Optimierung dar. Der Gewinn für große N kann die Verluste bei kleinen N leicht kompensieren.

Für FASTINC-2 erreicht der Wert von lb bei den realistischen Daten zwischen 75% und 90% der Häufigkeit der korrekten N-ten Kombination. Eine Erhöhung der Maximaltiefe mit FASTINC-3 hebt den Wert auf 80% bis 100%. Diese Verbesserung benötigt jedoch teils deutlich mehr Ressourcen als sie einspart.

In den Abbildungen 3.5.2 wird gezeigt, wie Rekursionsaufrufe durch FASTINC-2 bei einer Top-N-Anfrage auf dem SynthC-Datenbestand gespart werden konnten bzw. wie viele Rekursionsaufrufe jeweils insgesamt erfolgten. In Abbildung 3.9(a) stellt der oberste Graph die Anzahl benötigter Rekursionsaufrufe in Abhängigkeit von N ohne jegliche Vorverarbeitung dar. Der mittlere Graph zeigt für die gleiche Konfiguration die gesamten,



(a) Rekursionsaufrufe FASTINC-2 gegenüber TOP-N-ECLAT



(b) Vermiedene Rekursionsaufrufe

Abbildung 3.9.: Rekursionsaufrufe auf SynthC-Datenbestand

benötigten Rekursionsaufrufe bis zum korrekten Endergebnis mittels FASTINC-2. Die initiale Suche ist hier bereits enthalten. Im unteren Graph wird der Anteil der Rekursionsaufrufe für die initiale Suche angegeben. In Abbildung 3.9(b) werden diese Ergebnisse im Vergleich zu zwei weiteren Vorverarbeitungsmöglichkeiten gezeigt. Die Graphen stellen den Quotienten aus der Anzahl vermiedenen Rekursionsaufrufe im Vergleich von den einzelnen Strategien und der naiven Verarbeitung mit TOP-N-ECLAT dar. *Min Top-40%* stellt dabei die Strategie dar, nach der eine Kombination beim Eintritt in die Ergebnismenge mindestens einen bestimmten Rang erreichen muss. Hierbei wurde mindestens eine Platzierung in den ersten 40% gefordert. Es ist ersichtlich, dass zwar eine Optimierung stattfindet, diese aber FASTINC-2 deutlich unterlegen ist. Die Graphen *SE-20* und *SE-10* zeigen die Strategie, nach der Kombinationen ins Ergebnis aufgenommen werden, solange sie häufiger sind als 20% bzw. 10% der Häufigkeit des nächsten betrachteten Einzelelementes. Hierbei ergibt sich für SE-10 eine vergleichbare Optimierung wie im Falle von FASTINC-2. Bei SE-20 wird aber die Instabilität dieser Strategie ersichtlich. Obwohl sie bei geringen N noch ein gutes Ergebnis erreicht, bricht die Optimierung bei steigendem N extrem ein und benötigt sogar mehr Rekursionsstufen als die naive Ausführung. Dies ist eine Folge dessen, dass innerhalb der Vorverarbeitung die Top-N nicht vollständig gefüllt werden konnten und damit die untere Schranke für den naiven Durchlauf nicht angehoben wurde.

In der Betrachtung aller Abbildungen ist auffällig, dass der Gewinn durch eingesparte Rekursionsaufrufe nur teilweise in Laufzeitverbesserungen übertragen werden kann. Dies ist begründet durch Implementierungsdetails und den daraus entstehenden Berechnungsaufwand der einzelnen Verarbeitungsschritte pro Rekursionsaufruf. Es werden zwar mittels FASTINC-2 deutlich weniger Rekursionsaufrufe benötigt, welche jedoch pro Aufruf im Durchschnitt häufigere Kombinationen betrachten. Damit müssen entsprechend in jedem Schritt mehr Transaktionen betrachtet werden. Der durchschnittliche Aufwand pro Rekursionsaufruf wird erhöht und reduziert den theoretisch möglichen Laufzeitgewinn. Es werden somit durch eine initiale Suche vor allem kostspielige Kombinationen mehrfach untersucht. Sind genügend Ressourcen vorhanden, können doppelt betrachtete Kombinationen auch durch Speicherung des ersten Durchlaufes optimiert werden. Dafür müssen die für jede Kombination nötige Liste mit zutreffenden Transaktionen gespeichert werden. Für realistische Anwendungen mit mehreren Millionen Transaktionen ist dieses Verfahren nicht sinnvoll. Szenarien mit geringerem Umfang können eine solche Zwischenspeicherung jedoch nutzen und damit die zusätzlichen Kosten der Vorverarbeitung reduzieren.

3.5.3. Verteilte Top-N-Kombinationen

Im folgenden Abschnitt sollen die Algorithmen im Zusammenhang mit einer verteilten Top-N-Suche betrachtet werden. Dies umfasst den STH und den MAST-Algorithmus. Zur Evaluation von verteilten Top-N-Suchen wurden folgende Einflüsse auf die Verfahren untersucht:

1. Anzahl der Partitionen.
2. Einfluss von N.
3. Verschiedene Datencharakteristiken.
4. Anzahl nicht garantierter Kombinationen.
5. Mindesthäufigkeit für TPARTITION.
6. Anzahl falscher oder nicht gefundener Ergebnisse.

Auf konkrete Laufzeiten wird weitestgehend verzichtet, da der verwendete Algorithmus zur Bestimmung von Top-N-Kombinationen für STH und MAST frei wählbar ist. Entsprechend werden lediglich spezifische Eigenschaften der in dieser Arbeit vorgestellten Verfahren betrachtet. Untersucht werden jeweils Top-N-EHK mit variierender Anzahl an Datenpartitionen auf diversen Datenbeständen. Der Fokus liegt hierbei auf Verkaufsdaten. Vermessen wurden mehrere Varianten der Top-N-EHK mit verändertem N. Das Ziel ist die Darstellung des Verhaltens des Grenzwertes s, der Güte des Ergebnisses und deren Abhängigkeiten vom Grad der Partitionierung der Daten. Erwartet wird hierbei, dass in Szenarien mit gleichmäßiger Partitionierung nur wenige Kombinationen nicht durch STH garantiert werden. In diesen Fällen ist gleichzeitig auch die Anwendung von TPARTITION möglich, wodurch STH lediglich etwa die halbe Laufzeit des entsprechenden TPARTITION beanspruchen sollte. Für ungleichmäßig verteilte Daten sollten weniger Kombinationen durch STH garantiert sein, aber gleichzeitig die Laufzeit von STH signifikant unter der des äquivalenten TPARTITION liegen.

Zu Beginn wird in Abbildung 3.10 das Laufzeitverhalten von STH im Vergleich zu einem vollständigen TPARTITION untersucht. Hierbei wird eine Version des *Retail*-Datenbestandes verwendet, welche in zwei Partitionen unterteilt wurde. Eine Partition enthält dabei lediglich die kürzeren Transaktionen, die andere Partition speichert die längeren Transaktionen. Dies simuliert eine Aufteilung mit unterschiedlichen Datencharakteristiken pro Partition. Das kann in realen Szenarien ebenfalls auftreten, wenn beispielsweise nach Filiale partitioniert wurde und dabei Kioskverkäufe mit denen von Handelsketten gemeinsam untersucht werden. Im Ergebnis zeigt sich deutlich, dass vor allem für große N der Mehraufwand von TPARTITION signifikant den von STH übersteigt. Insbesondere wächst die Laufzeit von TPARTITION exponentiell an, was die Verwendung des Verfahrens für ungeübte Nutzer erschwert.

Die Tabellen 3.3, 3.4 und 3.5 beinhaltet Informationen über die Eigenschaften der Ergebnisse einer EHK bei verschiedenen Konfigurationen und Datenbeständen in einer Übersicht. Dabei bezeichnet *Partitionen* die Anzahl der betrachteten Partitionen, *Grenze s* die jeweilige Sicherheitsgrenze von STH und $sup.Abs(I_N)$ die absolute Häufigkeit der N-ten Kombination. Dies stellt gleichzeitig die absolute Mindesthäufigkeit u des Algorithmus

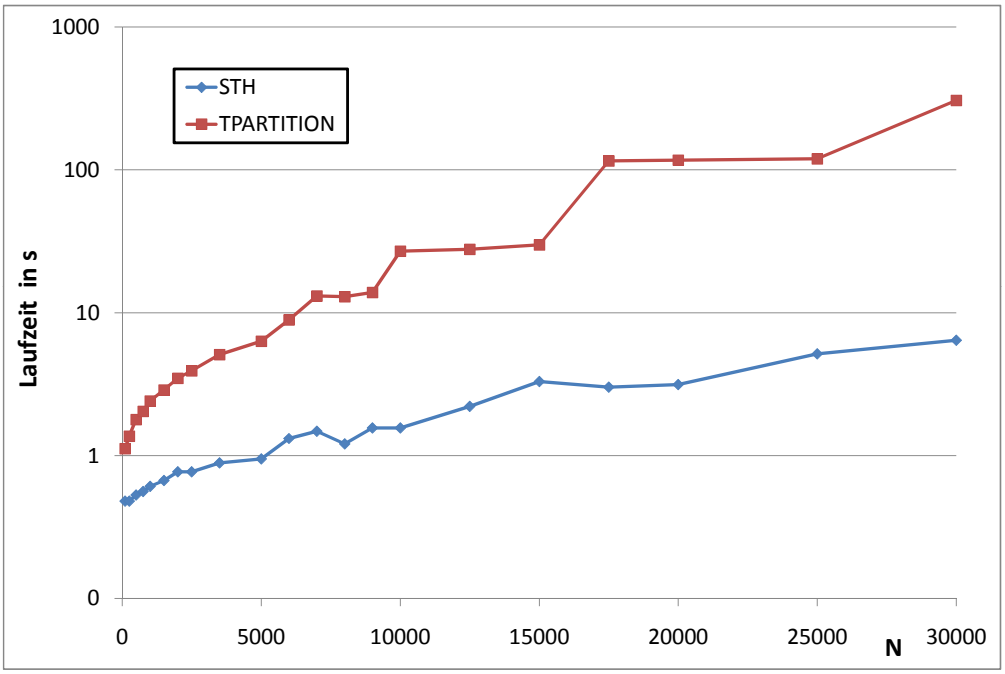
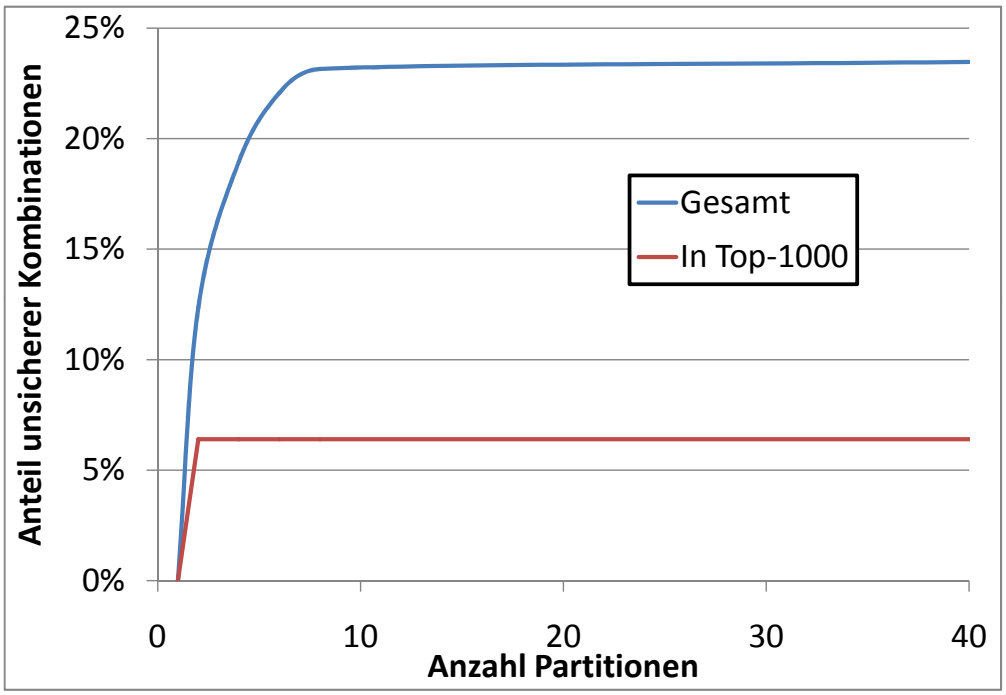


Abbildung 3.10.: STH und TPARTITION bei ungleichmäßigen Datencharakteristiken

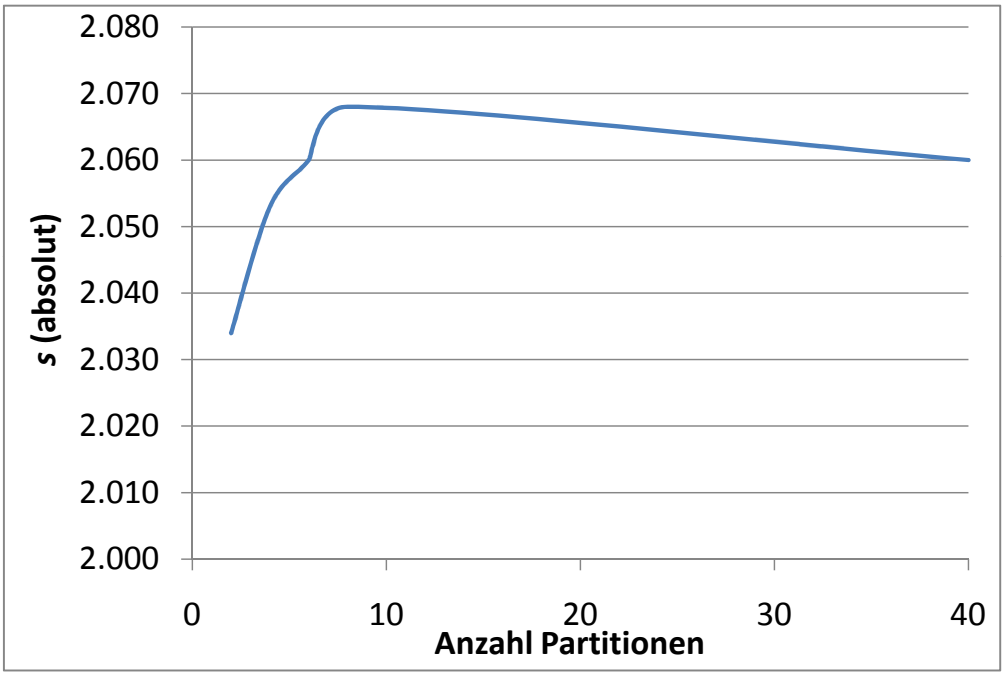
TPARTITION dar, mit dessen Hilfe das vollständige und korrekte Top-N-Ergebnis bestimmt werden kann. Dieser Wert ist nicht direkt aus den Daten ableitbar und benötigt eine initiale Top-N-Suche durch STH mit dem sich daraus ergebenden Aufwand. TPARTITION kann somit nicht effizienter werden als STH allein. Wenn keine nähere Beschreibung der Partitionierung erfolgt, gilt eine zufällige Aufteilung nach Transaktionskennzeichner und somit vergleichbare Datencharakteristiken der einzelnen Partitionen.

In der Spalte *Unsicher* wird die Anzahl an nicht zu garantierenden Kombinationen in der Ergebnismenge dargestellt. Die Spalte *Falsch erkannt* zeigt die Anzahl an Kombinationen, welche fälschlicherweise als Top-N-Ergebnis deklariert werden. α *für sichere Top-N* gibt den Wert für α an, ab welchem N Kombinationen auf diesem Szenario garantiert werden können. Ab den Top-$(\alpha \cdot N)$-Kombinationen wären also mindestens N Kombinationen garantiert. Wie leicht zu erkennen ist, wird für $\alpha = 1,07$ bei allen vorgestellten Beispielen das vollständige Top-N-Ergebnis bestimmt. Eine auffällige Eigenschaft ist hierbei die sinkende Anzahl nicht zu garantierender Kombinationen trotz steigendem N.

Eine wichtige Beobachtung bei dieser Untersuchung ist die weitestgehend pro Datenbestand stabile Grenze s von STH. Im Verhältnis zu den Häufigkeiten der Kombinationen ist die Änderung von s gering. Insbesondere ist s verhältnismäßig unabhängig von der Anzahl der beteiligten Partitionen. Der Grund für die leichten Schwankungen kann mit Änderungen in den Abhängigkeiten zwischen den Daten und den Datencharakteristiken erklärt werden, welche durch die jeweiligen Datenaufteilungen erzeugt werden.



(a) Anteil unsicherer Kombinationen



(b) STH Grenzwert s

Abbildung 3.11.: Nicht garantierbare Kombinationen (Top-1000, KundeB)

Abbildung 3.11 zeigt die Auswirkungen von STH auf die Ergebnisqualität in Abhängigkeit von der Partitionierung. Die Anzahl unsicherer Kombinationen in Abbildung 3.11(a) erhöht sich mit der Anzahl der Partitionen. Sie stabilisiert sich jedoch bei einem Anteil von etwa 23% unsicherer Einträge. Kombinationen mit der gleichen Häufigkeit wie die N-te Kombination sind hierbei jeweils Teil des Ergebnisses. Die Menge der davon betroffenen Kombinationen steigt mit zunehmender Partitionierung, da jeweils neue Ergebniskandidaten generiert werden. Diese Kombinationen werden den unsicheren Kombinationen hinzugefügt und bilden auch den größten Anteil von deren Wachstum. Die Güte des Ergebnisses wird hingegen sogar leicht verbessert, da durch neue Kandidaten weitere Kombinationen garantiert werden können.

Der untere Graph *In Top-1000* zeigt den prozentualen Anteil unsicherer Kombinationen innerhalb der ersten 1000 Kombinationen. Mehrfache Belegungen der Position 1000 werden also nicht betrachtet. Mit $6,3\%$ bleibt die Anzahl unsicherer Kombinationen über alle Partitionierungen konstant. Der Grund ist in Abbildung 3.11(b) erkennbar. Der absolute Grenzwert s von STH ändert sich nur geringfügig und erlaubt damit einen gleichbleibenden Umfang der Menge garantierter Kombinationen. Der Großteil der unsicheren Kombinationen findet sich somit zwar innerhalb der Top-1000, beschränkt sich jedoch auf weitestgehend auf mehrfach belegte schlechte Platzierungen.

Tabelle 3.4 zeigt eine Untersuchung für sehr ungleichmäßige Partitionsgrößen. Die dargestellten Spalten werden äquivalent zu Tabelle 3.3 bezeichnet. Zusätzlich gibt der Eintrag *Anzahl Kombinationen* die Kardinalität des ermittelten Ergebnisses an.

Für dieses Experiment wurde der KundeA-Datenbestand nach Datum in 22 bzw. 5 Partitionen zerlegt, 20 bzw. 3 mit wenigen Einträgen und jeweils zwei große Partitionen. Reguläre Kombinationssuchen können eine solche Datenaufteilung nicht effizient verarbeiten, da kleine Partitionen meist nicht in der Lage sind, ein sinnvolles lokales Top-N-Ergebnis zu erzeugen. Oft tritt dann der Fall ein, dass das N-te Ergebnis eine Häufigkeit von 1 aufweist und somit alle lokal vorhandenen Kombinationen zum Top-N-Ergebnis hinzugerechnet werden müssen. Damit erhöht sich die Laufzeit und die Anzahl erzeugter Kombinationen auf diesen Partitionen signifikant. Zum Vergleich wurde diese Suche mit dem regulären PARTITION-Algorithmus und der jeweils idealen Mindesthäufigkeit (entspricht TPARTITION) durchgeführt. Nach 20 Stunden musste der Lauf ohne Ergebnis abgebrochen werden. In realen Anwendungen hätten die 20 Stunden Auslastung deutliche Störungen der beteiligten Rechner bewirkt.

Zur Lösung wurde der in dieser Arbeit vorgestellte MAST-Algorithmus genutzt, welcher lediglich wenige Sekunden bis zur Berechnung des Ergebnisses benötigte. Für diese Experimente wurde für jede Partition eine minimale absolute Mindesthäufigkeit $mams = 2$ definiert. Da 20 bzw. 3 kleine Partitionen diese Grenze durch ihr lokales Top-N-Ergebnis nicht überschreiten konnten, wirkt sich die minimale absolute Mindesthäufigkeit mehrfach auf die globale Sicherheitsgrenze si aus. Ein derartig gehobener Grenzwert si bewirkt, dass global weniger Ergebnisse garantiert werden können. Trotzdem sind die meisten

Top-100	**Partitionen**	**Grenze s**	**sup.Abs** (I_N)	**Unsicher**	**Falsch erkannt**	α **für sichere Top-**100
	1		17.361	0	0	1,000
	2	17.378	17.361	1	0	1,030
	4	17.446	17.361	6	0	1,030
	6	17.426	17.361	5	0	1,050
	8	17.420	17.361	5	0	1,030
	10	17.398	17.361	3	0	1,030
	40	17.510	17.361	11	0	1,070
Top-10^3						**Top-**10^3
	1		9.227	0	0	1,000
	2	9.213	9.227	0	0	1,000
	4	9.228	9.227	1	0	1,001
	6	9.232	9.227	2	0	1,002
	8	9.233	9.227	2	0	1,002
	10	9.232	9.227	2	0	1,001
	40	9 249	9 227	4	0	1,006
Top-10^4						**Top-**10^4
	1		2.638	0	0	1,000
	2	2.636	2.638	0	0	1,000
	4	2.633	2.638	0	0	1,000
	6	2.634	2.638	0	0	1,000
	8	2.632	2.638	0	0	1,000
	10	2.631	2.638	0	0	1,000
	40	2.623	2.638	0	0	1,000

Tabelle 3.3.: Verhalten von STH auf SynthB

global ermittelten Kombinationen korrekt und entsprechen den Top-N-Ergebnissen eines äquivalenten unpartitionierten Datenbestandes. Begründet werden kann dies mit einem dominanten Einfluss der großen Partitionen auf das Endergebnis und die geringe Auswirkung der fehlenden Teilergebnisse der kleinen Partitionen. Dieses Verhalten konnte in praxisnahen Anwendungen vor allem in Kombination mit der durch die Anwendung spezifizierte Selektion relevanter Daten sowohl für PARTITION als auch für MAST bestätigt werden.

Top-N	Partitionen	si	sup.Abs Top-N	Un-sicher	Anzahl Kombinationen	Falsch erkannt
100	22	45	24	87	104	0
250	22	37	17	223	256	0
1.000	22	27	8	1.148	1.223	0
5.000	22	22	4	5.543	5.768	1
10.000	22	22	3	11.754	12.258	1
100	5	36	24	70	101	0
250	5	20	17	112	253	0
1.000	5	10	8	621	1.222	0
5.000	5	5	4	2508	5.764	1
10.000	5	5	3	9855	12.111	1

Tabelle 3.4.: MAST bei ungleichmäßigen Partitionsgrößen, KundeA mit $mams = 2$

Top-N	Partitionen	Grenze s	sup.Abs (I_N)	Un-sicher	Fehlende Kombinationen
100	2	23	24	0	0
250	2	16	17	0	0
1.000	2	7	8	0	1
5.000	2	2	4	0	0
10.000	2	2	3	0	0
100	4	24	24	0	0
250	4	15	17	0	0
1.000	4	8	8	0	15
5.000	4	2	4	0	0
10.000	4	0	3	0	0
100	10	22	24	0	0
250	10	13	17	0	0
1.000	10	9	8	276	0
5.000	10	0	4	0	0
10.000	10	0	3	0	0

Tabelle 3.5.: Verhalten von STH auf Datenbestand KundeA

3.5.4. Unscharf-geschlossene Kombinationen

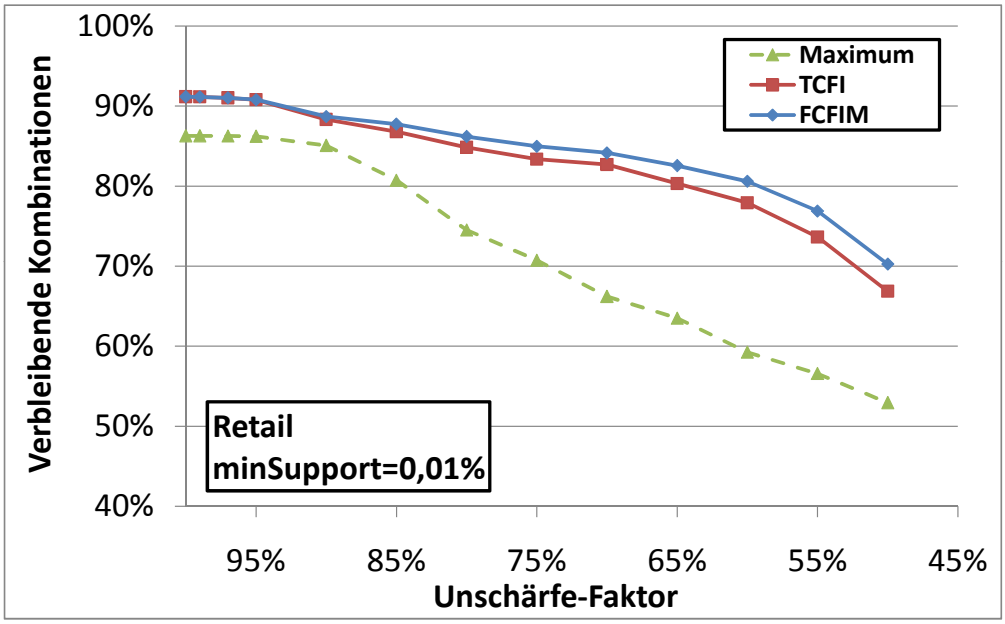
Der folgende Abschnitt untersucht die Auswirkungen auf das Ergebnis einer EHK bei der Verwendung unscharf-geschlossener Kombinationen. Betrachtet werden hierbei der in [JYP03] vorgestellte Algorithmus TCFI und der in dieser Arbeit erstellte Algorithmus FCFIM. Beide Algorithmen nutzen jeweils lediglich die vom Algorithmus ACHARM bekannte approximierte Bestimmung (unscharf-)geschlossener Kombinationen. Die Evaluation verdeutlicht die Qualität des Ergebnisses bei der Verwendung von FCFIM. Die maximal erreichbare Kompression des Ergebnisses ohne Approximierung soll daher, wenn nötig, durch die Bezeichnung *Maximum* repräsentiert werden.

Die Bestimmung von *Maximum* ist ein aufwändiger Vorgang für unscharf-geschlossene Kombinationen, da hierfür die Reihenfolge der Vereinigung der Kombinationen relevant ist. Theoretisch müssen also alle diese Möglichkeiten ausgewertet werden. Für diese Evaluation wurde die Suche auf 100 zufällige Reihenfolgen beschränkt und deren höchste Kompression als Referenz *Maximum* verwendet. Dabei zeigte sich, dass dieser Wert bei einer Vielzahl von Durchläufen annähernd erreicht wurde. Daher soll er an dieser Stelle vereinfacht als Obergrenze für die mögliche Kompression akzeptiert werden.

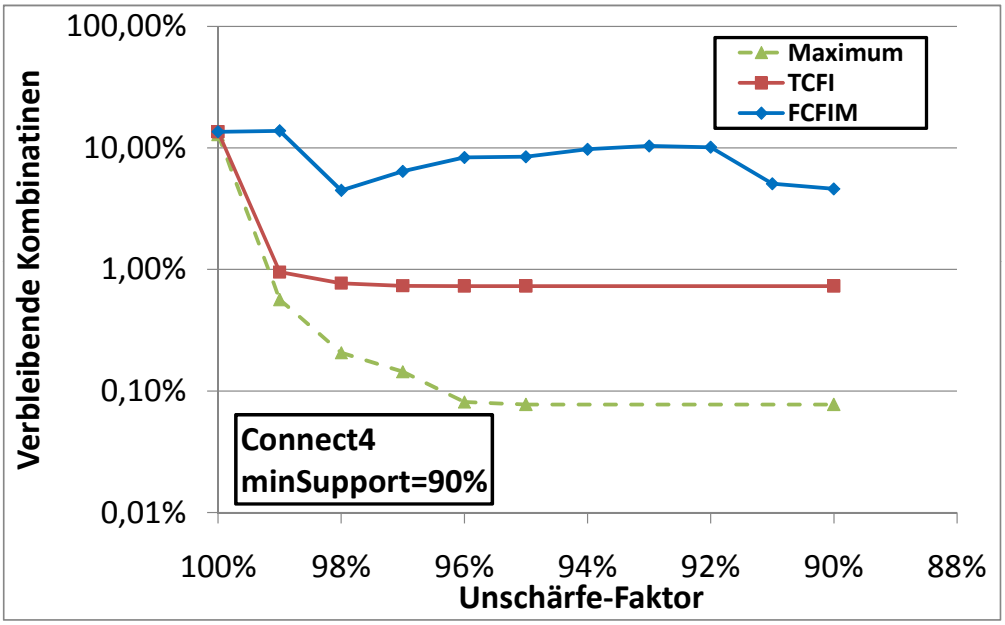
Abbildung 3.12 zeigt die Auswirkungen des Unschärfefaktors auf die dabei entstehende Ergebnismenge. Der als *Maximum* bezeichnete Graph stellt die mögliche Kompression bei nicht approximierten Verfahren dar. Ein Unschärfefaktor von 100% repräsentiert eine Suche nach vollständig geschlossenen Kombinationen. Für weniger als 100% wurde entsprechend die Menge der unscharf-geschlossenen Kombinationen bestimmt. Der dargestellte Prozentwert gilt jeweils in Relation zur vollständigen, nichtgeschlossenen Ergebnismenge.

Betrachtet werden sollen der Retail-Datenbestand als Vertreter der in dieser Arbeit adressierten Verkaufsdaten, sowie die Connect4-Daten. Letztere dienen in vergleichbaren Arbeiten oft als Beispiel für die Vorteile einer Suche nach geschlossenen Kombinationen gegenüber einer vollständigen EHK. Die Auswertung der Retail-Verkaufsdaten ist hierbei in diese Arbeit relevanter als der Extremfall der Connect-4-Daten. Für die Verkaufsdaten wurde eine Mindesthäufigkeit von $0,01\%$ gewählt, was ca. 143.000 gefundenen Kombinationen entspricht. Die Connect4-Daten wurden mit einer Mindesthäufigkeit von 90% untersucht, was ca. 27.000 häufige Kombinationen bestimmt. Der Wertebereich des Unschärfefaktors wurde in beiden Darstellungen so gewählt, dass entweder das daraus resultierende Ergebnis noch sinnvoll ist oder keine relevanten Änderungen der Kompression trotz erhöhter Unschärfe mehr feststellbar waren. Zu beachten ist hierbei auch die unterschiedliche Einteilung der Y-Achsen. Dies ist nötig, weil sich die jeweiligen Kompressionsraten bei den dargestellten Daten um Größenordnungen unterscheiden.

Der als *FCFIM* bezeichnete Graph stellt die Kardinalität des Ergebnisses unter Nutzung von ACHARM/FCFIM dar. Mit *TCFI* wird das resultierende Ergebnis durch Verwendung der Verfahren TCFI von Jia, Yao und Pei [JYP03] gezeigt. Der jeweils mögliche



(a) *Retail*



(b) *Connect-4*

Abbildung 3.12.: Ergebniskompression

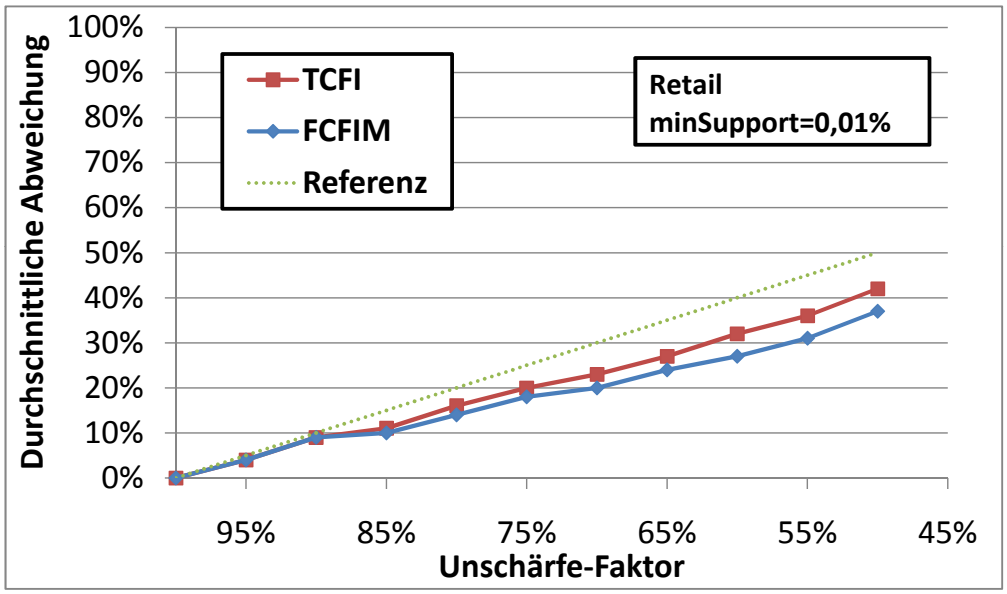
maximale Fehler ist bei FCFIM konstant, wohingegen er bei TCFI pro Vereinigungsschritt toleriert wird. Der Gesamtfehler wird für TCFI somit aufsummiert, wodurch er insgesamt den jeweiligen gewählten Unschärfefaktor deutlich überschreiten kann. Bei dem in dieser Arbeit vorgestellten FCFIM ist diese Erhöhung des Fehlers nicht möglich.

Der Unterschied zwischen der bei TCFI und FCFIM verwendeten Toleranz ist für die in Abbildung 3.12(a) gezeigten Verkaufsdaten gering. Eine Rekonstruktion des vollständigen Ergebnisses ist mit Hilfe von FCFIM mit einer maximalen Häufigkeitsabweichung für alle fehlenden Kombinationen möglich, während die TCFI-Ergebnisse teilweise deutlich größere Abweichungen zeigen. Abbildung 3.13 verdeutlicht, dass für TCFI der Unschärfefaktor signifikant überschritten werden kann. Zu erkennen ist dies im Vergleich zum Graph *Referenz*, welcher den jeweilig gewählten Unschärfe-Faktor repräsentiert. Die Kompression der Ergebnismenge ist im Vergleich zu TCFI bei FCFIM trotz besserer Qualität ähnlich. Für die Connect4-Daten reduziert TCFI die Ergebnismenge effizienter als FCFIM, was jedoch maßgeblich durch die deutlich geringere Minimalgenauigkeit bewirkt wird. Der entstehende maximale Fehler wird bei TCFI deutlich größer als bei FCFIM. Der durchschnittliche Fehler ist für beide Datenbestände und jeweils für TCFI und FCFIM ähnlich. Er ist jedoch nur bei FCFIM garantiert geringer oder gleich dem gewählten Unschärfefaktor.

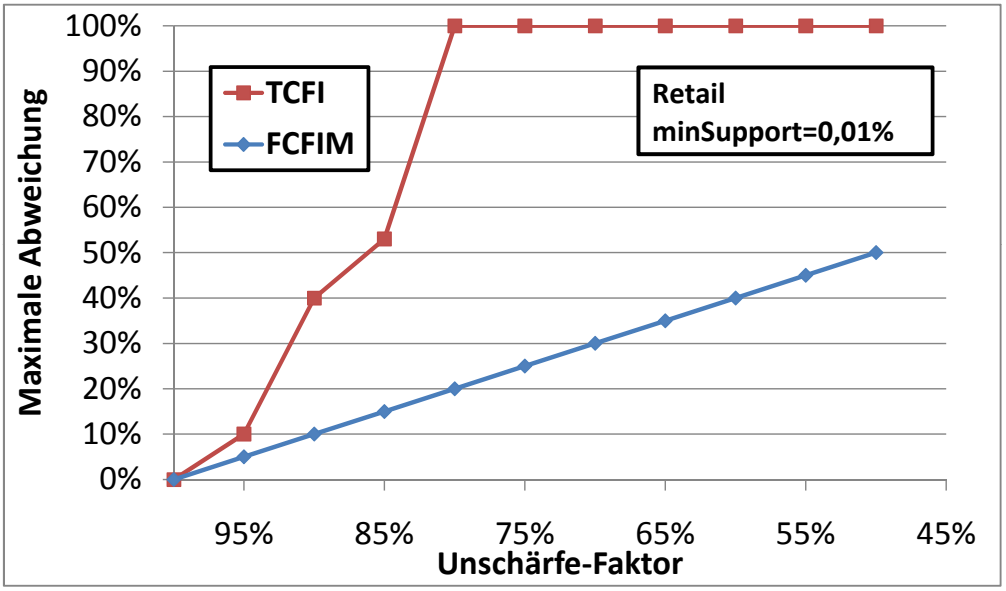
Für beide Datenbestände zeigt sich auch deutlich der Einfluss der verwendeten Vereinigungsstrategie für unscharf-geschlossene Kombinationen. Trotz des bei TCFI deutlich größeren Fehlers erreicht die optimierte Variante kompaktere Ergebnismengen. Zudem zeigt die extreme Reduktion der Ergebnismenge bei den Connect4-Daten die geringe Diversität der vollständigen Ergebnisse. Schon ab einem Unschärfefaktor von 95% bleibt die resultierende Ergebnismenge konstant.

Da bei verändertem Unschärfefaktor die Reihenfolge der Vereinigung von unscharf-geschlossenen Kombinationen variiert, kann die Ergebnismenge trotz steigendem Unschärfefaktor anwachsen. Erkennbar wird dies in Abbildung 3.12(b). Dieser Effekt ist jedoch nur zu beobachten, wenn sehr viele Kombinationen nicht unscharf-geschlossen sind und aus dem Ergebnis entfernt werden. Verkaufsdaten weisen diese Charakteristik nicht auf und verzeichnen den Effekt nur in geringem Maße.

Abbildung 3.15 stellt die benötigte Laufzeit für beide Datenbestände in Relation zu eine Suche der vollständigen häufigen Kombinationen mit ECLAT dar. Die normierte Laufzeit (Beschleunigungsfaktor=1) mit ECLAT betrug für die Connect4-Daten etwa 220 Sekunden, für die Retail-Daten wurden weniger als drei Sekunden benötigt. Dabei wird gezeigt, dass die durch eine Betrachtung einer Unschärfe entstehenden Kosten denen einer regulären Suche nach geschlossenen Kombinationen entsprechen oder diese unterschreiten. Somit gilt auch für die Suche nach unscharf-geschlossenen Kombinationen das in Abschnitt 2.2.7 gezeigte Laufzeitverhalten. Sowohl unscharf-geschlossene als auch geschlossene Kombinationen benötigen zur Evaluation jeweils einen geringen Mehraufwand pro untersuchten Knoten und beschleunigen die Suche entsprechend der Anzahl



(a) Durchschnittliche Genauigkeit



(b) Maximaler Fehler

Abbildung 3.13.: Ergebnisqualität *Retail*

der gegenüber der vollständigen Suche reduzierten Kombinationen. Eine mögliche Laufzeitverbesserung hängt daher signifikant von den untersuchten Daten ab. Für typische Verkaufsdaten ist der Laufzeitgewinn meist gering und die reduzierte Ergebnismenge stellt den größeren Vorteil für den Anwender dar.

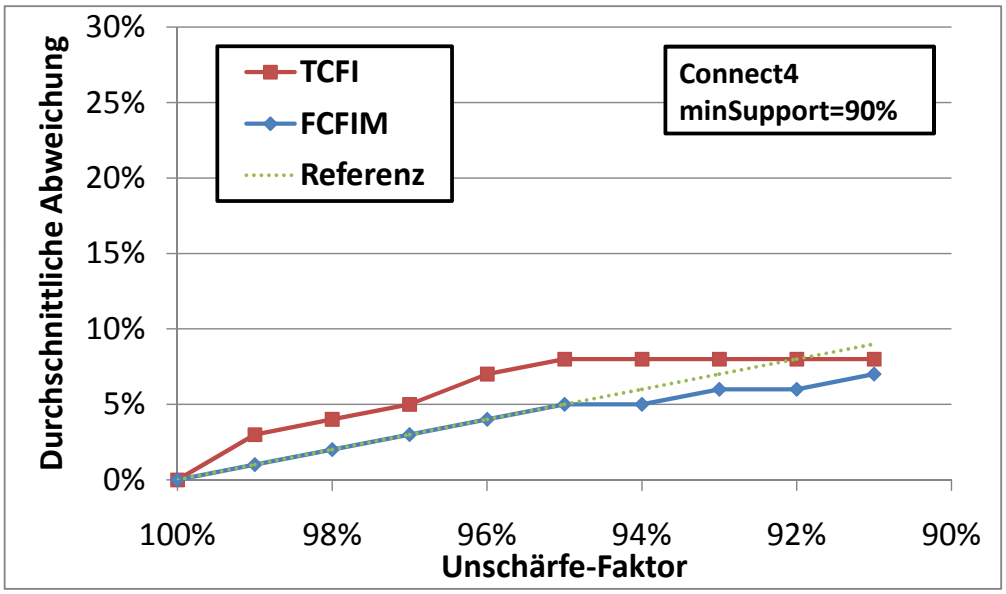
Die Laufzeit vom Retail-Datenbestand wurde beispielsweise wie in Abbildung 3.15(b) bei einem Unschärfefaktor 80% lediglich von $3,1$ Sekunden auf $2,9$ Sekunden gesenkt. Gegenüber der Suche nach den vollständigen häufigen Kombinationen wurde die Laufzeit sogar geringfügig erhöht. Hier ist vor allem die Reduktion der Ergebnismenge und die daraus resultierende vereinfachte Auswertung der Ergebnisse wichtig. Verkaufsdaten sind oft nur in geringem Maße nicht geschlossen. Die Verwendung von unscharf-geschlossenen Kombinationen kann bei sehr geringem Aufwand die mögliche Reduktion derartiger Ergebnisse erhöhen. Vor allem für den Fall, dass der dabei maximal entstehende Fehler bekannt ist, wird die Nutzung von unscharf-geschlossenen Kombinationen mit FCFIM bei Verkaufsdaten sinnvoll.

3.5.5. Heuristische Parameterbestimmung

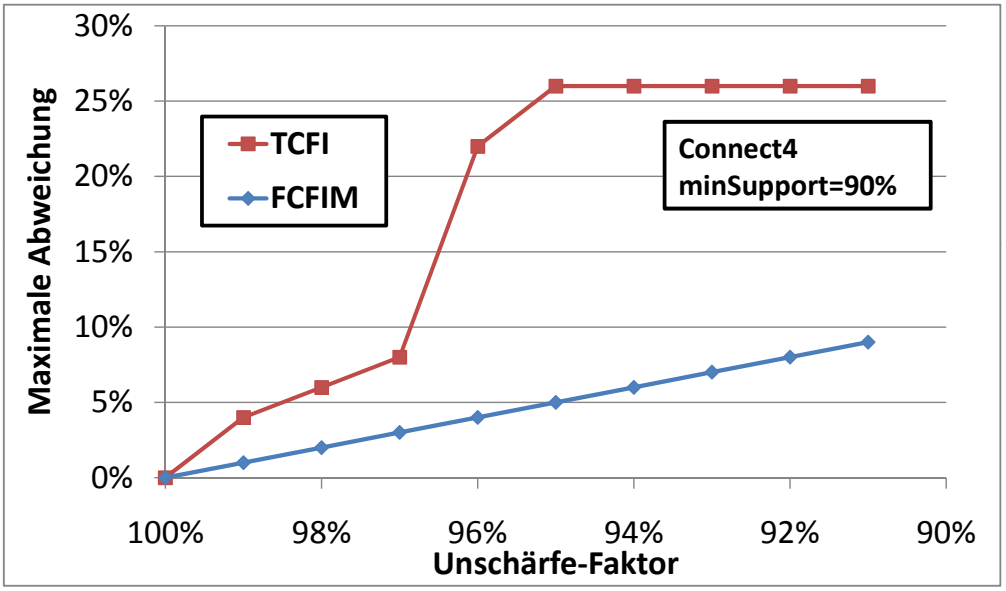
In Tabelle 3.6 wird die Schätzung einer sinnvollen Mindesthäufigkeit gezeigt, wenn eine EHK mittels EHKSIM simuliert wird. Die Häufigkeiten betrachteter Kombinationen werden nicht ermittelt. Stattdessen wird diese aus den Häufigkeiten der in der Kombination enthaltenen Elemente geschätzt. Das Ergebnis entspricht somit der Menge der statistisch zu erwartenden Ergebnisse einer EHK mit einer bestimmten Mindesthäufigkeit. Die Wahl der zu testenden Mindesthäufigkeiten erfolgt iterativ, wobei diese Evaluation lediglich einige repräsentative Beispiele für die jeweiligen Datenbestände zeigt. Diese Auswertung zeigt die geschätzte Ergebnismenge im Vergleich zu dem bei einer vollständigen EHK korrekt bestimmten Ergebnis. Zeigt dies für sehr unterschiedliche Datenbestände vergleichbare Größenordnungen, wird gezeigt, dass EHKSIM einen möglichen Ansatz zur Bestimmung sinnvoller Parameter der EHK darstellt.

Die Laufzeit von EHKSIM ist unabhängig von der Menge der zugrunde liegenden Daten, da lediglich die relativen Häufigkeiten der Einzelelemente verwendet werden. Die Laufzeit einer EHK steigt mit der betrachteten Datenmenge. Somit wird EHKSIM umso effizienter, je umfangreicher die vollständigen Daten und die darauf erfolgende EHK sind.

In Tabelle 3.6 stellen *Geschätzte Kombinationen* die vermutete Ergebniskardinalität und *Kombinationen* die Menge der tatsächlichen Ergebnisse dar. Die Menge der daraus resultierenden möglichen Assoziationsregeln sind in *Anzahl Regeln* zu finden. Die Spalte $|I_s|/|I|$ zeigt, wie die zu erwartende Ergebniskardinalität über- bzw. unterschätzt wurde. Je näher dieser Wert an 100% ist, umso genauer war die Vorhersage. Im Vergleich ergeben sich für einige Datenbestände lediglich geringe Abweichungen. Ein großer Anteil der Beispiele schätzt die richtige Ergebniskardinalität auf $\pm 20\%$ genau, was für eine Schät-

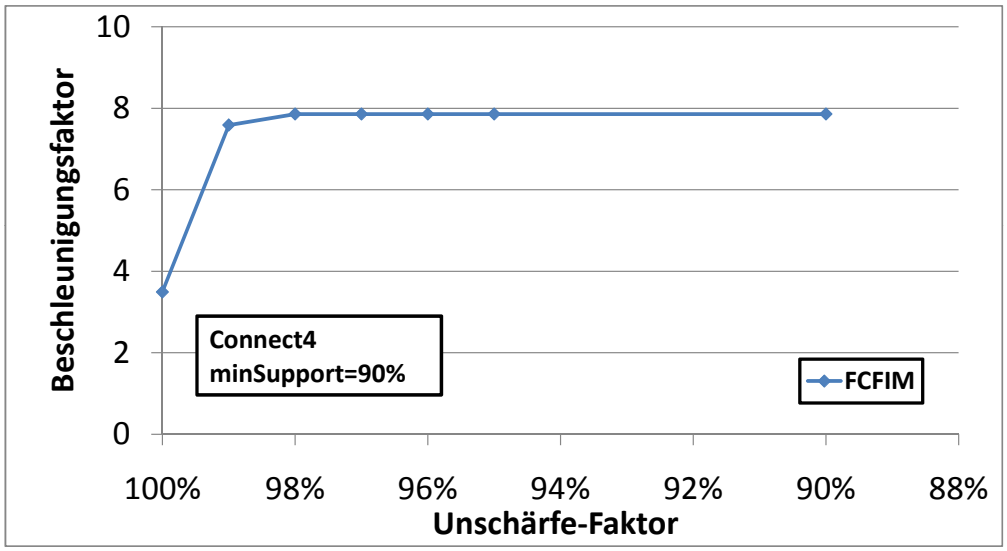


(a) Durchschnittliche Genauigkeit

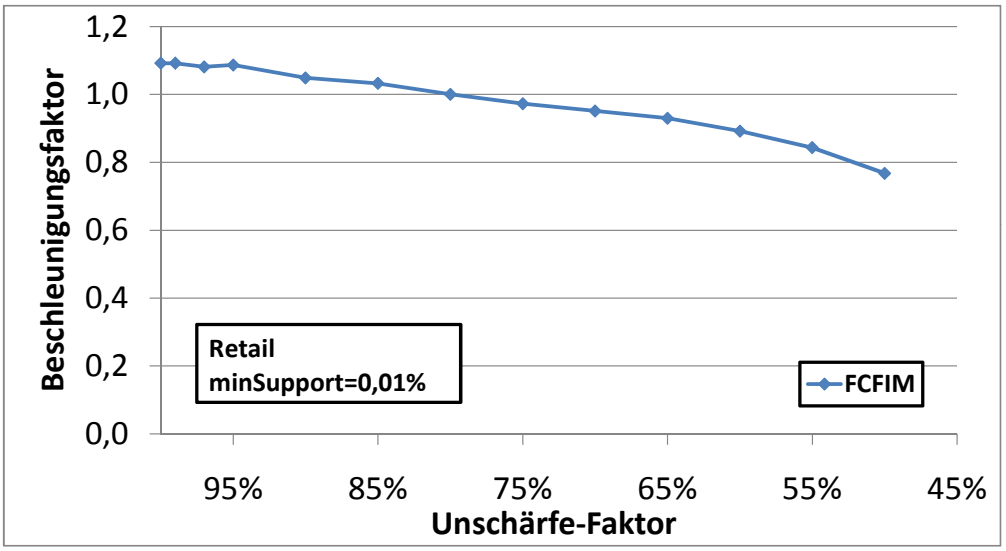


(b) Maximaler Fehler

Abbildung 3.14.: Ergebnisqualität *Connect-4*



(a) *Connect-4*



(b) *Retail*

Abbildung 3.15.: Beschleunigung FCFIM gegen ECLAT

zung der zu erwartenden Größenordnung einer Ergebnismenge oft ausreicht. Vor allem bei Datenbestand SynthA sind jedoch deutliche Abhängigkeiten in den Daten vorhanden, wodurch das Ergebnis der Schätzung stark von der Realität abweicht. Die Unterschätzung um jeweils ca. das achtfache reicht jedoch auch in diesem Fall zum Vermeiden einer Ergebnisexplosion aus. In den gezeigten Beispielen kann somit aus einer iterativ durchgeführten simulierten EHK auf die Größenordnung der korrekten Ergebnismenge geschlossen werden.

3.6. Zusammenfassung

In diesem Kapitel wurde die Vereinfachung von Verfahren zur Extraktion häufiger Kombinationen (EHK) untersucht. Dabei wurden sowohl technische Aspekte, wie eine verteilte Verarbeitung, als auch Möglichkeiten zur einfachen Nutzbarkeit erläutert. Diese konnten durch Verwendung von Top-N-Verfahren erreicht werden. Bisherige Lösungen erfordern meist die Wahl einer Mindesthäufigkeit, um die resultierende Ergebnismenge einzuschränken. Unerfahrene Anwender können damit leicht überfordert werden. Diesem Nutzerkreis fällt die Wahl einer gewünschten Ergebnismenge oft leichter.

Hierfür wurde in diesem Kapitel eine Top-N-Erweiterung für den EHK-Algorithmus ECLAT [ZPOL97] vorgeschlagen. Diese wurde mittels einer Optimierung mit dem Namen FASTINC durch heuristische Vorabsuchen beschleunigt und erlaubt eine Top-N-EHK mit geringem Mehraufwand gegenüber bisherigen mindesthäufigkeitsbasierten Lösungen.

Zusätzlich erschweren einige Datencharakteristiken die EHK. Für diese ist insbesondere eine verteilte Berechnung schwierig. Besonderheiten der Gesamtdaten sind nur bedingt in den Einzelpartitionen erkennbar. Um eine Suche effizient durchzuführen, müssen somit aus lokalen Informationen globale Eigenschaften geschätzt oder angenommen werden. Für mindesthäufigkeitsbasierte Lösungen kann der Algorithmus PARTITION [SON95] verwendet werden. Diese Arbeit stellte unter dem Namen TPARTITION ein Verfahren zur Berechnung verteilter Top-N-Kombinationen vor. Diese Lösung wurde mit dem Algorithmus Save-Threshold-Algorithmus (STH) optimiert, liefert aber ein nicht vollständig garantiertes Ergebnis. Hierfür wurde die Herleitung einer Grenze erläutert, ab deren Erreichen ein Ergebnis als garantiert korrekt gilt.

Bei partitionierten Daten kommt hinzu, dass der Umfang der einzelnen Datenbereiche sehr unterschiedlich sein kann. Das hat zur Folge, dass auf sehr kleine Partitionen alle Kombinationen mit sehr seltenem lokalem Auftreten bestimmt werden müssen. Auch für wenige betrachtete Daten können hier lokal extrem viele Kandidaten erzeugt werden. Bisherige Algorithmen setzen eine gleichmäßige Partitionierung der Daten voraus, doch realistische Szenarien erfüllen diese Voraussetzung oft nicht.

Retail				
minSupport	**Geschätzte Kombinationen** $\lvert I_s \rvert$	**Kombinationen** $\lvert I \rvert$	**Anzahl Regeln** $\lvert J \rvert$	$\lvert I_s \rvert / \lvert I \rvert$
1,00%	82	105	122	78%
0,50%	292	363	412	80%
0,25%	977	1.255	1.689	78%
0,10%	3.504	4.832	9.112	73%
0,05%	7.905	12.286	320.556	64%
Connect-4				
minSupport	**Geschätzte Kombinationen** $\lvert I_s \rvert$	**Kombinationen** $\lvert I \rvert$	**Anzahl Regeln** $\lvert J \rvert$	$\lvert I_s \rvert / \lvert I \rvert$
99,90%	1	0	0	—
99,50%	9	8	12	113%
99,00%	29	29	70	100%
90,00%	21.297	27.127	3.640.704	79%
85,00%	106.444	142.127	—	75%
KundeA				
minSupport	**Geschätzte Kombinationen** $\lvert I_s \rvert$	**Kombinationen** $\lvert I \rvert$	**Anzahl Regeln** $\lvert J \rvert$	$\lvert I_s \rvert / \lvert I \rvert$
1,000%	3	2	0	150%
0,500%	17	16	0	106%
0,250%	43	642	0	102%
0,100%	217	216	0	100%
0,050%	672	6.677	16	99%
0,010%	5.346	5.934	3.002	90%
0,005%	10.875	16.142	113.256	67%
SynthA				
minSupport	**Geschätzte Kombinationen** $\lvert I_s \rvert$	**Kombinationen** $\lvert I \rvert$	**Anzahl Regeln** $\lvert J \rvert$	$\lvert I_s \rvert / \lvert I \rvert$
5,00%	69	271	2.320	25%
2,50%	160	990	7.560	16%
1,50%	294	1.575	13.508	19%
1,00%	652	2.642	24.512	25%
0,75%	1.155	5.033	72.986	23%
0,50%	2.412	16.231	443.930	15%
0,35%	4.232	31.810	1.063.616	13%
0,05%	61.906	224.082	—	28%

Tabelle 3.6.: Häufige Kombinationen bei simulierter EHK

In dieser Arbeit wurde zur Lösung eine minimal erlaubte absolute Mindesthäufigkeit vorgeschlagen. Die Ergebnisexplosion auf sehr kleinen Partitionen kann damit verhindert werden. Das vollständige Ergebnis wird hingegen nicht garantiert. Dieser als MAST bezeichnete Algorithmus wurde mit dem STH Algorithmus kombiniert. Die Berechnung des Grenzwertes für garantierte Ergebnisse bei STH und MAST wurden dabei vereinigt und bilden einen gemeinsamen Schwellwert.

Als Alternative zur Verwendung von Top-N-Strategien wurden Möglichkeiten zur Ergebnisreduktion und zur Bestimmung sinnvoller Mindesthäufigkeiten diskutiert. Es wurde gezeigt, wie der Aufwand zur Bestimmung einer Obermenge geschlossener Kombinationen durch Approximation reduziert werden kann. Hierfür wurde auf eine vollständige Prüfung auf Geschlossenheit verzichtet, welche den Großteil der entstehenden Zusatzkosten verursacht.

Als weitere Möglichkeit zur Ergebnisreduktion wurden unscharf-geschlossene Kombinationen vorgestellt. Entgegen geschlossenen Kombinationen sind dabei maximale Abweichungen der Häufigkeiten zweier Kombinationen erlaubt. Der Umfang der Ergebnismenge kann dabei verringert werden, ohne wichtige Informationen zu verlieren. Das ermöglicht eine effiziente Verarbeitung des Ergebnisses.

Es wurde zudem eine Möglichkeit zur Schätzung günstiger Mindesthäufigkeiten gezeigt. Diese nutzt die Erwartungswerte der Häufigkeiten von Kombinationen, um den zu erwartenden Ergebnisumfang abzuschätzen. Dies bietet einen Anhaltspunkt zur Wahl der Mindesthäufigkeit und erlaubt eine effiziente manuelle Parametersuche, wobei Garantien jedoch nicht möglich sind.

Abschließend wurden alle vorgestellten Verfahren evaluiert. Dabei wurden synthetische und realistische Daten in verschiedenen Partitionierungen verwendet. Dabei konnte der Nutzen der vorgestellten Algorithmen gezeigt werden. Zur Ergebnismengenschätzung und Reduktion der Ergebnismenge wurde an Beispielen die Wirkungsweise der vorgestellten Verfahren dargelegt. Insgesamt zeigten die Erkenntnisse dieses Kapitels eine nutzerfreundliche EHK für ungeübte Nutzer.

4. Aufbereitung von häufigen Kombinationen zu Assoziationsregeln

Nachdem im vorherigen Kapitel die für die Assoziationsregelsuche nötige Datenanalyse und deren Durchführung untersucht wurden, befasst sich dieses Kapitel mit der automatischen Aufbereitung der Regeln. Entsprechend gelten in dieses Kapitel die Menge von Assoziationsregeln J als gegeben. Außerdem sollen für alle Regeln die Häufigkeiten von deren Bedingung, Schlussfolgerung und Vereinigung bekannt sein. Das Ergebnis der EHK bietet diese Informationen. Darauf aufbauend wird die Berechnung der in Abschnitt 2.3 vorgestellten Interessantheitsmaße zur Bewertung der Regel ermöglicht.

Assoziationsregeln beschreiben die Korrelation zwischen der Bedingung und Schlussfolgerung einer Regel, beispielsweise die Verbindung zwischen dem Verkauf zweier Produkte. Im Abschnitt 2.3 wurden bereits einige wichtige Interessantheitsmaße vorgestellt, welche Bewertungen der Regel ermöglichen. Diese Maße geben jedoch lediglich einen Hinweis auf den potenziellen, subjektiven Wert der Regel. Da bei den vorgestellten Maßen objektive, statistische Besonderheiten einer Regel im Vordergrund stehen und die subjektive Sicht des Anwenders nicht bekannt ist, bleibt die Wichtigkeit der Regel aus Sicht der automatischen Bewertung ungewiss. Insbesondere der subjektive Bewertungsmaßstab des Anwenders ist nicht bekannt.

Um das Problem zu verringern, ermitteln realistische Anwendungen eine große Anzahl Regeln aus den zu untersuchenden Daten. Damit soll der Anteil des extrahierten Wissens maximiert werden, indem auf möglichst wenige Regeln verzichtet wird. Dabei werden auch Regeln bestimmt, welche nach einigen Interessantheitsmaßen unwichtig erscheinen.

Eine manuelle Auswertung dieser Regelmengen ist oft nur begrenzt möglich, da die menschliche Wahrnehmung mit dieser Informationsmenge überfordert wird. Eine Betrachtung der Regeln unter verschiedenen Gesichtspunkten wird nötig. Mit der Annahme, dass ein Anwender seine subjektiv wichtigen Sichtweisen beschreiben kann, wird die Menge aller Regeln auf die jeweils wichtigen Regeln eingeschränkt. Dieser Abschnitt soll Lösungen zeigen, die dem Anwender eine Hilfe zur Wahl von interessanten Sichten bieten. Dies umfasst sowohl eine automatische Bewertung betrachteter Regeln, als auch deren nutzerfreundliche Aufbereitung und Darstellung.

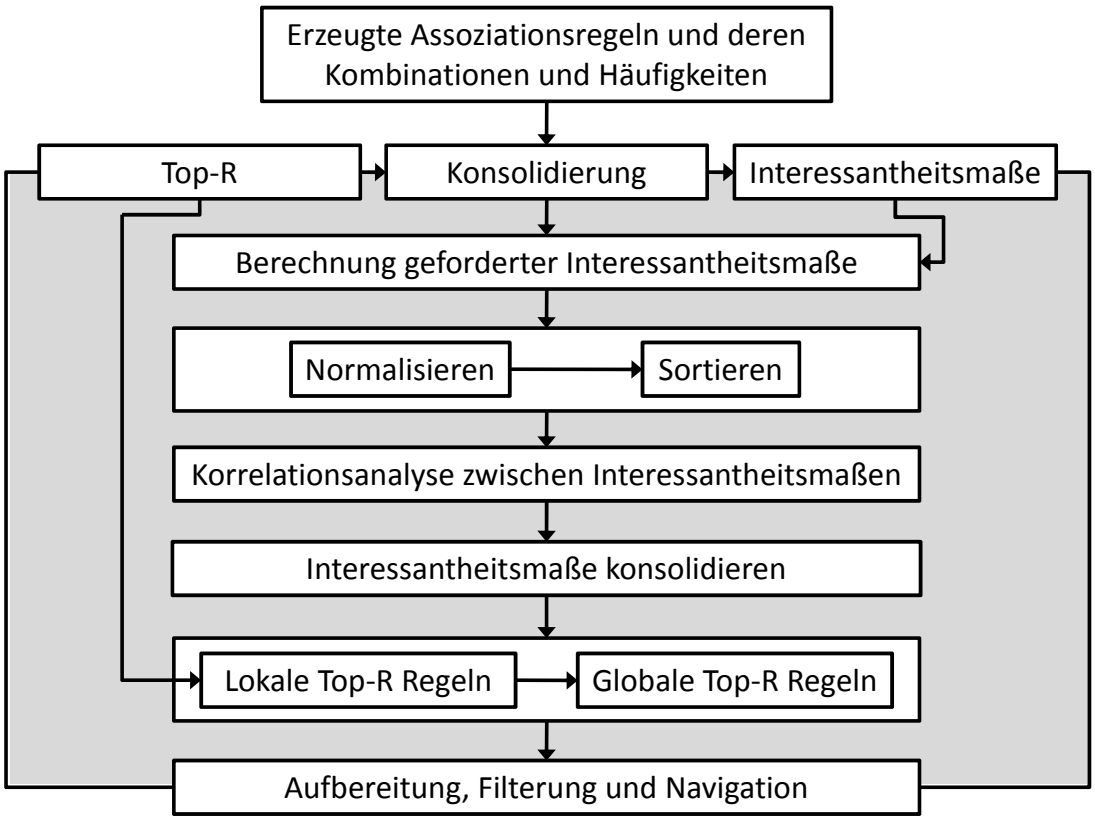


Abbildung 4.1.: Übersicht Regelbewertungsprozess

Abbildung 4.1 zeigt eine detaillierte Übersicht über die in dieser Arbeit verwendete Regelaufbereitung. Die einzelnen Schritte werden in den nachfolgenden Abschnitten betrachtet. Begonnen wird mit der Berechnung der Interessantheitsmaße für alle vorhandenen Regeln. Diese werden normalisiert und bilden sortiert eine Rangfolge pro Maß. Anschließend werden redundante Maße entfernt, um eine möglichst vielseitige Gesamtbewertung der Regeln zu erhalten. Jede Beurteilung der Regeln durch ein verbleibendes Maß wird im Folgenden als Blickwinkel auf die Regeln bezeichnet, wobei die Maße verschiedene Bewertungsmaßstäbe darstellen. Die konsolidierten Maße werden zu einer Gesamtrangfolge vereinigt. Diese Bewertung stellt dem Anwender eine Auswahl der Regeln zur Verfügung, welche aus mehreren Blickwinkeln interessant erscheinen. Damit wird die gezielte Navigation des Nutzers gemäß seinen Vorstellungen in der Regelmenge ermöglicht. Die folgenden Abschnitte erläutern das Verfahren zur Vereinigung von Interessantheitsmaßen zu einem allgemeinen Bewertungsmaß. Darauf basierend kann eine Auswahl potenziell wichtiger Regeln durchgeführt werden.

4.1. Vereinigung von Interessantheitsmaßen

Um die Überführung technischer Maßzahlen in eine allgemein verständliche Form zu ermöglichen, müssen betrachtete Maße in eine zueinander vergleichbare, verständliche Normalform gebracht werden. Damit wird sowohl das Erkennen einer Korrelation zwischen Interessantheitsmaßen, als auch eine Aggregation der verschiedenen Maße einer Regel zu einem Gesamtmaß ermöglicht. Die nötigen Aufbereitungsschritte sollen in den folgenden Abschnitten näher betrachtet werden.

4.1.1. Normalisierung von Interessantheitsmaßen

Als Annahme für die Vereinigung von Interessantheitsmaßen soll gelten, dass

1. alle Maße gleich gewichtet und damit gleichwertig sind, und

2. sich der Wert einer Regel gegenüber dem Wert anderer Regeln durch den Vergleich pro Interessantheitsmaß erkennen lässt.

Die erste Annahme dient lediglich einer Vereinfachung des Verfahrens. Sie kann verletzt werden, indem der Anwender die verfügbaren Interessantheitsmaße nach seinen Vorstellungen unterschiedlich gewichtet. Die Bevorzugung einzelner Maße kann sinnvoll sein, wenn der Nutzer sich über die Aussagekraft bestimmter Maße und deren korrekter Deutung bewusst ist. Ein solches Vorgehen benötigt jedoch einen zusätzlichen Eingriff auf den Regelaufbereitungsprozess. Dies erschwert die Parameterwahl und Reproduzierbarkeit des Ergebnisses und erfordert Anwender mit vertieftem Verständnis für die Interessantheitsmaße. Dies widerspricht den Anforderungen dieser Arbeit und soll somit nicht näher betrachtet werden.

Durch die zweite Annahme entspricht der Wert einer Regel für ein betrachtetes Interessantheitsmaß deren Rang. Statt dem eigentlichen Interessantheitsmaß wird die Platzierung der Regel im Vergleich zu dem restlichen Ergebnis verwendet. Diese Vereinfachung reduziert die Aussagekraft der exakten Interessantheitsmaße und verliert Informationen über den absoluten Wertunterschied der Regeln. Der Wertebereich wird hingegen für alle Interessantheitsmaße auf $1 \ldots |J|$ normiert und somit direkt maßeübergreifend vergleichbar. Wenn genügend Wissen über ein Maß vorhanden ist, kann die Platzierung für diese Regel auch gewichtet werden. Beispielsweise kann die wichtigste Regel auf den ersten Platz, die zweitwichtigste Regel auf den zehnten Platz, usw. gelegt werden. Zur Vereinfachung wird an dieser Stelle eine ungewichtete Bewertung angenommen. Die Integration in das Gesamtkonzept ist jedoch einfach möglich.

Die Platzierung einer Regel erfolgt unter Beachtung der Semantik des jeweiligen Interessantheitsmaßes. Die erste Platzierung erreicht die Regel mit der höchsten Interessantheit innerhalb des Maßes, was nicht zwangsläufig der Sortierung nach dem Zahlenwert des Maßes entsprechen muss. Vielmehr werden die nach Deutung eines Maßes interessantesten Regeln entsprechend gut platziert. Dabei kann je nach Suchziel beachtet werden, ob negative Korrelationen der Wichtigkeit vergleichbarer positiver Korrelationen entsprechen oder nicht. Der für jede Regel und jedes Interessantheitsmaß berechnete Wert wird durch die Vereinfachung auf die jeweilige Platzierung der Regel in der Regelmenge reduziert. Gleiche Werte für ein Maß zweier Regeln müssen hierbei jeweils auf gleiche Weise sortiert und platziert werden. Das erleichtert das Finden von Korrelationen zwischen zwei Maßen.

Einige Interessantheitsmaße ermitteln für bestimmte Regeln aufgrund semantischer Besonderheiten sehr hohe oder niedrige Bewertungen. Beispielsweise können die Interessantheitsmaße *Überzeugung* und *Odds Ratio* den Wert ∞ für eine Regel bestimmen. Somit scheinen diese Regeln aus Sicht der speziellen Interessantheitsmaße unendlich wichtiger als jede Regel mit endlichem Wert. Diese Besonderheiten entsprechen nicht den Erwartungen eines Anwenders, wenn dieser das Verhalten eines Interessantheitsmaßes nicht korrekt deutet oder deuten kann. Für den Fall erleichtert eine Darbietung als Platzierung einer Regel im Vergleich zur gesamten Regelmenge intuitiv und klar verständlich den Wert der Regel. Die wesentlichen Merkmale werden beibehalten und erlauben trotz Verlust an Aussagekraft des Interessantheitsmaßes eine Einschätzung von deren relativen Wichtigkeit innerhalb der betrachteten Regelmenge.

Daher wird vorgeschlagen, dass für jedes genutzte Interessantheitsmaß dessen absoluter Zahlenwerte durch die Platzierung der entsprechenden Regel in Relation zu allen weiteren Regeln ersetzt wird. Derartig normalisierte Interessantheitsmaße sind zueinander leicht vergleichbar, da sie die gleichen Wertebereiche aufweisen. Außerdem können sie auf weitere Interessantheitsmaße erweitert werden und vermeiden eine Überbewertung einzelner Regeln durch besonders gute Bewertungen in einzelnen Maßen. Die Regelplatzierung kann zusätzlich durch den Ausgangswert des Interessantheitsmaßes ergänzt werden. Für einen Anwender ist dies oft nur sinnvoll, wenn die Regelmenge gering ist und die Maße durch den Anwender korrekt gedeutet werden können. Zur groben Filterung sind die Platzierungen in vielen Fällen ausreichend.

Die Abschwächung besonders guter Bewertungen von Regeln durch einzelne Interessantheitsmaße und Blickwinkel auf die Regelmenge kann auch als Nachteil betrachtet werden. Wenn beispielsweise die Regel Y durch eine überdurchschnittliche Häufigkeit aus J hervorsticht, verliert Y bei durchschnittlichen Werten in anderen Interessantheitsmaße insgesamt an Bedeutung. Wenn dies vermieden werden muss, sind auch hier Gewichtungen möglich, um einzelne Spitzenplatzierungen im Gesamtmaß aufzuwerten. Die folgenden Abschnitte befassen sich mit der Aggregation der Maße zu einem Gesamtmaß und mit der dafür nötigen Konsolidierung der Interessantheitsmaße.

4.1.2. Reduktion von Interessantheitsmaßen

Verschiedene Interessantheitsmaße können aus verschiedenen Blickwinkeln sehr ähnliche Bewertungen von Regeln aufzeigen [TKS02]. Um jedoch eine möglichst vielseitige Gesamtsicht zu erhalten, sollen die zueinander ähnlichen Sichtweisen lediglich einfach gewertet werden. Dies vermeidet eine Abhängigkeit der globalen Bewertung von der nutzerdefinierten Auswahl an Interessantheitsmaßen. Die Auswirkungen einer möglicherweise ungünstige Wahl von Maßen werden verringert. Dabei stellt sich das Problem, dass Korrelationen zwischen Interessantheitsmaßen nicht immer für alle Regelmengen und Interessantheitsmaße gleich sind. Dabei kann beispielsweise eine Menge mit ausschließlich sehr häufigen Regeln keine Verbindung zwischen Regelhäufigkeit und Konfidenz aufweisen, während seltene Regeln zu hohen Konfidenzwerten neigen. Außerdem kann die Häufigkeit der Bedingung einer Regel in manchen Fällen mit der Gesamthäufigkeit der Regel korrelieren, was jedoch nicht zwingend für jede Regelmenge gilt. Entsprechend muss eine Korrelationsanalyse zwischen den berechneten Interessantheitsmaßen pro zu bewertender Regelmenge erfolgen.

Um Interessantheitsmaße möglichst neutral zu einem Gesamtmaß zu vereinigen, müssen die Korrelationen der einzelnen Maße zueinander bestimmt und entfernt werden. Indem voneinander abhängige Maße lediglich durch einen Repräsentanten in die endgültige Bewertung eingehen, kann die Anzahl der für eine Gesamtbewertung relevanten Einflussgrößen reduziert werden.

Damit ein Vergleich verschiedener Interessantheitsmaße möglich ist, werden für alle Regeln und betrachteten Maße die entsprechenden Wertausprägungen berechnet. Sortiert ergeben sich daraus pro Interessantheitsmaß die Platzierungen für jede Regel. Für alle Paarungen der Platzierungsvektoren zweier Interessantheitsmaße v und w kann ein Korrelationsmaß, wie in der folgenden Gleichung dargestellt, berechnet werden. Andere Maße zur Feststellung einer Korrelation sind hierfür ebenfalls möglich, aber als Beispiel soll an dieser Stelle der Pearson-Koeffizienten als Korrelationsmaß dienen (vgl. Spearmans Rho [Spe04]). Durch diesen Wert kann die Stärke einer Abhängigkeit $c_{v,w}$ zwischen den Maßpaaren v und w definiert werden. Besteht eine hohe Korrelation, können diese Werte jeweils durch ihren Partner repräsentiert werden und das zweite, redundante Maß entfällt. Zur Berechnung des Pearson-Koeffizienten wird die Gleichung

$$c_{v,w} = \frac{\frac{1}{m}\sum_{i=1}^{m}(v_i - \overline{v}) \cdot (w_i - \overline{w})}{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(v_i - \overline{v})^2} \cdot \sqrt{\frac{1}{m}\sum_{i=1}^{m}(w_i - \overline{w})^2}}$$

verwendet, wobei $\overline{v} = \frac{1}{m}\sum_{i=1}^{m} v_i$ und $\overline{w} = \frac{1}{m}\sum_{i=1}^{m} w_i$ die Durchschnitt von v und w darstellen. Die Kardinalität der Regelmenge wird durch $m = |J|$ beschrieben. Ab welchem Pearson-Koeffizienten zwei Interessantheitsmaße als korreliert gelten, ist von den

	SynthB	**Retail**	**SynthC**	**Connect-4**	**Gazelle**	**Mushroom**
Maße insg.	20	20	20	20	20	20
Korrelationen	14	17	15	17	13	15
Gruppen	3	3	3	5	3	3
Verbleibende Maße	9	6	8	8	10	8

Tabelle 4.1.: Maß-Korrelationen einiger Beispieldaten, Pearson-Koeff.≥0,8

Anforderungen des Anwenders abhängig, wobei zwei Vektoren linear umso abhängiger voneinander sind, je näher der Pearson-Koeffizient an 1 ist. Entsprechend gilt bei einem Pearson-Koeffizient von 0 Unabhängigkeit zwischen den Vektoren.

Der Berechnungsaufwand dieser Abhängigkeitsbestimmung steigt linear mit der Anzahl der betrachteten Regeln. Für eine große Anzahl zu analysierender Regeln kann es daher sinnvoll sein, durch die Wahl einer zufälligen Stichprobe eine repräsentative Regelmenge auszuwählen. Erfolgt die Berechnung der Korrelationen nur auf dieser Teilmenge, kann die Laufzeit der Abhängigkeitsermittlung reduziert werden. Dabei werden übereinstimmende Maße mit hoher Wahrscheinlichkeit weiterhin gefunden, wenn die Stichprobe groß genug ist, um die vollständige Ergebnismenge sinnvoll zu repräsentieren. Entstehende Fehler erzeugen dabei meist nur gering veränderte Gesamtbewertungen von Regeln, da lediglich ein eigentlich nicht korreliertes Maß entfernt wird oder ein Blickwinkel mehrfach gewertet wird. Sind die verbleibenden Sichten für den Anwender ausreichend, werden dessen Chancen zum Auffinden relevanter Informationen zwar verringert, bleiben jedoch im Wesentlichen erhalten.

Tabelle 4.1 verdeutlicht anhand von einigen Beispieldaten die Ergebnisse einer Korrelationsanalyse zwischen Interessantheitsmaßen. Begonnen wird in der ersten Zeile mit 20 verschiedenen Interessantheitsmaßen. Der Schwellwert, ab welchem zwei Interessantheitsmaße als korreliert gelten sollen, wird in dem Beispiel auf einen Pearson-Koeffizienten $0,8$ gesetzt. Die zweite Zeile zeigt die Gesamtzahl der so korrelierten Maße, welche durch die Anzahl der entstehenden Gruppen in der dritten Zeile repräsentiert werden können. Durch die Wahl eines Repräsentanten pro Gruppe und den jeweils unabhängigen Maßen ergibt sich in der vierten Zeile die Menge der verbleibenden Interessantheitsmaße. Nur diese Teilmenge bildet im folgenden Abschnitt die globale Bewertung einer Regel. Die Menge der Interessantheitsmaße kann in dem Beispiel durch eine Korrelationsanalyse jeweils mindestens halbiert werden. Die Anzahl der verbleibenden Sichten auf die Regelmenge wird reduziert. Es wird bereits ersichtlich, dass zwar die Anzahl der entstehenden Gruppen jeweils ähnlich ist, diese jedoch für jeden Datenbestand unterschiedlich sind. Basierend auf diesen reduzierten Maßen kann im Folgenden eine Vereinigung der einzelnen Bewertungen und eine Begrenzung auf die wahrscheinlich interessantesten R Regeln erfolgen.

4.1.3. Bildung eines allgemeinen Interessantheitsmaßes

Ist der Regelauswerteprozess gemäß Abschnitt 4.1.1 und 4.1.2 durchlaufen, dient die daraus entstehende repräsentative und normierte Auswahl von Interessantheitsmaßen als Grundlage für den folgenden Abschnitt. Hier soll untersucht werden, wie sich eine Menge von Interessantheitsmaßen zu einer Bewertung vereinigen lassen und damit eine globale Sortierung erlauben. Grundlegend ist hierbei die Wahl des effizientesten Interessantheitsmaßes, beispielsweise mit Hilfe der Data-Envelopment-Anlayse [CCR78], möglich. Um einerseits die Berechnung einfach zu halten und andererseits auch negative Bewertungen von Regeln zu erhalten, wird dieser Ansatz an dieser Stelle nicht untersucht. Stattdessen werden zur Vereinigung der Interessantheitsmaße zwei Strategien vorgeschlagen.

Die erste Variante summiert alle Maßplatzierungen einer Regel. Durch Sortierung der summierten Platzierung ergeben sich die nach der globalen Interessantheit geordneten Regeln. Als Annahme gilt hierbei, dass alle Interessantheitsmaße für eine Gesamtbewertung der Regel gleich wichtig sind und alle Platzierungen gleich gewichtet werden. Sind die sortierten Regeln auf die ersten R Einträge reduziert, stellt die verbleibende Menge die Top-R-Regeln gemäß der Addition der vorhandenen Interessantheitsmaße dar.

Erreicht beispielsweise eine Regel bei Konfidenz den dritten, bei Lift den achten Rang und bei Überzeugung den zehnten Rang, ergibt sich eine Gesamtbewertung von $3+8+10 = 21$. Dies wird für alle Regeln durchgeführt. Die aufsteigende Sortierung aller Regeln nach dieser Gesamtbewertung ergibt die globale Rangfolge.

Diese Art der Vereinigung vermeidet das übermäßig positive Bewerten einer Regel, wenn diese nur in einigen Interessantheitsmaßen gut platziert ist. Regeln mit konsequent positiven, jedoch selten überproportional guten Bewertungen werden aufgewertet. Eine Regel wird also nicht dadurch interessant, dass sie aus einem Blickwinkel sehr wichtig erscheint. Vielmehr muss deren Interessantheit mit einer guten Bewertung durch weitere Maße unterstützt werden. Eine aggregierte Platzierung besitzt bei i Interessantheitsmaßen einen Wertebereich von $i \ldots (i \cdot R)$. Wenn dieser Wert dem Nutzer präsentiert wird, kann eine Darstellung als globale Platzierung (Rangfolge der summierten Platzierungen) eine Auswertung erleichtern. Der Wert einer Regel durch einzelne gute Platzierungen wird bis zu einem bestimmten Grad erhalten. Solche Regeln finden weiterhin Beachtung, können jedoch von Regeln mit konsequent mittelmäßigen Platzierungen verdrängt werden. Die Gesamtbewertung zeigt Regeln mit mehrfachen guten Bewertungen und vermeidet die Überbewertung einzelner Bestplatzierungen. Da mehrfach als interessant erachtete Regeln eine gute globale Platzierung erreichen, sind diese wahrscheinlich auch aus subjektiver Sicht des Anwenders interessant. Die Ordnung kann somit als Vorschlag für die den Anwender interessierenden Regeln gelten.

Für die einzelnen Ränge sind Gewichtungen möglich, welche sehr gute Platzierungen aufwerten. Beispielsweise können Regeln für sehr gute Platzierungen in einem Maß be-

vorteilt werden. Hierbei entsteht jedoch das Problem, dass der Grad der Aufwertung vom Anwender festgelegt werden muss. Zusätzlich sind verschiedene Gewichtungsfunktionen pro Interessantheitsmaß möglich. Das widerspricht dem Ziel dieser Arbeit, dass auf nutzerdefinierte Parameter verzichtet werden soll, um die Anwendung der Verfahren zu erleichtern. Daher soll eine ungewichtete Addition der einzelnen Platzierungen verwendet werden.

Als zweiter Ansatz kann eine Aggregationsstrategie verwendet werden, welche die speziellen Eigenschaften der einzelnen Interessantheitsmaße mehr beachtet. Hierfür werden von dem Platzierungsvektor jedes Interessantheitsmaßes jeweils die besten R Regeln als globale Top-R-Kandidaten ausgewählt. Somit entstehen insgesamt maximal $i \cdot R$ globale Top-R-Kandidaten. Für diese wird die globale Platzierung wie in der ersten Variante bestimmt und damit die globalen Top-R-Kandidaten ermittelt. Eine Regel, die nicht in mindestens einem Interessantheitsmaß unter den besten R Regeln vertreten war, wird von der globalen Bewertung ausgeschlossen. Sie muss also in mindestens einem Maß gut platziert sein, um sich als globaler Top-R-Kandidat zu qualifizieren.

Für alle Top-R-Kandidaten wird äquivalent zur ersten Variante die globale Platzierung gebildet. Dabei werden für alle Kandidaten deren Platzierung in den relevanten Interessantheitsmaßen aufsummiert. Die Sortierung dieser globalen Kandidaten nach dem Gesamtmaß ermittelt eine Rangfolge der Kandidaten. Wird diese auf die besten R Regeln begrenzt, kann sie als Vorauswahl interessanter Regeln für den Nutzer betrachtet werden. Dieses Verfahren erweitert die erste Variante um die Bedingung, dass eine globale Top-R-Regel mindestens einmalig auch ein lokales Top-R-Ergebnis erreicht haben muss. Eine globale Top-R-Regel ist somit mindestens aus Sicht eines Interessantheitsmaßes interessant.

Abbildung 4.2 zeigt beispielhaft eine solche Top-R-Berechnung der zweiten Variante mit $R = 5$. Betrachtet werden die drei Interessantheitsmaße Häufigkeit, Lift und Überzeugung. Die Liste der $3x5$ Ergebniskandidaten umfasst konsolidiert sieben Top-5-Kandidaten. Für diese Kandidaten wird die globale Platzierung über alle Interessantheitsmaße gebildet. Dargestellt wird das durch die Pfeile. Hierbei werden auch Platzierungen außerhalb der lokalen Top-5-Regeln eines Maßes betrachtet. Nachdem diese Liste gemäß der globalen Platzierung sortiert ist, bilden die ersten fünf Regeln die globalen Top-5-Assoziationsregeln. Im Falle von $R10$ ergeben die lokalen Platzierungen $Support = 1$, $Lift = 5$ und $Conviction = 9$ die Gesamtwertung 15 und damit insgesamt den vierten Rang.

Wie alle bereits vorgestellten Interessantheitsmaße entziehen sich beide Variationen des globalen Maßes einer allgemeinen Gütebewertung. Der für einen Nutzer subjektive Wert einer Regel kann nicht objektiv erfasst werden, stellt jedoch das bestmöglich zu erreichende Ergebnis dar. Der Vergleich zwischen der Qualität, Aussagekraft und Güte von verschiedenen Interessantheitsmaßen ist somit nicht sinnvoll möglich. Als weitere Möglichkeit zur Vereinigung mehrere Platzierungen der Regeln kann auch die Wurzel der

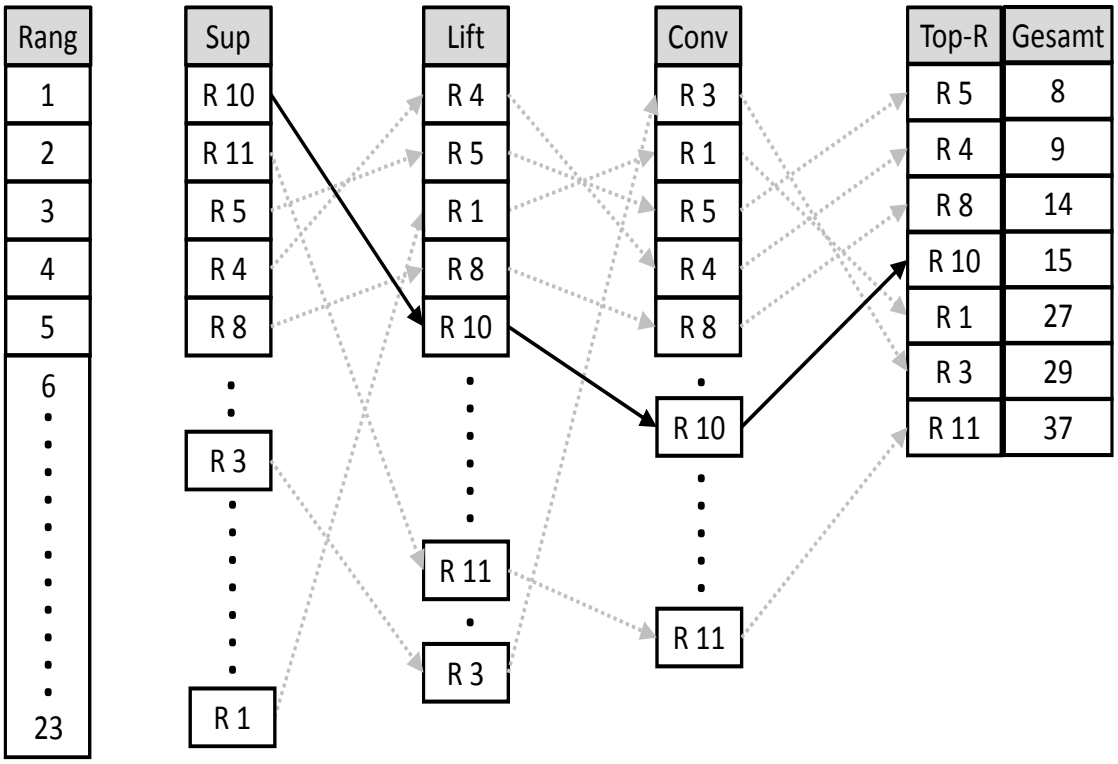


Abbildung 4.2.: Beispiel für eine Top-R-Sortierung mit drei Interessantheitsmaßen

summierten Quadrate der Ränge (der Betrag des Platzierungsvektors einer Regel) genutzt werden. Wenn schlechte Platzierungen den Wert einer Regel stärker senken sollen als bei einer Summierung, kann die Vereinigung der Ränge nach dem Betrag des Platzierungsvektors genutzt werden. In dieser Arbeit wird das nicht näher betrachtet.

Die in diesem Abschnitt erläuterten Verfahren zur Bildung eines allgemeinen Interessantheitsmaßes sollen somit lediglich als Alternative zur Verwendung einzelner Interessantheitsmaße betrachtet werden, welche die Bewertungsmaßstäbe verschiedener Einzelmaße zu einer Rangfolge vereinigen. Während ein fachkundiger Anwender mit einer gezielten Wahl eines oder weniger Maße die aus seiner Sicht interessantesten Regeln finden kann, stellt eine allgemeine Wertung einen Einstiegspunkt zur Regelauswertung ohne besondere Fachkenntnisse dar. Im folgenden Abschnitt wird untersucht, wie alternativ zu diesen Rangfolgen die effiziente Navigation in einer großen Regelmenge realisiert werden kann.

4.2. Gruppierung von Assoziationsregeln

Nachdem im vorangegangenen Abschnitt die Vereinigung von Regelmaßen zu einer globalen Bewertung vorgestellt wurde, wird im folgenden Abschnitt untersucht, wie eine Vielzahl von Regeln auf eine repräsentative Regelmenge reduziert werden kann. Dabei werden aus jeweils zueinander ähnlichen Regeln Regelgruppen gebildet, welche dem Nutzer eine effiziente Navigation bei der Analyse der Daten erlauben. Hierfür wird als Maß der Ähnlichkeit zwischen zwei Regeln deren Distanz zueinander ermittelt. Die folgenden Teilabschnitte erläutern zu Beginn den Begriff der *Nachbarschaft* und stellen anschließend syntaktische und semantische Gruppierungsmöglichkeiten vor.

4.2.1. Nachbarschaften von Assoziationsregeln

Als *r-Nachbarschaft* einer Regel Y_1 wird die Umgebung dieser Regel bezeichnet, welche durch den Radius r begrenzt wird. Alle Regeln Y_i, deren Entfernung gemäß eines Distanzmaßes $Distanz(Y_1, Y_i) \leq r$ ist, sind Nachbarn von Y_1. Der Bereich r um Y_1 wird entsprechend als Nachbarschaft von Y_1 definiert. Dabei kann durch die Wahl von spezifischen Gewichtungsparametern ρ der Wert einzelner Bestandteilen der zu untersuchenden Regel verändert werden.

Die Nachbarschaft $Nachbar$ der Regel Y_1 innerhalb der Regelmenge J und der Grenze r wird definiert als

$$Nachbar(r, J, Y_1) = \{Y_i | Distanz(Y_1, Y_i) \leq r, Y_i \in J\}.$$

Eine Möglichkeit zur Bestimmung der Interessantheit ist die Anzahl der Nachbarregeln von Y_1. Eine Regel Y_1 wird dabei als isoliert bezeichnet, wenn sich keine oder nur wenige Regeln innerhalb von r befinden. Dies kann sowohl besonders wichtig als auch besonders unwichtig für einen Nutzer sein. Die Interessantheit einer solchen Regel unterliegt somit dem subjektiven Empfinden des Anwenders.

Des Weiteren können r-Nachbarschaften zur grafischen Aufbereitung gefundener Assoziationsregeln genutzt werden. Durch gemäß ihrer Nachbarschaften vereinigter Regeln kann die Komplexität einer grafischen Darstellung vereinfacht werden, was die optische Erkennung und Auswahl zueinander ähnlicher Regeln bzw. von Regeln mit ähnlichen Eigenschaften erlaubt.

4.2.2. Syntaktische Gruppierung

Anders als bei der im Abschnitt 4.1.3 vorgestellten Vereinfachung zur Bestimmung von Top-R-Regeln besteht der Ansatz der Gruppierung in einer verbesserten Informationsex-

traktion durch ein gezieltes Fokussieren und Navigieren in der gesamten Regelmenge. Ein automatisches Verwerfen wahrscheinlich uninteressanter Regeln erfolgt explizit nicht. Unwichtige Informationen entfallen stattdessen durch eine Abstraktion von einzelnen Regeln auf Regelgruppen. Dieser Abschnitt erläutert eine Gruppierung nach dem Inhalt einer Regel.

Die syntaktische Distanz, vorgestellt von Dong und Li [DL98], bewertet den Abstand zweier Regeln im Bezug auf deren Struktur und Beschaffenheit. Semantische Eigenschaften, wie die Häufigkeiten von Regelteilen, werden nicht beachtet. Die Bedingung A, Schlussfolgerung C, sowie deren Vereinigung $(A \cup C)$ der Regel Y_1 werden auf ihre Distanz gegenüber einer Regel Y_2 untersucht. Die syntaktische Distanz vergrößert sich hierbei mit der Anzahl der unterschiedlichen Elemente zwischen den Teilen der untersuchten Regeln. Die Menge der nicht in beiden Kombinationen gemeinsam vorgefundenen Elemente wird als symmetrische Differenz $\triangle$ dieser Kombinationen bezeichnet. Mit Hilfe der nachfolgenden Gleichung zur Berechnung der syntaktischen Distanz $Distanz_{syn}$ können durch die Wahl von Gewichtungsparametern $\rho_{b,s,g}$ die einzelnen Bestandteile einer Regel verstärkt oder geschwächt werden.

$$Distanz_{syn}(Y_1, Y_2) = \rho_b \cdot |(A_1 \cup C_1) \triangle (A_2 \cup C_2)| + \rho_s \cdot |A_1 \triangle A_2| + \rho_g \cdot |C_1 \triangle C_2|$$

Durch $\rho_s = \rho_g = 0$, $\rho_b = 1$ wird die syntaktische Distanz zwischen den Regeln $Y_1 : ABC \Rightarrow D$ und $Y_2 : D \Rightarrow ABC$ berechnet und ergibt $Distanz_{syn}(Y_1, Y_2) = 0$. Die Distanz zwischen Y_1 und Y_2 ist bei dieser Gewichtung minimal, da beide Regeln auf derselben Ausgangskombination $ABCD$ basieren.

Je nach Wahl der Gewichtungsparameter $\rho_{b,s,g}$ kann bestimmt werden, ob gemeinsame Elemente in den Bedingungen, Schlussfolgerungen oder in der gesamten Regel eine hohe Ähnlichkeit der Regeln bedeuten. Diese Abwägungen sind von den Erwartungen des Anwenders abhängig und können nur bedingt automatisch erfolgen. Wie im vorherigen Abschnitt kann der Erfolg dieser Gruppierung lediglich subjektiv durch den Nutzer bewertet werden. Eine objektive Bewertung ist nur unzureichend und nicht allgemeingültig möglich.

Die Bestimmung der syntaktischen Nachbarschaft weist bei m Regeln eine Komplexität $\mathcal{O}((m^2 - m)/2)$ auf. In [DL98] wird hierfür eine Optimierung vorgeschlagen, welche als Gewichtungsparameter die Werte $\rho_b = 1, \rho_s = |E'| - 1/|E'|^2$ und $\rho_g = 1/|E'|^2$ voraussetzt. E' entspricht hierbei der Menge der in J auftretenden Elemente.

Durch Nutzung einer Datenstruktur mit den Kombinationen als Knoten und Verknüpfungen entsprechend den Teilmengenbeziehungen zwischen Kombinationen können Nachbarschaften effizient ermittelt werden. Dazu wird zu jedem Blatt die Menge aller Regeln der entsprechenden Kombination vorgehalten. Um ausgehend von dieser jeweiligen 1-Nachbarschaft die Umgebung mit einem Radius $r \geq 1$ zu ermitteln, erfolgt die Suche lediglich einmalig pro Blattknoten und bedient jeweils alle daran gebundene Regeln. Die Komplexität der Nachbarschaftsberechnung ändert sich auf $\mathcal{O}((|E'|^2 - |E'|)/2)$. Für den

Fall das $m < |E'|$ kann die Berechnung somit beschleunigt werden. Bei der von Dong und Li vorgeschlagenen Optimierung werden Regeln aus der gleichen Kombination immer mit einem geringeren Abstand bewertet als Regeln aus verschiedenen Kombinationen. Dies verhindert eine Gruppierung von Regeln mit sehr verschiedenem Inhalt. Es bewirkt auch, dass zwei Regeln mit den gleichen verwendeten Elementen und verschiedener Aufteilung auf Bedingung und Schlussfolgerung als zueinander ähnlich betrachtet werden. Diese Eigenschaft ist nicht allgemeingültig und kann somit zu unerwünschten Gruppierungen führen.

Sind die jeweiligen syntaktischen Distanzen zwischen den einzelnen Regeln einer Regelmenge nach den bereits erläuterten Verfahren ermittelt, kann über diese Abstandsdefinition eine Reduktion der Regelmenge auf eine bestimmte Anzahl diese Menge repräsentierende Regelgruppen erfolgen. Mögliche Algorithmen zur Bestimmung der Gruppen finden sich im Bereich der Clusteranalysen, beispielsweise dem K-Means-Cluster-Algorithmus [Mac94].

Ein Beispiel für die Gruppierung nach der syntaktischen Distanz und der r-Nachbarschaft wird in der Tabelle 4.3 gezeigt. Die in der Tabelle 4.2 beispielhaft dargestellten Top-12 Regeln werden zu insgesamt fünf Gruppen zusammengefasst. Die Parameter für die Bestimmung der r-Nachbarschaften sind $r = 2$, $\rho_b = 1$, $\rho_s = |E'| - 1/|E'|^2$ und $\rho_g = 1/|E'|^2$ mit $|E'| = |\{A, B, C, D, E, F\}| = 6$.

Rang	**Regel**	**Rang**	**Regeln**	**Rang**	**Regeln**
1	$ABC \Rightarrow D$	5	$A \Rightarrow BCD$	9	$EB \Rightarrow AC$
2	$D \Rightarrow E$	6	$AB \Rightarrow D$	10	$FA \Rightarrow B$
3	$AC \Rightarrow E$	7	$DE \Rightarrow A$	11	$ABCE \Rightarrow D$
4	$B \Rightarrow F$	8	$E \Rightarrow DB$	12	$E \Rightarrow F$

Tabelle 4.2.: Beispiel: Ausgangsregeln Top-12

Gruppe	**Repräsentant**	**Regeln**	**Distanz zum Repräsentant**
G1	$ABC \Rightarrow D$	$A \Rightarrow BCD$	0,33
		$AB \Rightarrow D$	1,38
		$A \Rightarrow BCDE$	1,361
G2	$D \Rightarrow E$	$DE \Rightarrow A$	1,194
		$E \Rightarrow DB$	1,361
G3	$AC \Rightarrow E$	$EB \Rightarrow AC$	1,638
G4	$B \Rightarrow F$	$FA \Rightarrow B$	1,472
G5	$E \Rightarrow F$		

Tabelle 4.3.: Beispiel: syntaktische Gruppierung nach r-Nachbarschaft

In Tabelle 4.3 wird eine Gruppennummer, ein Repräsentant der Gruppe, die der Gruppe zugeordneten Regeln und deren Distanz zu dem jeweiligen Gruppenrepräsentanten dargestellt. Gruppenrepräsentant ist die bestplatzierte Regel einer Gruppe. Die Distanz wird mittels $Distanz_{syn}$ berechnet. Die zwölf Ausgangsregeln können in dem Beispiel auf fünf Gruppen aufgeteilt werden, deren syntaktische Interessantheit durch gleiche enthaltene Elemente für den Anwender ähnlich ist. Somit muss lediglich der Wert von fünf Regelgruppen durch den Nutzer bestimmt werden, statt für jede einzelne der ursprünglichen zwölf Regeln. Der Aufwand wird insgesamt mehr als halbiert. Außerdem kann die Übersicht über die Ergebnismenge vereinfacht werden. Die Eigenschaften der Regeln, wie z.B. deren Konfidenz, werden jedoch nicht betrachtet.

Ein Nachteil der syntaktischen Gruppierung ist außerdem, dass die Möglichkeiten begrenzt sind. Der Abstand durch unterschiedliche Elemente zwischen den Teilen zweier Regeln ist stets ganzzahlig. Die Faktoren ρ sind für die gesamte untersuchte Regelmenge konstant. Somit ergeben sich vor allem bei kurzen Regeln nur vergleichsweise wenige mögliche Distanzen. Die Gruppierungfähigkeit der Regelmenge durch veränderten Radius um den Repräsentanten ist somit nur bedingt kontinuierlich möglich, wodurch die Regel(gruppen)menge nicht gezielt auf eine bestimmte Größe reduziert werden kann. Ein Ansatz, dies zu erreichen, bietet die semantische Gruppierung.

4.2.3. Semantische Gruppierung

Während syntaktische Analysen die Beschaffenheit und die Elemente von Regeln zur Bestimmung der Ähnlichkeit nutzen, verwenden semantische Analysen Übereinstimmungen in semantischen Eigenschaften. Als Semantik werden hierbei Interessantheitsmaße genutzt und mit dem Wert der Regel gleichgesetzt.

Die durch Toivonen [TKR+95] eingeführte und später von Gupta [GSG99] erweiterte semantische Distanz beschreibt den Abstand zweier Regeln $Y_1 : A_1 \Rightarrow C_1$ und $Y_2 : A_2 \Rightarrow C_2$. Sie betrachtet allein die Anzahl gemeinsamer Transaktionen von Y_1 und Y_2 bzw. deren Regelteilen. Dabei sind zwei Regeln umso weiter voneinander entfernt, in je weniger Transaktionen sie gemeinsam auftreten. Die Bestimmung der semantischen Distanz $Distanz_{sem}$ zwischen Y_1 und Y_2 erfolgt durch

$$\begin{aligned} Distanz_{sem}(Y_1, Y_2) \quad = \quad & supportAbs(A_1 \cup C_1) + supportAbs(A_2 \cup C_2) \\ & -2 \cdot supportAbs(A_1 \cup C_1 \cup A_2 \cup C_2) \end{aligned}$$

Zwei Regeln sind also ähnlich in je mehr Transaktionen sie gemeinsam vorkommen. Kommen sie ausschließlich in den gleichen Transaktionen vor, gilt $supportAbs(A_1 \cup C_1) = supportAbs(A_2 \cup C_2) = supportAbs(A_1 \cup C_1 \cup A_2 \cup C_2)$. Als Distanz wird folglich 0 ermittelt. Gemäß der Downward-Closure-Eigenschaft stellt dies auch den minimal möglichen Abstand dar, da $(A_1 \cup C_1 \cup A_2 \cup C_2)$ nicht häufiger sein kann als seine Telmengen $(A_1 \cup C_1)$ und $(A_2 \cup C_2)$.

Während die Häufigkeiten von $(A_1 \cup C_1)$ und $(A_2 \cup C_2)$ in einem regulären Assoziationsregelsuchprozess ermittelt werden und zur Berechnung von Interessantheitsmaßen vorhanden sind, ist die Häufigkeit von $(A_1 \cup C_1 \cup A_2 \cup C_2)$ nicht zwangsläufig ebenfalls verfügbar. Diese Kombination ist aufgrund ihrer größeren Länge oftmals signifikant seltener als $(A_1 \cup C_1)$ oder $(A_2 \cup C_2)$. Daher erreicht sie Kombination oft nicht die geforderte Mindesthäufigkeit, während die restlichen benötigten Kombinationen diese überschreiten und Regeln bilden können. Wird die Häufigkeit von $(A_1 \cup C_1 \cup A_2 \cup C_2)$ nicht im Rahmen der regulären EHK bestimmt, muss sie gezielt nachträglich aus den untersuchten Daten ermittelt werden. Der dabei entstehende Berechnungsaufwand kann sehr groß werden.

Zudem ergibt sich bei der semantischen Gruppierung nach Toivonen das Problem, dass gruppierte Regeln jeweils sehr unterschiedliche Ausprägungen in Interessantheitsmaßen aufweisen können. Die Häufigkeiten der einzelnen Regelteile werden bei der Gruppierung nicht beachtet. Beispielsweise können die Regeln $Bier \Rightarrow Milch$ und $Windeln \Rightarrow Milch$ als ähnlich erkannt werden, da die Kombinationen $(Bier, Milch)$, $(Windeln, Milch)$ und $(Bier, Windeln, Milch)$ je gleich häufig auftreten. Während jedoch $Windeln$ und $Milch$ stark korreliert sind und einen Konfidenz von 100% aufweisen, können $Bier$ und $Milch$ nahezu unkorreliert sein, da $Bier$ auch häufig ohne Milch verkauft wird. Diese Gruppierung kann somit die Auswertung der Regelgruppen erschweren.

Im Rahmen dieser Arbeit wird eine Möglichkeit vorgeschlagen, eine semantische Gruppierung nach einer Menge von Interessantheitsmaßen durchzuführen. Dabei werden Ähnlichkeiten zwischen den Kontingenzmatrizen betrachteter Regeln herangezogen.

Während sich semantische Gruppierungen oft nur einer Auswahl von Interessantheitsmaße bedienen, können die Regelbewertungen durch den Nutzer weitere Maße erfordern. Ohne Wissen über die genaue Vorgehensweise und Präferenz von dessen Regelfilterung kann die semantische Gruppierung nach einem aus Sicht des Nutzers ungünstigem Interessantheitsmaß kontraproduktiv sein. Erfolgt beispielsweise eine Gruppierung nach ähnlichen Konfidenzwerten und der Nutzer möchte Regeln mit hoher Konfidenz und großer Häufigkeit als Ergebnis, wird ihm das Erkennen der gewünschten Regeln erschwert.

Um dieses Problem zu lösen, wird in dieser Arbeit eine Gruppierung von Assoziationsregeln nach ihren jeweiligen Kontingenzmatrizen vorgeschlagen. Diese aus Abschnitt 2.3.1 bekannte Datenstruktur wird zur Berechnung einer Vielzahl objektiver Maße benutzt. Da die sich ergebenden Interessantheitsmaße einzig aus der entsprechenden Kontingenzmatrix der Regel ergeben, entsteht eine für viele Maße starke Korrelation zwischen der Matrix und den daraus berechneten Maßen.

Somit wird im Folgenden angenommen, dass aus Matrizen mit ähnlichen Elementen auch ähnliche Interessantheitsmaße gebildet werden. Wird eine Regelmenge nach den Kontingenzmatrizen der Regeln in Gruppen aufgeteilt, bilden diese durch gleichartige Werte in den Matrizen vergleichbare objektive Interessantheitsmaße. Der Grad der Ähnlichkeit für einzelne Maße ist pro Gruppe abhängig von der jeweiligen Berechnungsvorschrift des

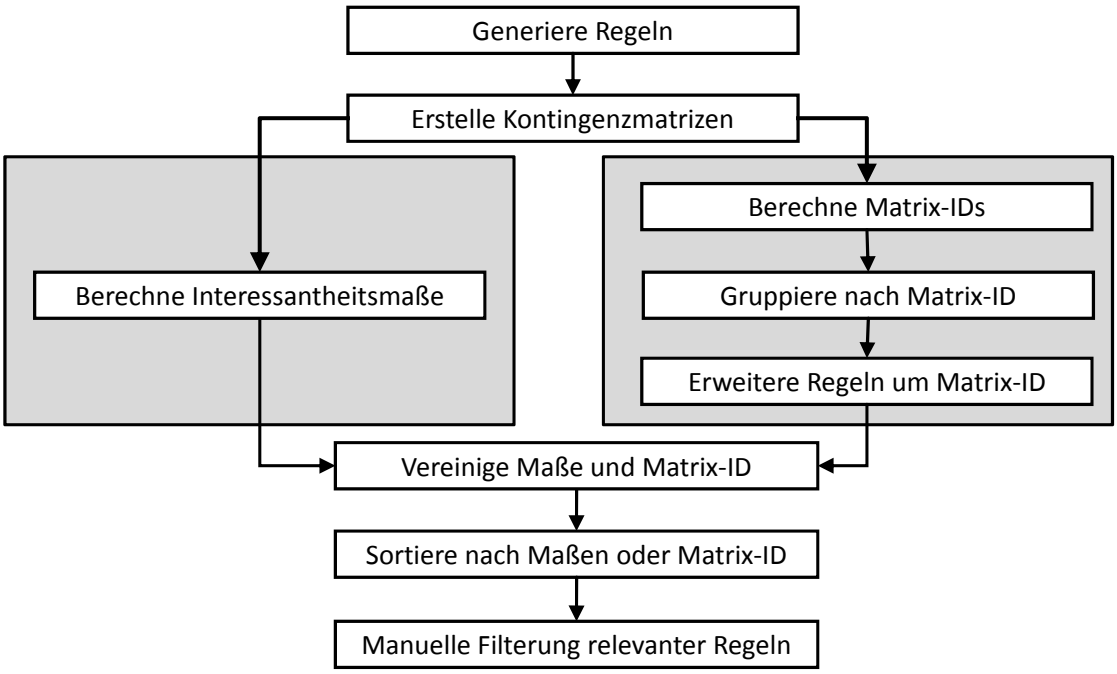


Abbildung 4.3.: Ablauf Regelgruppierung nach Kontingenzmatrix

Interessantheitsmaßes. Außerdem müssen Regeln mit ähnlichen Werten für ein bestimmtes Maß nicht zwangsläufig der gleichen Gruppe angehören. Die betrachtete Regelmenge kann jedoch mit dieser Maßnahme unabhängig von einzelnen Kriterien auf eine Menge von Regelgruppen reduziert werden.

Pro Gruppe kann die Berechnung von Interessantheitsmaßen erfolgen. Ein Gruppenrepräsentant und dessen zugehörigen Maße werden dem Anwender zur Auswertung dargestellt. Dieser wählt aus allen Gruppen diejenigen aus, welche seinen Anforderungen im Bezug auf bestimmte Maße oder Maßkombinationen entsprechen. Der Nutzer kann die ihm gezeigten Gruppen auf deren subjektiven Wert prüfen und effizient die Regelmenge auf die für ihn relevanten Einträge reduzieren.

Abbildung 4.3 zeigt den Ablauf dieses Verfahrens. Der linke Pfad führt die Berechnung der einzelnen Interessantheitsmaßen durch, während die Regeln im rechten Pfad parallel dazu um ihre Kontingenzmatrix erweitert und nach ihr gruppiert werden. Die dabei entstehende Gruppenkennzeichnung, z.B. der Gruppenrepräsentant, wird jeder Regel hinzugefügt. Anhand dieser Kennzeichnung kann die Darstellung und Filterung durch den Nutzer auf Ebene der Gruppen erfolgen.

Die Matrizen enthalten hierbei die absoluten Häufigkeiten der einzelnen Regelelemente. Zwei Regeln Y_1 und Y_2 gelten als Nachbarn, wenn sich jedes Element der Kontingenzma-

trix von Y_1 in der r-Nachbarschaft von dem äquivalenten Eintrag der Matrix Y_2 befindet. Als Definition der Nachbarschaft wird in dieser Arbeit eine maximaler prozentualer Unterschied ϕ zwischen zwei Elementen vorgeschlagen. Sind also alle Elemente einer Kontingenzmatrix um höchstens ϕ von ihrem Gegenstück verschieden, gelten sie als Nachbarn und gehören damit zur selben Gruppe. Das bewirkt, dass sich alle Gruppenmitglieder in jedem ihrer Matrixelemente im Radius ϕ um den Repräsentanten befinden. Der maximale Radius des Clusters ist somit auf $2 \cdot \phi$ begrenzt.

Für viele andere Clusterverfahren gilt diese Eigenschaft nicht. Sie bewirkt jedoch gleichartige Interessantheitsmaße für alle Mitglieder einer Gruppe. Begonnen wird die Zuordnung von Regeln zu Gruppen absteigend nach der Platzierung der Regeln innerhalb der gesamten Regelmenge. Befindet sich eine Regel nicht in Nachbarschaft eines Gruppenrepräsentanten, bildet sie eine neue Gruppe. Auf diese Weise wird jeweils die bestplatzierte Regel zum Repräsentanten der Gruppe. Anschließend werden alle weiteren nicht gruppierten Regeln Y_1 einer Gruppe zugeordnet, wenn sie sich in der ϕ-Nachbarschaft des Repräsentanten der Gruppe befinden. Der Wert der Regel wird nicht durch deren spezifischen Wert oder deren Aussage, sondern durch deren Gruppe bestimmt.

Das folgende Beispiel in Tabelle 4.4 soll das Verfahren verdeutlichen. Das Verfahren beruht auf der Hoffnung, dass eine große Menge von Regeln auf eine geringe Anzahl an Regelgruppen reduziert wird. Entsprechend müssen deutlich mehr Regeln als Gruppen vorhanden sein. Das Beispiel muss daher mit unrealistischen Regeln gezeigt werden, um diese Abweichung von den Annahmen des Verfahrens auszugleichen. Es werden somit Kennzahlen für eine Auswahl an sechs Regeln definiert, welche anschließend zwei Gruppen bilden.

Regel	**Kontingenzmatrix**				**Gruppe**
$A \Rightarrow C$	$\lvert A \cup C\rvert$	$\lvert \neg A \cup C\rvert$	$\lvert A \cup \neg C\rvert$	$\lvert \neg A \cup \neg C\rvert$	
R1	100	50	80	770	G1
R2	200	300	330	170	G2
R3	80	45	85	790	G1
R4	220	280	300	200	G2
R5	180	310	320	190	G2
R6	110	55	80	755	G1

Tabelle 4.4.: Beispiel: semantische Gruppierung

Der erlaubte Unterschied pro Element soll maximal 20% betragen. Die vier Werte der Regeln entsprechen deren Kontingenzmatrix. Die Regeln $R1$ bis $R6$ sind bereits gemäß einer Gesamtplatzierung sortiert. Somit beginnt die Gruppierung mit $R1$. Da noch keine Gruppe vorhanden ist, bildet $R1$ den Repräsentanten der Gruppe $G1$. Die nachfolgende Regel $R2$ weicht in jedem Element um mehr als 20% von $G1$ ab. Sie bildet somit eine neue Gruppe $G2$. Die Regel $R3$ ist vollständig in der Nachbarschaft von $R1$, dem Repräsentanten von $G1$. Somit wird $R3$ der Gruppe $G1$ zugeordnet. $R4$ ist kein Nachbar von

$R1$, jedoch in der Nähe von $R2$. Somit kann $R4$ der Gruppe $G2$ mit dem Repräsentanten $R2$ hinzugefügt werden. Äquivalent wird $R5$ als Nachbar von $R2$ ein Mitglied der Gruppe $G2$ und $R6$ Teil von $G1$.

Eine Regel kann vollständig in der Nachbarschaft mehrerer Gruppen sein, wenn sie sich in der Nachbarschaft derer Gruppenrepräsentanten befindet. In diesem Fall wird die Regel der Gruppe mit dem bestplatzierten Repräsentanten zugeordnet. Alternativ kann hier auch die Gruppe mit der größeren Übereinstimmung gewählt werden. Auch eine jeweils zufällige Zuordnung ist möglich. In dieser Arbeit wird der Fall einer mehrfachen Zuordenbarkeit einer Regel als seltenes Ereignis betrachtet, dessen Behandlung keinen signifikanten Einfluss auf das Ergebnis nimmt. Die Reduktion der Einzelergebnisse wird in allen Fällen gleichermaßen erreicht, da diese Regel jeweils durch eine bereits vorhandene Gruppe repräsentiert wird und damit nicht mehr einzeln bewertet werden muss.

Im Gegensatz zu einer syntaktischen Gruppierung erfolgt eine semantische Gruppierung vollständig unabhängig von der durch die Regeln repräsentierten Aussage. Daher sind keine Metainformationen nötig, welche die zu gruppierenden Regeln erläutern. Ein Anwender muss somit keine zusätzlichen Informationen bereitstellen. Das begünstigt und vereinfacht die maschinelle Aufbereitung der Informationen. Syntaktische Eigenheiten wie beispielsweise hierarchische Abhängigkeiten werden nicht betrachtet. Eine derartige Aufbereitung ist weitestgehend unabhängig von dem jeweiligen Szenario und kann automatisiert durchgeführt werden. Auf diese Weise wird eine effiziente Einschränkung der kompletten Regelmenge auf einige wenige subjektiv wichtige Regeln ermöglicht, welche anschließend auch anhand ihrer Syntax bewertet werden können.

Der folgende Abschnitt soll die Tauglichkeit der vorgestellten Verfahren zur automatischen Aufbereitung großer Assoziationsregelmengen untersuchen.

4.3. Evaluation

Die Auswertung der in diesem Kapitel vorgestellten Verfahren nutzt die im Anhang A, Seite 183 dargestellten Datenbestände. Die Parametrisierung der gegebenenfalls nötigen EHK wird gegebenenfalls im Text beschrieben. Ein direkter Vergleich zu aus der Literatur bekannten Verfahren oder verschiedenen Interessantheitsmaßen erfolgt nur in geringem Maße, da jeweils die Qualität des entstehendes Resultat wie auch bei deren Auswertungen stark subjektiv vom Betrachter abhängt.

Die Güte eines Ergebnisses kann somit nur durch den Nutzer pro Suchziel und Datenbestand erfolgen. Eine allgemeine Bewertung ist nicht möglich. Die folgende Auswertung zeigt stattdessen anhand von Beispielen die Wirksamkeit der erarbeiteten Verfahren.

	Conf	**Lift**	**Conv**	**Nov**	**Cov**	**Prev**	**Chi**	**Top-R**
R01	9	23	7	8	65	29	2	143
R02	4	15	3	17	83	28	3	153
R03	20	78	21	2	45	5	8	179
R04	16	69	17	3	52	13	16	186
R05	12	26	8	27	90	33	11	207
R06	3	12	2	45	107	40	10	219
R07	39	54	31	5	46	31	14	220
R08	10	64	14	13	73	15	39	228
R09	40	22	22	21	66	58	5	234
R10	44	77	55	1	16	39	7	239

Tabelle 4.5.: Ausschnitt Top-100 aus *Retail*

4.3.1. Regelbewertung durch Interessantheitsmaße

Dieser Abschnitt verdeutlicht die Eigenschaft von Interessantheitsmaßen, Assoziationsregeln sehr unterschiedlich zu bewerten. Tabelle 4.5 zeigt exemplarisch die Platzierungen von zehn Regeln aus dem Retail-Datenbestand mit einer Mindesthäufigkeit von 1%, sortiert nach ihrer aufsummierten Platzierung (Top-R) über die gewählten Interessantheitsmaße. Genutzt wurden die Maße Konfidenz, Lift, Überzeugung, Novelty, Coverage (Häufigkeit der Bedingung), Prevelance (Häufigkeit der Schlussfolgerung) und χ^2. Es ist ersichtlich, dass alle Interessantheitsmaße die zugrunde liegenden Regeln teilweise sehr verschieden beurteilen. Dabei sind sowohl ähnliche Bewertungen erkennbar (*Conf* und *Conv*) als auch verschiedene Platzierungen (*Conf* und *Lift*). Die Beurteilung einer Regel hängt somit signifikant von einem jeweils betrachteten Interessantheitsmaß ab. Sie erfordert einen hohen Grad an Verständnis für die Bedeutung der einzelnen Maße. Während ein professioneller Nutzer die wichtigste Bewertung gemäß seiner gewünschten Sichtweise wählen kann, muss eine automatisch agierende Lösung ohne Syntaxinformationen alle Betrachtungsweisen gleiche gewichten. Jede Betrachtungsweise ist als die vom Nutzer bevorzugte Variante möglich und muss erhalten bleiben.

Die Top-R-Bewertung zeigt, dass keines der genutzten Einzelmaße die Gesamtbewertung widerspiegelt. Allerdings ist ersichtlich, dass die Gesamtbewertung vor allem Regeln beachtet, welche mehrfach gut platziert waren. Die Chance einer Regel, dem bevorzugten Bewertungsmaßstab eines Anwenders zu entsprechen wird damit erhöht, was deren gute Gesamtplatzierung rechtfertigt.

Auswertungen auf weiteren Datenbeständen zeigen ähnliche Ergebnisse wie *Retail* in Tabelle 4.5. In dieser Arbeit soll sich jedoch auf dieses Beispiel beschränkt werden.

4.3.2. Korrelationen zwischen Interessantheitsmaßen

Um unterschiedliche Korrelationen zwischen Interessantheitsmaßen zu zeigen, werden im Folgenden drei Beispiele angeführt. Betrachtet werden die Top-100-Kombinationen mit jeweils mehr als einem Element. Die Korrelation zwischen zwei Maßen soll im Bezug auf die Platzierungsvektoren der Maße einen Pearson-Koeffizienten von mindestens $0,8$ als Mindestähnlichkeit aufweisen, was gemäß einem Koeffizienten von 0 bei Unabhängigkeit und bei 1 vollständiger Korrelation eine hohe Abhängigkeit zwischen den Maßen bedeutet.

Der SynthA-Datenbestand liefert neben den zueinander unabhängigen Bewertungen von Überzeugung, All-Confidence, Minimum Improvement, Odds Ratio, Recall, Relative Risk, YulesQ, Coverage und Prevalence folgende Maßgruppen

1. Lift, Konfidenz, χ^2, Change of Support, Mutual Support, J-Measure, Accuracy, Certainty Factor und Pearson-Koeffizient.

2. Häufigkeit und Novelty

Für Retail-Daten sind Überzeugung, Accuracy, Relative Risk und Prevalence unabhängig. Gruppen werden gebildet durch

1. Lift, χ^2, Mutual Support, Odds Ratio, J-Measure, Recall und Pearson-Koeffizient.

2. Konfidenz, Change of Support, Certainty Factor und Yules Q.

3. Häufigkeit, Novelty und Coverage.

4. All-Confidence und Minimum Improvement.

Überzeugung, Accuracy, Relative Risk und Prevalence sind für KundeA unabhängig. Außerdem gelten folgende Gruppierungen:

1. Lift, χ^2, Mutual Support, Odds Ratio, J-Measure, Recall und Pearson-Koeffizient.

2. Konfidenz, Change of Support, Certainty Factor.

3. Häufigkeit, Novelty und J-Measure.

Dabei kann beobachtet werden, dass einige Maße für alle Beispiele in den verschiedenen Datenbeständen gleich korrelieren, aber andere Maße auch die Gruppe wechseln oder als unabhängig betrachtet werden. Insgesamt kann gezeigt werden, dass Interessantheitsmaße in Abhängigkeit von den zugrunde liegenden Daten unterschiedlich miteinander korreliert sein können. Während einige Maße (wie hier beispielsweise Lift und χ^2) auch

Gruppiere nach	**SynthB**	**Retail**	**SynthC**	**Connect-4**	**Gazelle**	**Mushroom**
Ausgangs-kombination	29/20	83/13	40/34	37/37	53/19	30/24
Bedingung	45/21	92/6	45/19	36/12	75/15	54/27
Schluss-folgerung	37/26	14/10	49/22	49/13	39/18	15/11
Nachbar-schaft	15/15	43/17	25/21	19/19	29/17	16/12
Generali-sierung	80/21	53/31	88/24	68/8	67/28	35/31

Tabelle 4.6.: Aus 100 Regeln gebildete und regelreduzierende Gruppen (1)

auf verschiedenen Regelmengen ähnliche Bewertungen erbringen, gilt dies nicht für alle Maße. Während die Aussage von Accuracy im Retail-Datensatz beispielsweise durch andere Maße bereits repräsentiert wird, gilt dies nicht für die SynthA-Daten. Pro Datenbestand ist somit eine Korrelationsanalyse mit nachfolgender Auswahl repräsentativer Maße sinnvoll.

4.3.3. Gruppierung von Assoziationsregeln

Tabelle 4.6 zeigt die Kompressionsfähigkeit verschiedener Gruppierungsmöglichkeiten auf einer Regelmenge. Dies zeigt die verschiedenen Ansätze vergleichend im Überblick. Gruppiert wurden pro Datenbestand jeweils 100 Regeln. Das Gruppierungskriterium wird in der ersten Spalte beschrieben, beispielsweise nach Regelbedingung, Schlussfolgerung oder Nachbarschaft. Pro Eintrag werden sowohl die Anzahl der entstehenden Gruppen aus den 100 Ausgangsregeln als auch die Anzahl der Gruppen mit mehr als eine Regel gezeigt. Nur diese reduzieren die Ergebnismenge.

Als beste Regel einer Generalisierungsgruppe gilt in dieser Betrachtung die Regel mit den meisten Elementen in der Bedingung. Alle spezielleren Regeln sind von dieser repräsentiert. Eine gruppierte Darstellung ist dadurch ein sinnvoller Einstiegspunkt für die Navigation und eine übersichtliche Darstellung.

Die Zusammenfassung der Regeln nach der Gesamtkombination reduziert die Ergebnismenge um bis zu Faktor sieben. Hierbei sind sowohl Gruppen mit einer großen Anzahl an Assoziationsregeln, als auch solche mit sehr wenigen oder nur einer Assoziationsregel interessant. Ist der Umfang einer Gruppe sehr groß, ist die zugrunde liegende Kombination wahrscheinlich wichtig, da sie eine Vielzahl an Regeln erzeugt. Die einzelnen Regeln hingegen sind eher unwichtig, da sie den anderen Mitgliedern der Gruppe ähneln.

Gruppen mit sehr wenigen zugehörigen Regeln bilden Ausreißer und stellen damit eine Besonderheit und eine daraus resultierende erhöhte Interessantheit dar.

Die Gruppierung nach Bedingung oder Schlussfolgerung einer Regel erlaubt ähnliche Kompressionsraten, vereinigt jedoch auch Regeln mit sehr unterschiedlichen Schlussfolgerungen bzw. Bedingungen. Trotzdem ist diese Darstellung sinnvoll, da ein effizientes hierarchisches Filtern nach bestimmten Regelteilen ermöglicht wird.

Durch das Verwenden der kompakten Darstellung kann die dem Nutzer zur Verfügung gestellt Regelmenge deutlich erhöht werden, da dieser nicht das komplette Ergebnis auswerten muss. Damit bleiben Regeln erhalten, welche bei der Reduktion auf die Top-R-Regeln entfallen. Zu beachten ist hierbei, dass außer bei der Gruppierung nach Nachbarschaft keine Steuerparameter möglich sind. Dies gezeigten Gruppierungen sind in diesen Fällen die maximal mögliche Reduktion. Ist diese also nicht ausreichend, kann bei den entsprechenden Verfahren keine Verbesserung des Ergebnisses erreicht werden.

Die Zeile *Nachbarschaften* in Tabelle 4.6 zeigt die Anzahl resultierender Gruppen für r-Nachbarschaften als Gruppierungskriterium. Im verwendeten Beispiel werden die Gewichtungen $\rho_b = 1$, $\rho_s = |E'| - 1/|E'|^2$ und $\rho_g = 1/|E'|^2$ definiert. Des Weiteren wird $r = 2$ verwendet. Durch diese Art Gruppierung können durchschnittlich zehn Regeln zu einer Gruppe zusammengefügt werden. Aber auch hier sind die möglichen Radien sehr begrenzt, da zumindest in Verkaufsdaten zwar viele Elemente enthalten sind, aber häufige Kombinationen nur wenige Elemente enthalten. Der maximal mögliche Abstand zwischen zwei Regeln bei völlig disjunkten (Teil-)Kombinationen ist somit oft gering und bietet damit wenig Raum bei der Wahl von r.

Semantische Regelgruppierung

In der folgenden Auswertung wird demonstriert, wie Regelmengen durch die semantische Gruppierung nach Kontingenzmatrix und der semantischen Distanz nach Toivonen im gegenseitigen Vergleich komprimiert werden können. Dabei wird die bessere Kompression bei gleichzeitig ähnlicheren Werten der Interessantheitsmaße pro Gruppe bei Anwendung der Matrixgruppierung gezeigt. Eine Assoziationsregel wird einer Gruppe zugeordnet, wenn die Häufigkeiten aller Matrixelemente jeweils um maximal ϕ von denen des Gruppenrepräsentanten abweichen. Die Darstellung erfolgt äquivalent zur Tabelle 4.6. Die erste Zahl zeigt jeweils die Anzahl entstehender Gruppen, gefolgt von der Anzahl der Gruppen mit mehr als einem Mitglied.

Der Umfang der Resultate ist abhängig vom untersuchten Datenbestand, aber die Reduzierung ist in den Beispielen jeweils deutlich erkennbar. Wie schon bei den Untersuchungen der syntaktischen Gruppierungen kann auch aus semantischer Sicht ein sehr unterschiedliches Kompressionsverhalten beobachtet werden. Die entstehenden Gruppen unterscheiden sich jedoch sowohl in ihrem Umfang als auch dem Inhalt deutlich in den jeweiligen Gruppierungsmethoden.

	SynthB	**Retail**	**SynthC**	**Connect-4**	**Gazelle**
sem. Dist. $\phi = 0,1\%$	63/33	91/06	89/11	36/21	100/00
sem. Dist. $\phi = 1\%$	62/33	90/06	40/21	07/07	83/09
sem. Dist. $\phi = 5\%$	06/06	88/06	06/04	01/01	68/13
sem. Dist. $\phi = 10\%$	04/04	86/06	05/04	01/01	66/13
sem. Dist. $\phi = 20\%$	04/04	73/14	05/04	01/01	56/14
sem. Dist. $\phi = 40\%$	04/04	66/18	05/04	01/01	38/19

Tabelle 4.7.: Aus 100 Regeln gebildete und regelreduzierende Gruppen (2)

Tabelle 4.7 zeigt die Gruppierungsfähigkeit nach der von Toivonen vorgeschlagenen semantischen Distanz. Um einen besseren Vergleich der verschiedenen Verfahren zu ermöglichen, wurde die für die semantische Distanz relevante Abdeckung zweier Regeln modifiziert. Nach Toivonen ist die semantische Distanz zwischen zwei Regeln als die Summe der absoluten Regeläufigkeiten minus der doppelten absoluten Häufigkeit der Vereinigung der beiden Regeln definiert. Dies erschwert eine Gruppierung, da der Wert der Distanz von den jeweils in einem Datenbestand vorhandenen Kombinationen und deren absoluten Häufigkeiten abhängt. In dieser Evaluation wird die semantische Distanz daher definiert als

$$Distanz_{sem}(A_1 \Rightarrow C_1, A_2 \Rightarrow C_2) = 1 - \frac{2 \cdot support(A_1 \cup C_1 \cup A_2 \cup C_2)}{support(A_1 \cup C_1) + support(A_2 \cup C_2)}.$$

Mit dieser Rechenvorschrift wird die Überschneidung zweier Regeln entsprechend Toivonen beachtet, wobei 100% eine vollständige Deckung bedeuten und 0% die nicht vorhandene Abdeckung. Diese Deutung entspricht der Distanzmessung zwischen den Abständen zweier Kontingenzmatrizen. Als Gruppierungsalgorithmus wurde das Verfahren der Kontingenzmatrixgruppierung angewendet. Die häufigste Regel wird Repräsentant und alle weiteren Regeln innerhalb der Distanz ϕ um den Repräsentanten werden Teil der Gruppe.

Tabelle 4.8 verdeutlicht die Auswirkungen des Gruppenradius ϕ nach ähnlicher Kontingenzmatrix. Dabei wird ersichtlich, dass die Gruppierung mit steigendem Radius verstärkt wird und somit durch Variation von ϕ die Regelmenge dynamisch verringert werden kann. Da jeweils 100 Regeln als Ausgangsbasis dienten wird jedoch auch ersichtlich, dass die Gruppierungsfähigkeit von den untersuchten Daten abhängt. Der Prozess der Gruppierungen ist dabei effizient durchführbar, was eine schrittweise Anpassung von ϕ erlaubt. Auf diese Weise kann ein Nutzer gezielt die Regelmengen auf den gewünschten Umfang verkleinern.

	SynthB	**Retail**	**SynthC**	**Connect-4**	**Gazelle**	**Mushroom**
Kont. mat. $\phi = 0,1\%$	78/15	100/100	36/21	53/27	100/00	56/29
Kont. mat. $\phi = 1\%$	65/15	100/100	34/21	42/26	99/01	51/31
Kont. mat. $\phi = 5\%$	40/17	83/12	22/17	30/18	89/10	32/24
Kont. mat. $\phi = 10\%$	24/14	67/19	19/15	18/15	42/20	22/18
Kont. mat. $\phi = 20\%$	14/11	43/19	14/12	12/10	42/17	11/11
Kont. mat. $\phi = 40\%$	10/08	24/16	07/06	06/06	18/10	05/05

Tabelle 4.8.: Aus 100 Regeln gebildete und regelreduzierende Gruppen (3)

	SynthB		**Retail**		**SynthC**		**Connect-4**		**Gazelle**	
	Dist	Kont	Dist	Kont	Dist	Kont	Dist	Kont	Dist	Kont
$\phi = 0,1\%$	3	0	10	0	9	0	22	0	0	0
$\phi = 1\%$	20	0	11	0	9	0	27	0	18	0
$\phi = 5\%$	44	2	12	0	40	1	76	1	14	1
$\phi = 10\%$	43	4	14	1	56	2	82	1	16	1
$\phi = 20\%$	54	11	9	2	52	6	81	9	17	5
$\phi = 40\%$	41	22	9	4	46	14	93	24	19	17

Tabelle 4.9.: Durchschnittlicher Fehler pro Gruppe in Prozent

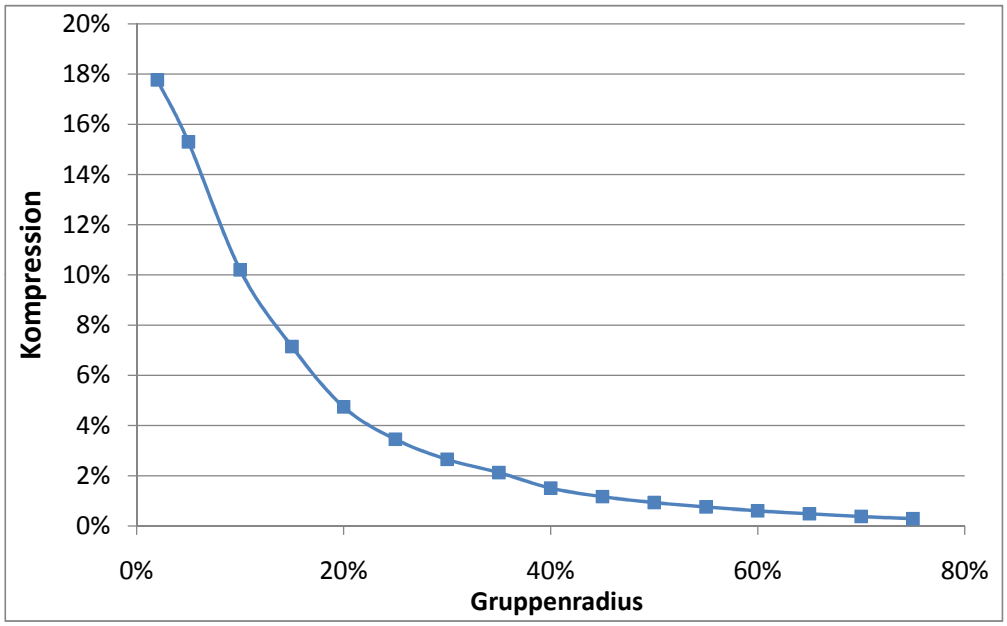
Im Vergleich der Gruppierungen nach semantischer Distanz und nach Kontingenzmatrix zeigen sich ähnliche Gruppierungsmöglichkeiten. Die Abhängigkeit der Gruppierung von ϕ ist jedoch bei der Kontingenzmatrixgruppierung deutlich harmonischer, was die Wahl dieses Parameters erleichtert. Die semantische Distanz nach Toivonen neigt in einigen Datenbeständen zum Übersteuern und gruppiert extrem stark. Ein wichtiger Punkt ist zudem die Diversität innerhalb der entstehenden Gruppen. Nach Toivonen werden die einzelnen Regelelemente der gruppierten Regeln nicht beachtet. Entsprechend deutlich schwanken die in einer Gruppe auftretenden Interessantheitsmaße der Regeln. Bei der Gruppierung nach Kontingenzmatrix ist dies nicht der Fall, was die Bewertung der Gruppe deutlich erleichtert. Hinzu kommt der oft nötige Mehraufwand zur Berechnung der Häufigkeit des gemeinsamen Auftretens zweier Regeln, was für deren semantische Distanz benötigt wird. Diese sind oft nicht vorhanden und müssen nachträglich bestimmt werden, was vor allem bei vielen zu gruppierenden Regeln einen signifikanten Mehraufwand zur Berechnung der Gruppierung bedeuten kann.

Tabelle 4.9 zeigt die Ähnlichkeiten der Interessantheitsmaße pro Gruppe an einigen Beispielen. Betrachtet wird der durchschnittliche Fehler bei den Interessantheitsmaßen Novelty, Überzeugung, Recall, Regelhäufigkeit, Konfidenz, Lift, Odds Ratio, Change of Support, Mutual Support und χ^2. Der Fehler gilt dabei jeweils in Relation zu den in den Regeln bei jedem Interessantheitsmaß vorhandenem Wertebereich. Wenn beispielsweise Konfidenzen zwischen 30% und 100% vorhanden sind, ist der Wertebereich der Konfidenz 70%. Weichen die Mitglieder einer Gruppe im Durchschnitt 5% von der Konfidenz des Repräsentanten ab, ist der Fehler $5\%/70\% = 7,1\%$. Je geringer der Fehler ist, umso kleiner ist der in einer Gruppe vorhandene Wertebereich eines Maßes. Dabei zeigt sich, dass bei der Gruppierung nach Kontingenzmatrix gegenüber der semantischen Distanz ein deutlich geringerer Fehler auftritt und die Mitglieder einer Gruppe somit sehr ähnliche Interessantheitsmaße besitzen.

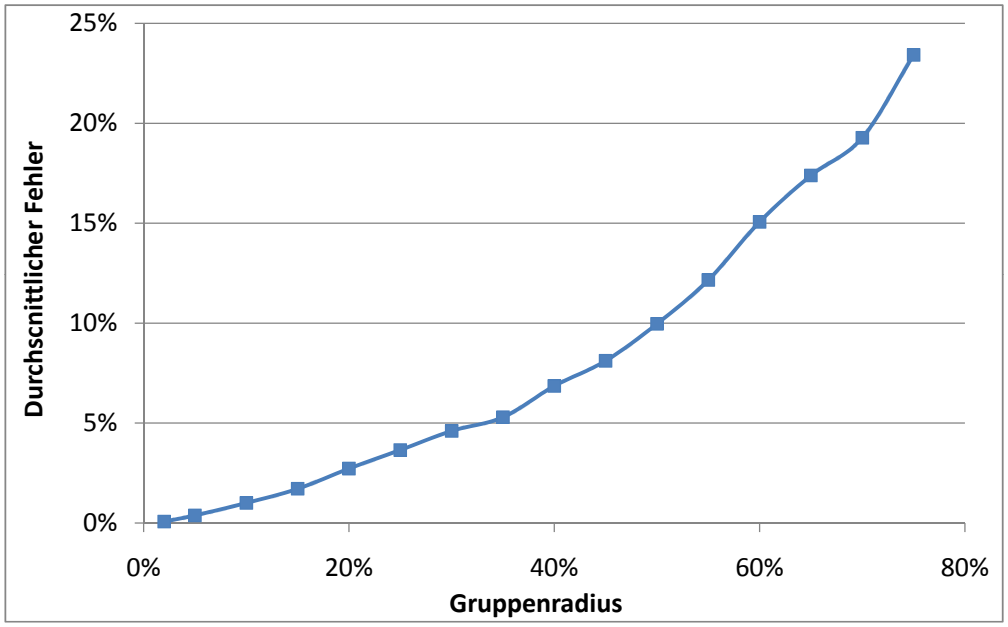
Eine vereinfachte Navigation in der betrachteten Regelmenge kann sowohl durch syntaktische als auch semantische Gruppierung gezeigt werden. Die Art der Filterung ist hingegen sehr unterschiedlich. Während eine syntaktische Gruppierung die Interessantheitsmaße einer Regel ignoriert, wird in einer semantischen Betrachtung die Aussage der Regel nicht beachtet. Somit sind beide Möglichkeiten ungeeignet, ein endgültiges, abschließendes Urteil über eine Regel zu erzeugen. Eine auf diesen Gruppierungen basierende Bereinigung der zu untersuchenden Regelmenge kann hingegen deutlich vereinfacht werden. An dieser Stelle kann der Anwender je nach der Vorgehensweise seine Navigation zwischen semantischer und syntaktischer Gruppierung wählen.

Abbildung 4.4 zeigt das Verhalten der Regelgruppierung an einem Beispiel mit den Top-7.500 Regeln des Retail-Datenbestandes. Die Reduktion der einzelnen Regeln auf Regelgruppen verdeutlicht Abbildung 4.4(a). 100% entsprechen hierbei den 7.500 Ausgangsregeln. Der maximale Unterschied $\phi = 5\% \ldots 75\%$ zwischen den Kontingenzmatrizen der einzelnen Regeln zum jeweiligen Gruppenrepräsentanten werden auf der X-Achse dargestellt. Die verbleibenden Gruppen werden bei wachsendem ϕ deutlich reduziert. Bereits bei $\phi = 5\%$ werden durch eine Gruppe im Durchschnitt fünf Regeln repräsentiert, bei $\phi = 40\%$ bereits mehr als 50 der 7.500 Regeln. Abbildung 4.4(b) zeigt im Vergleich dazu den durchschnittlichen Fehler bei den Interessantheitsmaßen Novelty, Überzeugung, Recall, Regelhäufigkeit, Konfidenz, Lift, Odds Ratio, Change of Support, Mutual Support und χ^2. Die Berechnung des Fehlers erfolgt äquivalent zu Tabelle 4.9. Dabei zeigt sich, dass der durchschnittliche Fehler auch bei großen ϕ noch sinnvolle Ergebnisse erlaubt. Bei $\phi = 40\%$ werden die 7.500 Ausgangsregeln zu 115 Gruppen vereinigt, wobei die zehn Interessantheitsmaße im Durchschnitt weniger als 7% von denen des Gruppenrepräsentanten abweichen. Eine gemeinsame Bewertung der Gruppe ist damit oft möglich. Die Navigation und Filterung der Ergebnismenge kann mit diesem Verfahren insgesamt vereinfacht und beschleunigt werden.

Diese Evaluation zeigt, dass je nach den Anforderungen des Anwenders verschiedene Vorgehensweisen bei der Aufbereitung einer Regelmenge sinnvoll sind. Nicht alle Möglichkeiten sind gleichermaßen automatisiert möglich. Die Syntax der Regel ist nicht immer zum



(a) Regelreduktion



(b) Durchschnittlicher Fehler

Abbildung 4.4.: Regelgruppierung mit Top-7.500 bei *Retail*

Zeitpunkt der Regelaufbereitung bekannt oder kann vom Anwender bereitgestellt werden. In diesen Fällen ist eine semantische Gruppierung die einzig verbleibende Möglichkeit der Filterung. Denkbar sind auch Szenarien, in denen die Syntax der Regel wenig bedeutend ist und deren Wert lediglich durch die Ausprägungen einiger Interessantheitsmaße gezeigt wird. Ein Anwender wird hier schrittweise Regel mit unerwünschten Werten ausschließen bis eine gewünschte Regelmenge verbleibt. Die Elemente einzelner Regeln bleiben vorerst vergleichsweise unbedeutend, da diese oft schwierig zu bewerten sind. Dieses Vorgehen, beispielsweise verwendet bei Fehleranalysen, wird durch eine semantische Gruppierung besser unterstützt als durch die Verwendung syntaktischer Eigenschaften.

4.4. Zusammenfassung

Zur Unterstützung des Anwenders wurden in diesem Kapitel Verfahren aufgezeigt, große Regelmengen derartig aufzubereiten, dass die manuelle Auswertung effizient möglich wird. Dies erfolgte einerseits durch die Bestimmung einer Rangfolge für die einzelnen Regeln. Andererseits wurden Ansätze untersucht, welche mehrere Regeln zu Gruppen mit jeweils ähnlichen Eigenschaften vereinigen.

Ein Problem des ersten Ansatzes, der Bildung einer Rangfolge, besteht in der meist subjektiven Bewertung der Regeln durch den Anwender. Während objektive Interessantheitsmaße, wie z.B. die Regelhäufigkeit, jeweils aus ihrem Blickwinkeln eine sinnvolle Bewertung der Regeln aufzeigen, unterscheiden sich diese Einschätzungen von Maß zu Maß teilweise deutlich. In dieser Arbeit wird eine signifikante Entlastung des Nutzers angestrebt. Der Anwender muss somit ausdrücklich nicht in der Lage sein, das gewünschte Suchziel in die Wahl und Gewichtung einzelner objektiver Maße zu überführen. Eine Maschine kann jedoch nicht entscheiden, wie ein Anwender die Regeln subjektiv werten wird und hat lediglich einige Interessantheitsmaße zur Verfügung.

In Abschnitt 4.1 wurde die Bildung einer Menge von Interessantheitsmaßen vorgeschlagen, welche möglichst viele verschiedene Bewertungen der Regeln darstellen. Um Vielseitigkeit zu bewahren, mussten korrelierte Maße je nach untersuchter Regelmenge konsolidiert werden. Hierfür wurde gezeigt, wie die Maße normalisiert und auf Korrelationen untersucht werden können. Darauf basierend wurde eine aufsummierte globale Rangfolge der Regeln bestimmt. Der Anwender kann die besten Regeln nach dieser Bewertung auswählen und seine Auswertung auf Ergebnisse beschränken, welche aus mehreren Blickwinkeln wichtig erscheinen. Auf diese Weise konnte die robuste Erkennung wichtiger Regeln erreicht werden. Der Anwender kann zudem hoffen, zumindest einen Teil seiner subjektiv wichtigen Regeln in den oberen Gesamtplatzierungen zu finden. Werden diese Top-R-Regeln als Navigationsvorschlag in Form von Regelvorlagen genutzt, sind in einem nachfolgenden Schritt Suchen nach ähnlichen Regeln möglich. Ein Nutzer kann

damit eine einfache Spezifikation seiner Suchziele beschreiben, ohne konkrete objektive Maße kennen und deuten zu müssen.

Hierbei besteht die Gefahr, dass die subjektive Sichtweise des Nutzers nicht genügend in die Bewertung eingeflossen ist. Wichtige Regeln können ungewollt entfernt oder schlecht platziert worden sein. Daher kann es nötig sein, die vollständige, umfangreiche Regelmenge auszuwerten. Deren Eingrenzung auf wichtige Regeln muss dabei mit vertretbarem Aufwand des Nutzers möglich sein. Entsprechend müssen unwichtige Regeln in großen Blöcken entfernt werden. Hierfür wurden Möglichkeiten zur Gruppierung von Regeln gezeigt. Diese konnten sowohl syntaktisch nach dem Inhalt der Regeln durchgeführt werden als auch semantisch nach den Werten von Interessantheitsmaßen.

Besonders wurden in dieser Arbeit semantische Gruppierungen gezeigt, welche einfach automatisch zu realisieren sind. Dabei wurden Regeln entsprechend ihrer Kontingenzmatrix mit den Häufigkeiten der Regelelemente zu einer Gruppe vereinigt, sollten diese Matrizen zueinander ähnlich sein. Unter der Annahme, dass diese Matrizen auch eine Vielzahl ähnlicher objektiver Interessantheitsmaße bilden, sind die Bewertungen aller Regeln einer Gruppe ähnlich. Sie können somit vereinigt durch einen Repräsentanten dargestellt werden. Dies hat den Vorteil, dass die Gruppierung unabhängig von den für den Nutzer wichtigen objektiven Bewertungskriterien ist. Die Gruppen können somit bereits vor der Präsentation für den Nutzer gebildet werden. Eine Interaktion oder die Bereitstellung von Metainformationen ist nicht erforderlich. Die abschließende Evaluation in Abschnitt 4.3 untersuchte die vorgestellten Verfahren und zeigt deren Verhaltensweisen anhand exemplarischer Regelmengen auf. Das folgende Kapitel erläutert abschließend die prototypische Implementierung der vorgeschlagenen Verfahren dieser Arbeit im SAP BW Accelerator.

5. Prototypische Implementierung

Dieses Kapitel soll die prototypische Implementierung der in dieser Arbeit vorgestellten Verfahren erläutern. Sämtliche Algorithmen wurden auf der Plattform des *SAP NetWeaver Business Warehouse Accelerator (BWA)* realisiert. Der nächste Abschnitt soll zunächst eine kurze Übersicht über die Besonderheiten und Eigenschaften des BWA geben. Danach wird gezeigt, wie die Lösungen dieser Arbeit zur Assoziationsregelsuche und -aufbereitung in die Architektur des BWA integriert wurden.

5.1. Aufbau des SAP BW Accelerator

Die Aufgabe des BWA ist die Berechnung von Aggregationen in einem erweiterten Sternschema des SAP BI. Dieses besteht aus einer zentral positionierten Faktentabelle mit Belegeinträgen, wie z.B. *Preis* und *Menge* eine Bestellung [CCS; KRTR98]. Diese Einträge sind im Regelfall aggregierbar. Informationen wie *Kunde* oder *Material* werden teilweise normalisiert in den die Faktentabelle umlagernden Dimensionstabellen abgelegt, welche über Fremdschlüsselbeziehungen mit den Belegeinträgen der Faktentabelle verknüpft werden.

Eine Weiterführung des Sternschemas mit zusätzlichen Normalisierungen wird als Schneeflockenschema bezeichnet. Abbildung 5.1 zeigt das Schema des in dieser Arbeit benutzen SAP BI InfoCube Architekturmodells [MWDI02]. Die zentrale Faktentabelle wird hier neben den Dimensionstabellen auch von X, Y und S Tabellen umgeben. Dabei speichern die X-Tabellen zeitabhängige (z.B. Adressen) und die Y-Tabellen zeitunabhängige Daten (z.B. Geburtsjahr). Die Dimensionstabellen (D) speichern die Informationen der einzelnen Dimensionen, z.B. *Material* oder *Kunde* und verbinden die X/Y und S-Tabellen mit der Faktentabelle. In S-Tabellen werden Surrogate-IDs gespeichert, welche Verweise in die zu einem InfoCube gehörenden Stammdaten darstellen.

Da einige Tabellen aus mehreren dieser Schemas die gleichen Daten enthalten und diese Daten auch die gleichen Informationen widerspiegeln, können Tabellen schemaübergreifend gemeinsam genutzt werden. Diese Eigenschaft erschwert die Optimierung der kompletten Datenbanklandschaft, erleichtert jedoch die Datenkonsistenz.

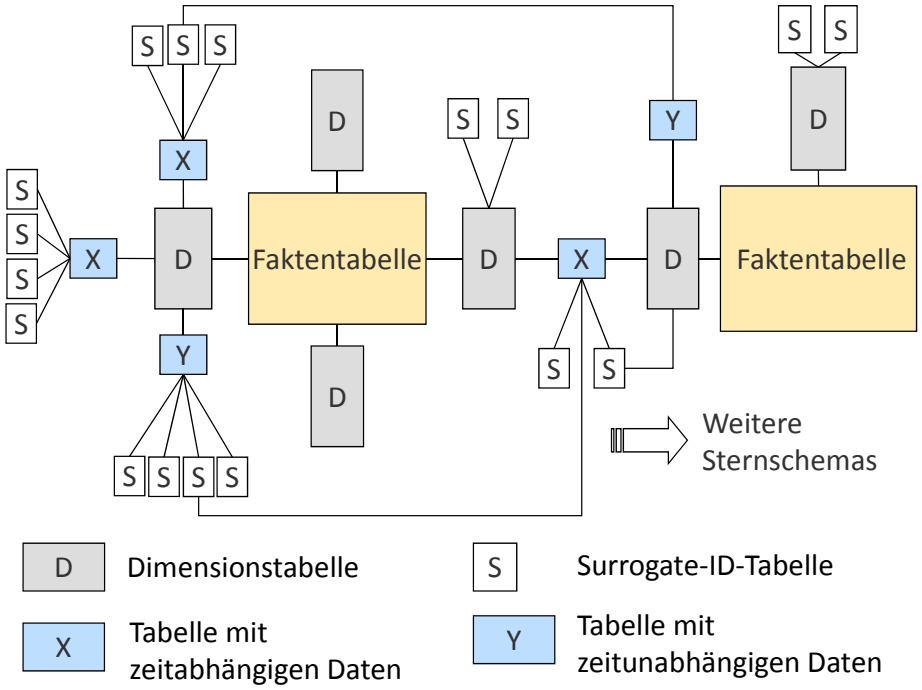


Abbildung 5.1.: Erweitertes SAP Sternschema

Anfragen auf das Schema benötigen oftmals mehrere aufeinanderfolgende Verbundoperationen über teilweise umfangreiche Tabellen mit mehreren Millionen Zeilen. Damit können Abbildungen, Hierarchien usw. aufgelöst und Daten aus der Faktentabelle selektieren werden, auf denen eine Aggregation erfolgt. Um Berechnungen bei großen Datenbeständen von bis zu mehreren Milliarden Einträgen in der Faktentabelle zu ermöglichen, werden in klassischen Datenbanksystemen materialisierte Sichten verwendet [GHQ95; BPT97]. Diese halten Daten aus einem Ausschnitt des Sternschemas in verdichteter Form redundant vor. Durch Umformung von Anfragen wird eine Nutzung der materialisierten Sichten ermöglicht. Da diese Daten im Vergleich zur ursprünglichen Faktentabelle kompakter abgelegt sind und weniger oder keine Verbundoperationen erfordern, kann die Anfrage in kürzerer Zeit beantwortet werden. Der nötige Aufwand zur Pflege der Sichten ist signifikant, da Änderungen in den Ursprungsdaten ein Nachführen in die materialisierten Sichten erfordern. Das verursacht sowohl zusätzlichen Rechenaufwand als auch die Gefahr von Inkonsistenzen. Treten beispielsweise Probleme während eines Änderungslaufes auf, müssen alle Änderungen transaktional sicher verworfen oder wiederholt werden. Die Wahrscheinlichkeit für fehlgeschlagene Schreiboperationen steigt somit an. Nicht zu vernachlässigen sind außerdem die Kosten zur Haltung der redundanten Daten, deren Umfang leicht ein Vielfaches der Ausgangsdaten beanspruchen kann.

Der SAP BWA löst die materialisierten Sichten ab und berechnet die Anfragen direkt auf den Ausgangsdaten. Durch Ross [Ros09] oder in [BD06; LLR06] sind weitere Infor-

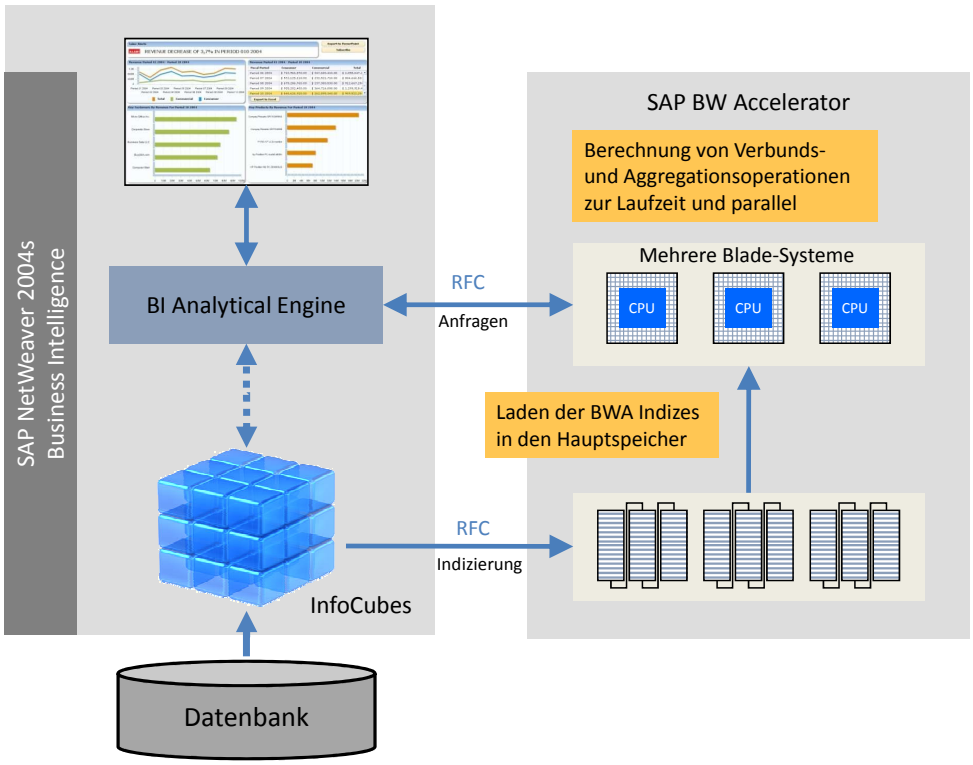


Abbildung 5.2.: Architektur des SAP BW Accelerator

mationen zum BWA verfügbar. In dieser Arbeit sollen lediglich dessen Grundkonzepte betrachtet werden. Der SAP BW Accelerator basiert auf der Suchmaschinentechnologie von SAP TREX. Alle Daten werden in einer einheitlichen internen Tabellenstruktur abgelegt und benötigen keine Zusatzstrukturen, wie z.B. Datenbankindizes oder Histogramme. Da keine Optimierung auf bestimmte Anfragen erfolgt, werden sowohl häufige als auch seltene Anfragen gleichermaßen effizient beantwortet. Abbildung 5.2 zeigt eine Übersicht über das Konzept des BWA, welche nachfolgend näher betrachtet werden.

Hauptspeicherbasierte Verarbeitung

Die Datenhaltung des BWA erfolgt in Regelfall vollständig im Hauptspeicher der beteiligten Knoten. Alle Daten werden zusätzlich auf nichtflüchtigen Speicher gehalten, sollten jedoch für eine hohe Leistungsfähigkeit möglichst selten von da gelesen werden. Vor der Beantwortung von Anfrage werden alle benötigten Daten in den Hauptspeicher geladen, wo sie bis zu ihrer Verdrängung durch andere Daten verweilen.

Der BWA speichert Tabellen spaltenweise. Ähnliche Datenstrukturen werden auch in C-Store [SAB+05], Vertica [Ver07] oder MonetDB [Bon05] verwendet. Derartig organisierte

Strukturen erlauben das gezielte Laden benötigter Attribute in den Hauptspeicher. Selten angefragte Attribute können bis zu ihrer ersten Verwendung auf externem Speicher verbleiben. Die Untersuchung von Kundensystemen und deren Anfragen ergaben, dass oft der Großteil der an einen SAP BI InfoCube gestellten Anfragen nur einen kleinen Teil der vorhandenen Attribute verwendet. Während eine spaltenorientierte Ablage nur relevante Attribute im Hauptspeicher vorhalten muss, benötigt eine zeilenorientierte Ablage die komplette Zeile mit allen Attributen. Für Analysezwecke ist eine spaltenorientierte Ablage einer Zeilenorientierung somit meist überlegen.

Skalierbare Multi-Server Architektur

Der BWA nutzt Blade-Systeme, einen Verbund von Einzelrechnern. Die Anzahl der zu einem Verbund gehörenden Knoten ist frei definierbar und kann dynamisch um Rechenknoten erweitert oder reduziert werden. Eine BWA-Installation kann je nach Menge der zu verwaltenden Daten über $6 - 100$ Blades mit je zwei bis acht Prozessoren und je acht Gigabyte (GB) Hauptspeicher[1] verfügen. Bei unzureichendem Hauptspeicher erfolgt eine Auslagerung ungenutzter Tabellen oder Attribute auf externe Speicher, was aus Effizienzgründen jedoch vermieden werden sollte.

Ein wichtiges Merkmal des BWA ist die Partitionierung großer Tabellen. Bisherige Ansätze zur verteilten Berechnung in einem Datawarehouse partitionieren die Faktentabelle auf alle zu einem Rechnerverbund gehörenden Knoten und replizieren die jeweils benötigten, umliegenden Tabellen [BM00]. Damit können verteilt Verbundoperationen effizient berechnet werden. Mit diesem Verfahren werden globale Anfragen auf den Einzelteilen der Faktentabelle parallel berechnet und zum Endergebnis vereinigt, ohne hohe Kommunikationskosten durch vorherige externe Datenleseoperationen zu erzeugen. Im BWA wird Redundanz vermieden, da diese zusätzlichen Speicher benötigen und die damit verbundenen Konsistenzanforderungen die Effizienz des Systems beeinträchtigen. Verbundoperationen werden vielmehr durch intelligente Verteilung der einzelnen Tabellen optimiert [LLR07].

Der BWA partitioniert Tabellen horizontal, wodurch effiziente parallele Aggregationen möglich sind. Auf diese Weise können Tabellen verarbeitet werden, deren Gesamtgröße den Hauptspeicher eines einzelnen Rechenknotens überschreitet. Zur Verwaltung einer verteilten Tabelle wird eine logische Tabelle definiert, welche Anfragen zentral entgegen nimmt und auf die entsprechenden physischen Partitionen aufteilt. Darauf hin werden die entstehenden lokalen Ergebnisse konsolidiert und an den Nutzer geliefert. Die Verteilung ist somit nach außen transparent.

[1] Stand: März 2009

Kompression durch ganzzahlige Abbildungen

Um Hauptspeicher zu sparen und ein schnelles Laden von Daten aus externem Speicher zu ermöglichen, werden beim BWA Daten komprimiert abgelegt. Da der Großteil der gespeicherten Daten konstant bleibt und bei einem regulären Datawarehouse-Szenario nur verhältnismäßig selten Einfüge- und Änderungsoperationen erfolgen, werden die Daten leseoptimiert abgelegt. Beim BWA erfolgt dies durch die Abbildung aller Werte auf natürliche Zahlen mit Hilfe von Wörterbüchern. Es wird dabei jeweils ein Wörterbuch pro Attribut angelegt. Diese Abbildung erlaubt eine kompakte Darstellung redundanter Datenbestände und ermöglicht effiziente Suchoperationen [LSF09]. Das Wörterbuch wird ebenfalls durch Front- und Golomb-Kodierung komprimiert [WMB99b]. Durchschnittliche Geschäftsdaten können damit auf ca. 10% bis 20% der Ausgangsdaten reduziert werden [LLR06; SV08].

Um Änderungen von Tabellen im begrenzten Umfang zu ermöglichen, nutzt der BWA einen Delta-Mechanismus. Änderungen werden in schreiboptimierter Form parallel neben der leseoptimierten Haupttabelle vorgehalten. Suchanfragen erfolgen jeweils auf Haupt- und Delta-Tabelle und die gefundenen Treffer werden vor der Rückgabe vereinigt. Diese Verarbeitung erfolgt transparent.

Optimierte Datenstrukturen

Das Metamodell des BWA stellt SAP InfoCubes logisch innerhalb des Systems dar. Dafür werden neben den an einem Schema beteiligten Tabellen auch Verbundpfade, erlaubte Verknüpfungen zwischen Tabellen, definiert. Die Vereinigung dieser Pfade stellt das in diesem InfoCube verwendete Schneeflockenschema dar. Verbundoperationen sind nur entlang dieser definierten Pfade möglich. Die Menge aller realisierbaren Berechnungen wird eingeschränkt, was gezielte Optimierungen ermöglicht. Der nachfolgende Abschnitt erläutert die Implementierung dieser Verbundpfade.

5.2. Assoziationsregelsuche im SAP BW Accelerator

Ein Ziel der Integration der Assoziationsregelsuche in den BWA ist die direkte Verarbeitung bereits für Analyseprozesse bereitgestellter Daten. Das beinhaltet die Verarbeitung von SAP BI InfoCubes inklusive der dafür bereitgestellten Funktionalitäten und Selektionen.

Realisiert werden diese Operationen im Wesentlichen über Abbildungsfunktionen, welche von der äußersten Tabelle bis zur Faktentabelle in jedem Verbundschritt transformiert werden. Ausgehend von einer Abbildung einer bestimmten Selektion auf deren ganzzahli-

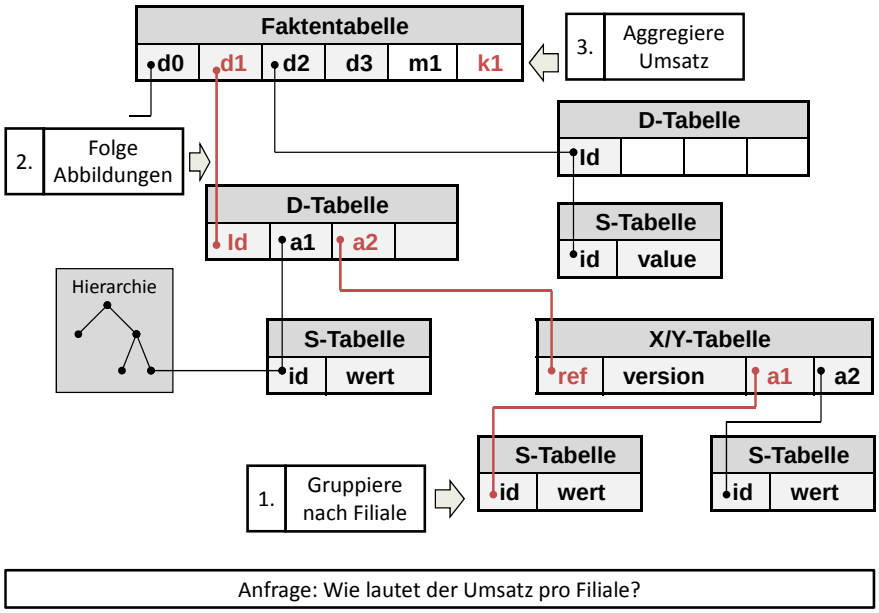


Abbildung 5.3.: Selektion und Verbund per Abbildungsfunktion

ge ID ergibt sich nach einer Umformung pro Verbundoperation eine Abbildung aller IDs in den Fremdschlüsseln der jeweiligen Dimension in der Faktentabelle.

In Abbildung 5.3 wird ein solcher Verlauf dargestellt. Die Abbildungsfunktion wird entlang der Markierung transformiert. Begonnen wird entsprechend der Anfrage an der äußersten Tabelle, hier beispielsweise die S-Tabelle mit den vorhandenen Filialen. Deren *id* wird mit dem Attribut *a1* der zugehörigen X/Y-Tabelle verknüpft und *ref* der X/Y-Tabelle in der Abbildungsfunktion durch die entsprechenden *id* ersetzt. Dies erfolgt gleichermaßen für die D-Tabelle und Faktentabelle. Die letzte Abbildung enthält für alle Wertausprägungen von *d1* die entsprechende *id* der Filiale. Nicht selektierte Einträge werden auf *NULL* gesetzt und ignoriert. Wird diese Abbildung für die Aggregation der jeweiligen Umsätze genutzt, kann sehr effizient in einem Durchlauf der Umsatz pro Filiale ermittelt werden ohne temporäre Tabellen als Zwischenergebnisse vorhalten zu müssen. Diese endgültige Abbildung aller Wertausprägungen der Faktentabelle auf den gewünschten Wert (in diesem Fall Filiale) ist nur möglich bei einer Folge von $1 : n$ Abbildungen von der äußersten Tabelle zur Faktentabelle.

Die Suche nach häufigen Kombinationen, der laufzeitintensivste Anteil bei der Assoziationsregelsuche, arbeitet im BWA auf der Faktentabelle und benötigt hierfür eine Menge an Abbildungsfunktionen. Diese werden äquivalent zum Falle einer Aggregation erstellt, wodurch keine Modifikationen der bestehenden Implementierung nötig sind. Die finale Aggregationsoperation wird lediglich durch den Assoziationsregelsuchprozess ersetzt. Mit diesen Werkzeugen kann eine Aufbereitung der zu untersuchenden Daten direkt pro

Anfrage erfolgen und muss nicht wie bei einigen bestehenden Lösungen auf replizierten Daten durchgeführt werden.

Die Implementierung orientiert sich an dem aus Abschnitt 2.2.5 bekannten PARTITION-Algorithmus. Vorerst soll die EHK lediglich abstrakt betrachtet werden. Der Algorithmus besteht im Wesentlichen aus drei Schritten, (1) die parallele Suche lokal häufiger Kombinationen, (2) die Vereinigung lokaler Ergebnisse und (3) die Nachsuche fehlender Ergebnisse.

Die benötigte Selektion und Transformation der gewünschten Daten wird von der im BWA bestehenden Aggregationsoperation geerbt. Die jeweils zwischen den Schritten von PARTITION übertragenden Datenformate sind für alle implementierten EHK-Verfahren einheitlich. Damit ist der grundlegende Ablauf von PARTITION unabhängig von der spezifischen EHK und übernimmt lediglich die Verwaltung der verteilten Berechnung.

Die Vereinigung erfolgt durch das Einfügen der lokal auf einer Partition gefundenen Kombinationen in die globale Ergebnismenge. Diese Operation kann durch geschickte Implementierungen, wie beispielsweise Hash-Abbildungen, effizient erfolgen und nimmt damit selbst bei umfangreichen Teilergebnismengen meist nur einen geringen Teil der Gesamtlaufzeit in Anspruch. Dieselbe Operation erfolgt abschließend nochmals zum Bilden des endgültigen Ergebnisses.

Die Paarung aus EHK und Nachsuche ist hingegen spezifisch für die jeweiligen Algorithmen aufeinander abgestimmt. Die folgenden Abschnitte erläutern die Implementierungen dieser Verfahren im BWA. Begonnen wird mit dem Verfahren zur Bestimmung von Attributwertkorrelationen. Anschließend wird die Implementierung der Assoziationsregelsuche in Warenkörben vorgestellt.

5.3. Bestimmung von Attributwertkorrelationen

Bei der Bestimmung von Attributwertkorrelationen wird eine dem aus Abschnitt 2.2.4 bekannten BUC-Algorithmus ähnliche Implementierung verwendet. Die Problemstellung bei diesen Verfahren ist deutlich einfacher als eine vollständige Warenkorbanalyse, da lediglich Korrelationen zwischen verschiedenen Attributen gesucht werden. Es ist beispielsweise nicht möglich, Korrelationen zwischen zwei Farben eines Produktes zu finden, da dieses stets nur ein Attribut Farbe besitzt.

Eine Übertragung dieser Problemstellung auf eine Warenkorbanalyse ist möglich, indem jeweils eine Zeile der zu untersuchenden Tabelle auf eine Transaktion des Warenkorbes abgebildet wird. Dies bedeutet jedoch einen signifikanten, unnötigen Mehraufwand, da Informationen über den Ursprung eines Elementes (z.B. Farbe) nicht beachtet werden

und damit Korrelationen zwischen Farben gesucht werden. Der Vorteil einer reduzierten Komplexität soll genutzt werden, wodurch eine separate Implementierung zur Attributwertkorrelation realisiert wurde. Die Umkehrung, eine Abbildung der Warenkorbanalyse auf das Problem der Attributwertkorrelation, ist nicht möglich. Hierbei würde durch die Zuordnung eines Elementes zu einem Attribut eine für den Algorithmus relevante Einschränkung hinzugefügt, welche Korrelationen in den Daten durch Aufteilung auf mehrere Attribute entfernt.

Das Vorgehen bei diesem Algorithmus besteht im Wesentlichen aus der Bestimmung jeweils häufiger Attributausprägungen pro Attribut. Für jede Ausprägung wird eine Liste mit relevanten Zeilen erstellt. Für die einzelnen Ausprägungen wird die Häufigkeit der Überschneidungen der Zeilen bestimmt. Die Menge der verbleibenden Zeilen bildet die Häufigkeit der Kombination aus den miteinander geschnittenen Zeilenlisten der Elemente. Ausgehend von den Baumstrukturen aus Abschnitt 2.2.4 kann diese Häufigkeitsbestimmung rekursiv für aufeinander aufbauende Kombinationen erfolgen. Beispielsweise bildet die Überschneidung der Zeilen eines Datenbestandes mit *(Farbe=rot)* und der von *(Qualität=hoch)* die Zeilenliste für die Kombination *(Farbe=rot, Qualität=hoch)*.

Die hierfür nötigen Datenstrukturen sind denen im BWA verwendeten Tabellen sehr ähnlich. Das erleichtert die Vorverarbeitung und Aufbereitung der Daten. Lediglich eine Liste mit den für eine Kombination relevanten Zeilen wird zusätzlich benötigt. Die Suche kann nahezu vollständig auf den bereits im BWA vorhandenen Strukturen erfolgen. Dazu werden auf den in jedem Schritt gewählten Zeilen häufige Attributausprägungen weiterer Attribute ermittelt. Da diese Suchen im BWA sehr effizient durchführbar sind, werden keine speziellen Datenstrukturen benötigt. Neben dem Vorteil, dass diese Zugriffe bereits weitreichend optimiert sind, ergibt sich auch, dass die untersuchten Daten nicht repliziert werden müssen. Sie repräsentieren damit jeweils einen konsistenten, aktuellen Zustand der Daten.

Die Bestimmung häufiger Kombinationen kann außerdem für jeden Zweig des Suchraumes unabhängig erfolgen, da alle Geschwisterknoten jeweils nicht voneinander abhängig sind und entsprechend parallel analysiert werden können. Damit wird eine parallele Verarbeitung pro Rechenknoten ermöglicht, womit auch moderne Mehrkern-Prozessoren vollständig ausgelastet werden.

Der entstehende Speicherbedarf beschränkt sich neben die im BWA bereits vorhandenen Ausgangsdaten auf die Liste der relevanten Tabellenzeilen der jeweils aktuell untersuchen Kombinationen. Diese Listen sind bei Verkaufsdaten um Größenordnungen kleiner als die gesamten Ausgangsdaten und daher in der Praxis wenig relevant. Der Großteil der zur Analyse benötigten Informationen kann direkt aus den Datenstrukturen des BWA gewonnen werden. Hinzu kommt der Speicherbedarf zur Ablage der gefundenen Kombinationen und deren Häufigkeit.

Transaktion			Element	Zusatzinformation		
Ort	**Nr.**	**Datum**	**Produkt**	**Preis**	**Währung**	**Verkäufer**
Dresden	1	4.2. 2009	Windeln	3,50	EUR	Heinz Müller
Dresden	1	4.2. 2009	Bier	1,35	EUR	Heinz Müller
Walldorf	8	4.2. 2009	Windeln	3,10	EUR	Anna May
Dresden	1	4.2. 2009	Nudeln	3,50	EUR	Heinz Müller
Dresden	1	4.2. 2009	Kaugummi	0,85	EUR	Heinz Müller
Dresden	2	4.2. 2009	Eier	0,80	EUR	Helga Kunze

Tabelle 5.1.: Beispiel: Ausschnitt aus Tabelle mit Verkaufsdaten

5.4. Durchführung einer Warenkorbanalyse

Das derzeit für den BWA bevorzugt nachgefragte Szenario ist die Warenkorbanalyse, eine Assoziationsregelsuche auf Warenkörben. Dabei liegen die zu untersuchenden Daten bereits in Form eines SAP BI InfoCubes mit Verkaufsdaten für Analysezwecke im BWA vor. Eine einzelne Transaktion bzw. Warenkorb wird auf je eine Menge von Zeilen der Faktentabelle des InfoCubes abgebildet. Zusätzlich wird zu jeder dieser Zeilen die für InfoCubes üblichen Zusatzattribute und Kennzahlen, wie beispielsweise Menge oder Preis, gespeichert. Tabelle 5.1 zeigt dies an einem Beispiel. Besonders zu beachten sind hierbei die nicht zwangsläufig gemeinsam gespeicherten Einträge einer Transaktion und die mehrfach redundante Ablage der Zusatzinformationen.

5.4.1. Datenaufbereitung

Die Reihenfolge der einzelnen Zeilen ist nicht vorgegeben. Es wird im Folgenden lediglich gefordert, dass alle Elemente einer Transaktion auf derselben Tabellenpartition vorliegen. Da Transaktionen zwar vollständig auf einer Partition, jedoch nicht im Zusammenhang gespeichert sind, wird für eine effiziente Suche nach häufigen Kombinationen eine Aufbereitung der Daten notwendig.

Dafür werden beginnend auf einer nutzerdefinierten Selektion, äquivalent zu der Selektion bei der Aggregation im BWA, die Menge häufiger Elemente bestimmt. Nach der Downward-Closure-Eigenschaft können nur Kombinationen aus diesen Elementen häufig sein. Auf diesen entstehenden, meist deutlich reduzierten Datenbeständen erfolgt eine Transformation in eine für die Bestimmung häufiger Kombinationen optimierte Datenstruktur. Diese wird nachfolgend erläutert.

Vorerst werden die relevanten Transaktionen der Faktentabelle extrahiert und deren Elemente (ItemID) und Transaktions-ID (TID) ermittelt. Durch die im BWA verwendete Wörterbuchkodierung liegen alle diese ID-Werte in einem Wertebereich von $0 \leq id \leq$

$maxID$ vor, wobei $maxID$ jeweils die Menge der verschiedenen Wertausprägungen des betrachteten Attributes annimmt. Dieser Wertebereich wird nicht zwingend vollständig ausgenutzt, da durch Selektion und Filterung seltene Wertausprägungen nicht für die Suche relevant sind. Trotzdem bewegt sich dieser Bereich in Größenordnungen, welche die direkte Abbildung in spezifische, effiziente Datenstrukturen ermöglichen und damit komplexe Zugriffsverfahren wie Hash-Tabellen nicht benötigen. Um beispielsweise die Häufigkeit von zehn Elementen zu bestimmen, kann eine Feldstruktur mit zehn Zählern genutzt werden. Der für ein Element gültige Zähler ergibt sich dabei direkt aus der ItemID, was einen effizienten Zugriff erlaubt.

Für den Fall mehrdimensionaler Transaktions-IDs, beispielsweise *Ort*, *Zeit* und *Filiale*, müssen die Kombination der betroffenen Attribute in ihren jeweiligen Ausprägungen vereinigt und auf eine eindeutige ID eines ganzzahligen Bereiches von $0 \leq id \leq maxID$ abgebildet werden. Der Wert $maxID$ soll möglichst klein sein, muss aber die eindeutige Abbildung der Menge aller auftretenden Transaktions-IDs auf eine natürliche Zahl ermöglichen. Diese Abbildung ist jeweils abhängig von den selektierten Daten und benötigt bei großen Datenbeständen oft einen signifikanten Anteil der Gesamtlaufzeit der EHK. Insbesondere bei sehr eingeschränkten Suchen häufiger Kombinationen, beispielsweise nach Kombinationen mit drei Elementen als Maximallänge, kann dieser Arbeitsschritt mehr Zeit beanspruchen als die eigentliche EHK. Durch eine entsprechende Vorberechnung oder durch Nutzung der Kompressionstrategien des BWA kann dieser Vorgang optimiert werden.

Die für die EHK verwendete Datenstruktur, bestehend aus Tupeln der Form (TID, Element), wird aufsteigend sortiert. Dank der bereits erläuterten Wörterbuchkodierung müssen an dieser Stelle keine Sortierverfahren mit einer Laufzeitkomplexität $\mathcal{O}(n \cdot \log n)$ verwendet werden. Andere Algorithmen, wie beispielsweise das Verfahren Pigeonhole-Sortierung [IS56], weisen für derartige Daten lediglich eine lineare Komplexität von $\mathcal{O}(|E| + o)$ bei einer Menge von betrachteten Elementen E und o Einträgen auf. Ist ein solches Vorgehen aus Speichermangel nicht möglich, muss auf andere Sortierverfahren mit logarithmischem Laufzeitverhalten zurückgegriffen werden. Um diesen teilweise sehr rechenaufwändigen Arbeitsschritt zu beschleunigen, können auch Cell-Prozessoren [GBY07] oder die Recheneinheiten von Grafikkarten [GGKM06; GRM05] genutzt werden.

Mit Hilfe dieser sortierten Struktur kann die für die EHK relevanten Transaktionen effizient analysiert und aufbereitet werden. Hierfür ist mit linearem Zeitaufwand eine Entfernung von doppelt auftretenden Elementen möglich. Außerdem können Transaktionen, welche für die gewünschten Kombinationen keine Informationen enthalten, entfernt werden. Ist beispielsweise die Mindestlänge l einer Kombination definiert, können alle Transaktionen mit weniger als l Elementen aus der Struktur entfernt werden. Ähnlich verhält es sich mit Transaktionen bestehend aus nur einem Element. Diese sind für die Häufigkeit der Einzelelemente noch relevant, können aber keine Kombinationen X mit $|X| > 1$ erzeugen. Aus der Datenbasis zur Ermittlung mehrelementiger Kombinationen können diese Transaktionen entfernt werden.

Nach dieser Aufbereitung werden zur optimierten Abarbeitung der Bestimmung häufiger Kombinationen die noch vorhandenen Elemente nach ihrer Häufigkeit sortiert. Seltene Elemente können damit schrittweise um Elemente mit aufsteigender Häufigkeit erweitert werden. Wie in 2.2.3 erläutert, wird mit dieser Heuristik versucht, die Anzahl der insgesamt betrachteten Transaktionen zu minimieren. Diese Sortierung ist auch bereits zu einem früheren Zeitpunkt möglich. Durch die Bereinigung der Daten ändern sich jedoch diese Elementhäufigkeiten teilweise signifikant. Realitätsnahe Daten besitzen oft einzelne Elemente, welche sehr häufig sind aber oft nur allein in einer Transaktion auftreten. Daher ist eine Umsortierung nach dem Entfernen der Einträge sinnvoll, wobei die Elemente auf ihren Rang im Bezug auf ihre Häufigkeit abgebildet werden. Angelehnt an das Beispiel aus Abbildung 3.1 ergibt sich Tabelle 5.2.

Element	D	C	B	A
ItemID	1	2	3	4

Tabelle 5.2.: Datenstruktur Element-Abbildungen

Tabelle 5.3 zeigt einen Ausschnitt aus der dabei entstehenden Datenstrukturen.

TID	1	1	1	1	2	3	4	5	6	7	7	7	8	9	9	9	...
ItemID	1	2	3	4	2	1	1	1	1	1	2	3	1	2	3	4	...

Tabelle 5.3.: Ausgangsdatenstruktur Warenkorbanalyse

Diese Struktur lässt sich in ein komprimiertes, vertikales Datenformat transformieren. Wie in Abbildung 5.4 gezeigt, werden dabei zwei Strukturen gebildet: (1) eine Abbildung der Transaktions-ID auf eine Liste mit deren Elementen und (2) eine Abbildung von Element-IDs auf eine Liste mit den Transaktionen, welche dieses Element enthalten. Damit wird ermöglicht, sowohl aus Sicht der Elemente als auch aus Transaktionssicht schnell auf die jeweils zugehörigen Daten zuzugreifen. Der Berechnungsaufwand erreicht in beiden Fällen bei o Einträgen eine lineare Laufzeitkomplexität von $\mathcal{O}(o)$.

TID	1				2	3	4	5	6	7		8	9			...	(1)
ItemID	1	2	3	4	2	1	1	1	1	1	2	3	1	2	3	4	...	

ItemID	1								2							...	(2)
TID	1	3	4	5	6	7	8	11	1	2	7	9	10	11	12	...	

Tabelle 5.4.: Analysedatenstrukturen Warenkorbanalyse

Nach dieser Transformation wird die Struktur in Tabelle 5.3 nicht länger benötigt, da sie durch die Darstellungen in 5.4 repräsentiert wird. Die Transformationsregeln aus Tabelle

5.2 müssen jedoch erhalten bleiben, um nach der Bestimmung häufiger Kombinationen die Rückformung auf die Originaldaten zu ermöglichen.

Der zur EHK nötige Speicherbedarf durch die Strukturen in 5.4 beträgt insgesamt etwa doppelt soviele nummerische Werte wie ausgewählte Zeilen, da die Daten jeweils nach Element und Transaktionsschlüssel sortiert vorliegen. Hinzu kommt die Ablage der während der EHK extrahierten Kombinationen. Im Rahmen der Vorverarbeitung kann dieser Bedarf, vor allem bei Sortierung mit linearer Laufzeit, kurzzeitig noch erhöht werden. Dabei wird je nach verfügbaren Ressourcen zwischen Algorithmen mit minimalem Speicherverbrauch oder minimaler Laufzeit gewählt.

5.4.2. Bestimmung lokal häufiger Kombinationen

Die Suche nach häufigen Kombinationen erfolgt auf den im vorherigen Abschnitt vorgestellten Datenstrukturen gemäß dem Algorithmus 3.1. Die Datenstruktur in Tabelle 5.4 (1) wird verwendet, um den Start der Suche von Elementkombinationen zu beschleunigen. Dieser Schritt inklusive der Datenstruktur kann entfallen, wenn pro zu untersuchendem Startknoten die Liste mit relevanten Transaktions-IDs aus den verbleibenden Strukturen extrahiert wird. Der Speicheraufwand der verwendeten Datenstrukturen wird dabei etwa halbiert, pro Startelement muss aber eine Suche in den Daten erfolgen. Für praxisnahe Anwendungen mit $|I| > 10.000$ ist die Vorberechnung und der entsprechend erhöhte Speicherverbrauch meist die sinnvollere Alternative.

Für die Bestimmung der Häufigkeit von 1-Kombinationen sind die Transaktionslisten nicht nötig. Hierbei sind die aus der Vorverarbeitung ermittelten Häufigkeiten ausreichend. Die erste Ebene des Suchraumes in Abbildung 3.1 ist somit bereits implizit ermittelt. Die Transaktionslisten werden lediglich für tiefere Ebenen mit Kombinationen von mehr als einem Element benötigt, wodurch Transaktionen mit nur einem Element nicht vorgehalten werden müssen.

Ausgehend von der Transaktionsliste für jedes häufige Element erfolgt rekursiv die Durchsuchung des entsprechenden Teilbaumes. Hierbei werden die Häufigkeiten aller Kinder des Startknotens gleichzeitig bestimmt. Dies ist effizient möglich, indem über die Liste mit Transaktionen iteriert wird und deren Elemente absteigend bis zu dem jeweiligen Elternelement gezählt werden. Nach diesem Schritt können die noch häufigen Elemente in Abhängigkeit vom Elternelement ermittelt und mit demselben Verfahren die exakte Transaktionsliste für jeden Kindknoten bestimmt werden. Die Zuordnung der für einen Kindknoten relevanten Transaktions-IDs kann auch direkt erfolgen und für jeden Kindknoten repräsentiert die Größe der Transaktionsliste zugleich die Häufigkeit dieses Knotens. Evaluationen haben ergeben, dass meist nur ein kleiner Teil der Kindknoten die Mindesthäufigkeit erreichen. Die Kosten der Ablage der Transaktions-IDs für alle

Elemente ist damit signifikant höher als zwei separate Durchläufe zum Zählen der Häufigkeiten und dem Bestimmen der Transaktionslisten pro häufigem Element.

Zwischen diesen beiden Schritten erfolgt auch ein Großteil der Erkennung von geschlossenen bzw. unscharf-geschlossenen Kombinationen. Alle Kindknoten, welche zu ihrem Elternknoten geschlossen sind, werden mit diesem kombiniert und von der weiteren Untersuchung ausgeschlossen. Bezogen auf Abbildung 3.6 werden geschlossene Knoten eine Ebene in Richtung Wurzelknoten verschoben.

Außerdem kann nach dem Zählen im Falle des Erreichens der maximalen Kombinationslänge die Ermittlung der Transaktionslisten übersprungen werden. Diese werden nicht für weitere Analysen benötigt, da keine weiteren Elemente zu den Kombinationen hinzugefügt werden. Die Kombinationen mit maximal möglicher Länge können bereits nach der Häufigkeitsbestimmung gespeichert werden ohne ihre Transaktionen separat zu bestimmen. Sind derartige Optimierungen nicht möglich, erfolgt für alle auf diesen Knoten häufige Elemente ausgehend von deren Transaktionslisten eine rekursive Untersuchung der entsprechenden Teilbäume.

5.5. Nachsuchen fehlender Kombinationshäufigkeiten

Werden bei der verteilten Suche nach häufigen Kombinationen nicht auf allen Partitionen dieselben Kombinationen ermittelt, müssen nach der Vereinigung der lokalen Teilergebnisse die fehlenden Teilhäufigkeiten einiger Kombinationen nachträglich ermittelt werden.

Diese Nachsuche fehlender Häufigkeiten erfolgt sowohl bei der Bestimmung von Attributwertkorrelationen als auch bei der Warenkorbanalyse ähnlich zu der jeweiligen EHK-Implementierung. Die Vorverarbeitung der Daten beschränkt sich hierbei lediglich jeweils auf alle in den nachzusuchenden Kombinationen enthaltenen Elementen ohne eine geforderte Mindesthäufigkeit. Die Datenstrukturen der EHK können nicht wiederverwendet werden, da sie seltene Elemente nicht enthalten. Wie auch die Extraktion der häufigen Kombinationen erfolgt die Nachsuche pro Partition auf dem dazugehörigen Rechenknoten.

Während für Attributwertkorrelationen keine aufwändigen Vorverarbeitungen benötigt werden, erfolgt für die Daten der Nachsuche der Warenkorbanalyse der Aufbau der gleichen Datenstrukturen wie in Abschnitt 5.4.1. Die Bestimmung der jeweilig ausstehenden Häufigkeit einer Kombination erfolgt einzeln für jedes Element der Kombination. Weisen nacheinander untersuchte Kombinationen äquivalente Teilkombinationen auf, werden die Teilergebnisse der vorherigen Kombination genutzt. Es ist somit sinnvoll, die nachzusuchenden Kombinationen derartig zu sortieren, dass möglichst viele Teilkombinationen mehrfach verwendet werden können. In dieser Arbeit werden die Elemente der

Transaktionen nach ihrer Häufigkeit sortiert, was nicht zwangsläufig der optimalen Wiederverwendbarkeit entspricht. Bei den untersuchten Analysen auf Verkaufsdaten war die Laufzeit der Nachsuchen jedoch meist unbedeutend gegenüber den lokalen EHK. Die entsprechende Optimierung war somit nicht erforderlich.

5.6. Mehrdimensionale Regeldarstellung

Basierend auf den im Abschnitt 2.4.2 vorgestellten Verfahren wird im Rahmen dieser Arbeit vorgeschlagen, die jeweiligen Vorteile der bereits gezeigten Darstellungsmöglichkeiten zu vereinigen. Dies umfasst die Verwendung von mehreren Interessantheitsmaßen, Regelgruppierungen und syntaktischen Aufbereitungen wie Regelgraphen oder Mosaikdarstellungen. Durch die Kombination dieser Vielzahl an Möglichkeiten soll eine effiziente Auswahl und Fokussierung auf wertvolle Regeln innerhalb einer größeren Regelmenge ermöglicht werden.

Dafür wurde im Rahmen dieser Arbeit eine grafische Oberfläche entwickelt, welche gleichzeitig eine Auswahl verschiedener Repräsentationsformen der Regeln darstellt. Innerhalb jeder Darstellung können potenziell interessante Regeln markiert werden. Diese werden in allen weiteren Ansichten hervorgehoben. Auf diese Weise ist es möglich, eine Regelmenge effizient auf eine deutlich geringere Menge an potenziell wertvollen Regeln zu reduzieren. Diese genügt durch die in mehreren Ansichten durchgeführte Auswahl der Regeln jeweils mehreren Auswahlkriterien des Anwenders.

Die mehrdimensionale Darstellung stellt die Ergebnisse der Assoziationsregelanalyse dem Betrachter im Überblick dar. Auffälligkeiten einer Sichtweise können effizient in anderen Sichten betrachtet werden. Weist eine Gruppe von Regeln aus einer Sicht in anderen Darstellungen Besonderheiten auf, kann die ursprüngliche Regelgruppe durch neue Einschränkungen verkleinert werden. Dieses Vorgehen ist ohne die Beachtung der Regelsyntax auf den semantischen Eigenschaften, den Interessantheitsmaßen der Regeln, realisierbar. Durch diese Einschränkung der Regelmenge auf Regeln mit spezifischen Interessantheitswerten kann ein Regelauswerteprozess vereinfacht werden.

Ist die Darstellung der durch semantische Einschränkungen reduzierten Regelmenge in einer syntaktischen Repräsentation darstellbar, kann zusätzlich zu den semantischen Darstellungen eine syntaktische Aufbereitung durch Graphen erfolgen. Damit sind Gruppen von ähnlichen Regeln und auffällige Kombinationen effizient optisch erkennbar. Der Anwender kann die Regeln in Form von ersichtlichen Regelgruppen auswerten. Das ermöglicht eine effiziente Filterung. Schränkt der Nutzer mit Hilfe dieser Darstellung seine Auswahl weiter ein, kann sie auf wenige Regeln verringert werden. Auf den verbleibenden Ergebnissen einer solchen Gruppe kann schließlich eine umfassende Untersuchung der Regeln auf deren praktischen, subjektiven Wert für den Nutzer erfolgen.

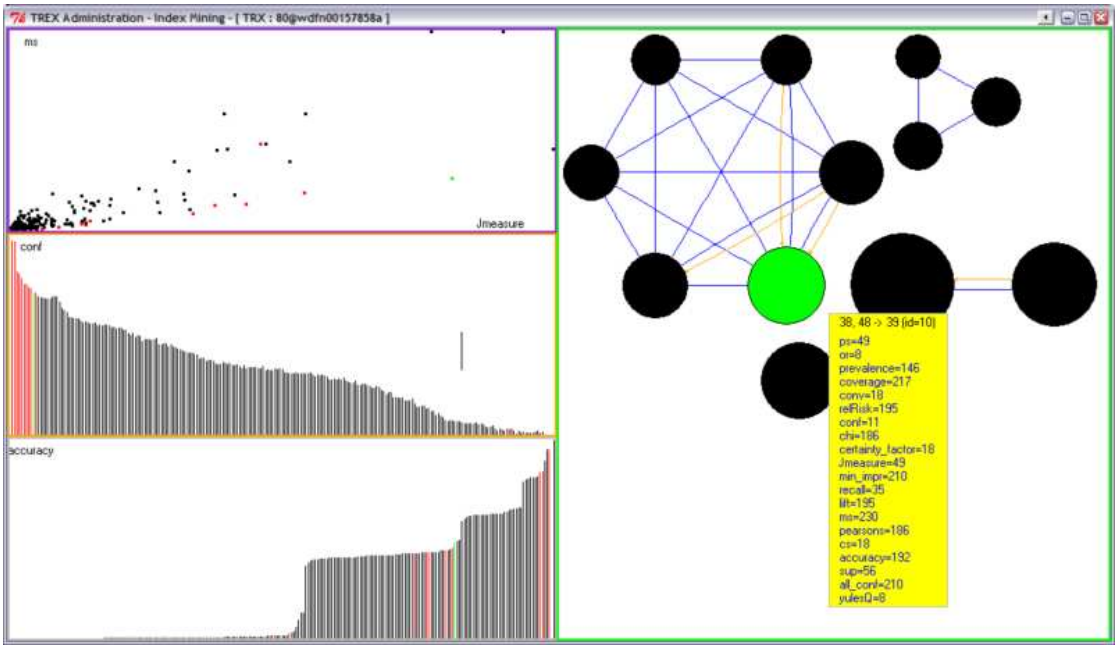


Abbildung 5.4.: Beispiel: Mehrdimensionale Regelrepräsentation

Abbildung 5.4 zeigt eine im Rahmen dieser Arbeit erstellte Beispielimplementierung des Verfahrens mit vier gleichzeitig dargestellten Sichtweisen. Die linke Hälfte zeigt ein Streudiagramm mit den Interessantheitsmaßen *Mutual Support* und *J-Measure* und zwei Säulendiagramme für die Maße *Konfidenz* und *Accuracy*. Beide Darstellungen sind jeweils sortiert nach deren Wertausprägung. In den Steudiagrammen bilden die einzelnen Regeln die X-Achse. Die farblich hervorgehobenen Einträge zeigen in jedem Diagramm die markierten Regeln.

In der rechten Hälfte wird gleichzeitig eine syntaktische Darstellung gezeigt, sobald die gewählte Regelmenge klein genug für eine sinnvolle Darstellung ist. Die blauen Kanten zeigen gemeinsame Bedingungen zwischen den Regeln und orangen Kanten gemeinsame Schlussfolgerungen. Der grüne Knoten stellt die jeweils aktuelle Auswahl einer einzelnen Regel dar. Für diese werden alle verfügbaren semantischen Informationen für den Nutzer aufbereitet. Dieses Beispiel zeigt ein typisches realistisches Szenario. Interessantheitsmaße sind meist relativ ungleichmäßig auf ihren Wertebereich verteilt und oft bestehen die Regeln innerhalb eines markierten Bereiches aus nur wenigen unterschiedlichen Kombinationen.

5.7. Evaluation

Die folgenden Abschnitte untersuchen, wie sich die Laufzeit der vorgestellten Implementierungen der Assoziationsregelanalyse im Bezug auf verschiedene Datenbestände und Parameter verhält. Dies zeigt die Fähigkeit dieser Lösung, im Rahmen des SAP BW Accelerator große Mengen Verkaufsdaten zu analysieren. Begonnen wird mit der Extraktion häufiger Kombinationen in Form der Attributwertkorrelation. Anschließend wird die Implementierung der Warenkorbanalyse untersucht. Dabei wird zum einen die Fähigkeit der Top-N-Suche im Vergleich zu einer regulären EHK betrachtet, sowie die Skalierung der Verfahren auf partitionierten Daten. Beides stellt eine wichtige Anforderung für die Verwendung in praxisnahen Szenarien dar.

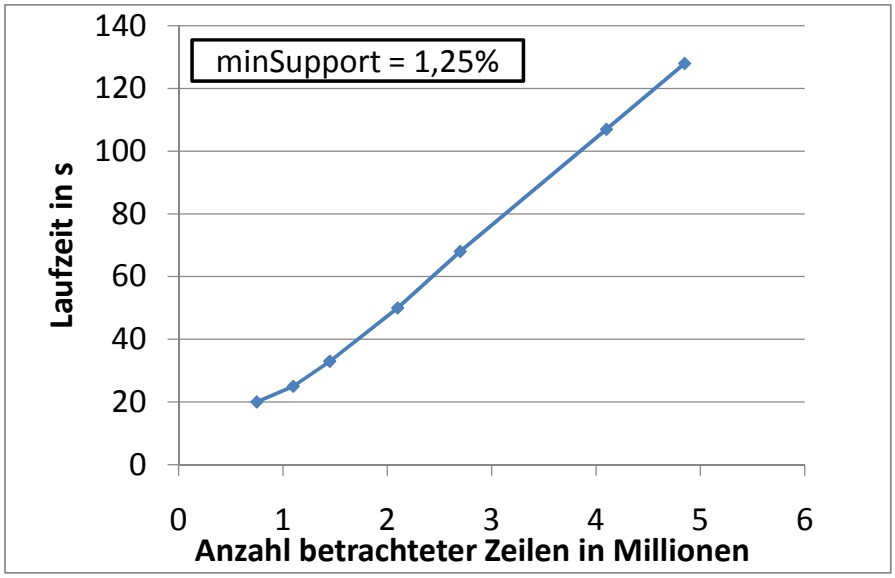
5.7.1. Attributwertkorrelation mit BUC

Die Evaluation zur Attributwertkorrelation und Verwaltung verteilter Datenbestände erfolgte auf bis zu elf Blade-Servern mit je zwei 32-Bit Intel Xeon 3.2 GHz Prozessoren und vier GB Hauptspeicher verbunden über Gigabit-Ethernet. Als Betriebssystem dient Microsoft Windows Server 2003 Enterprise Edition SP1. Sämtliche Algorithmen wurden in der Programmiersprache C++ in die Architektur des SAP BWA integriert.

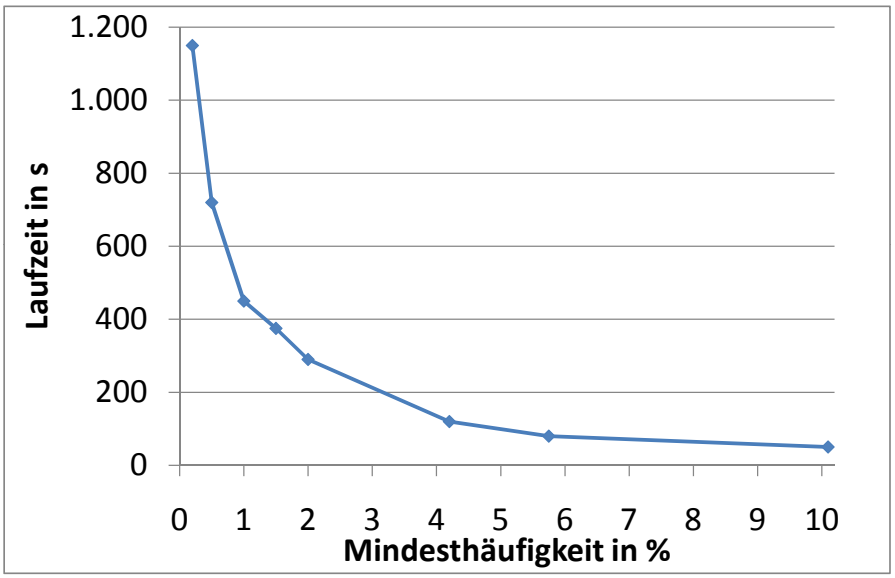
Bei den realistischen Kundendaten wurden Dimensionsschlüssel mit je 5 bis 100.000 Wertausprägungen auf Korrelationen untersucht. Abbildung A.2 im Anhang A zeigt die Skalierung der Ergebnismenge mit sinkender Mindesthäufigkeit für die in dieser Arbeit verwendeten Datenbestände.

Die Laufzeit von BUC mit steigender Datenmenge wird in Abbildung 5.5(a) dargestellt. Die Daten sind in dieser Untersuchung nicht partitioniert und verwenden entsprechend lediglich einen Blade-Server. Die Ergebnisse zeigen eine lineare Skalierung mit der Datenmenge. Dies ist eine Konsequenz aus einer Eigenschaft des BWA, welcher Selektionen jeweils mit dem Durchsuchen des vollständigen Attributes realisiert, ohne weitere Baum- oder Indexstrukturen zu nutzen. Die Laufzeit zum Suchen von Attributausprägungen wächst somit weitestgehend linear mit der Anzahl der untersuchenden Zeilen.

Abbildung 5.5(b) zeigt die Laufzeiten für realistische Kundendaten mit 180 Millionen Zeilen verteilt auf elf Partitionen mit einer Größe von je etwa 430 MB. Die Ergebnisse zeigen Laufzeiten, welche spontane Analysen mit realistischen Datenmengen in wenigen Minuten erlauben. Derartige Analyse benötigen derzeit laut Aussage von SAP-Kunden teilweise mehrere Stunden, was die Nutzbarkeit dieser Lösung für Analysen ohne aufwändige Vorverarbeitungen im SAP BWA unterstreicht.



(a) Laufzeit Mushroom3



(b) Laufzeiten KundeC

Abbildung 5.5.: Laufzeitskalierung bei veränderter Datenmenge und Mindesthäufigkeit

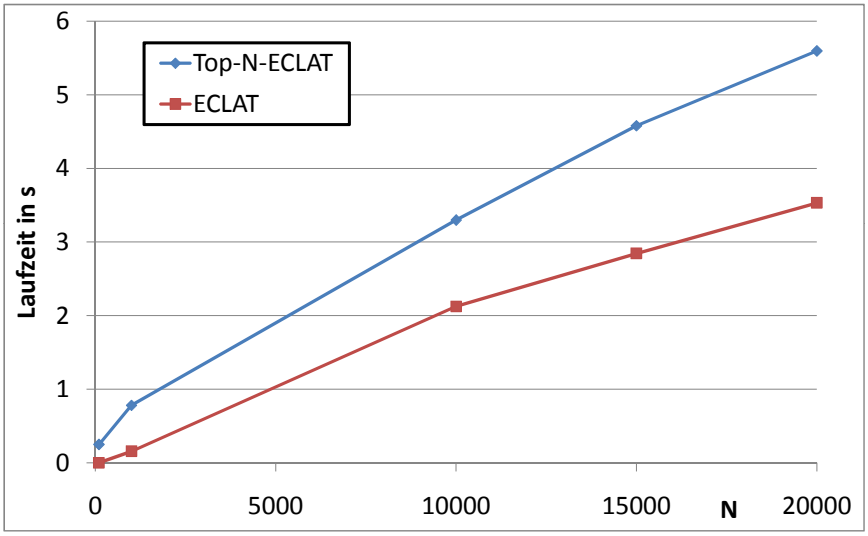


Abbildung 5.6.: Vergleicht Laufzeit Top-N vs. Direkt

5.7.2. Warenkorbanalyse und Top-N

Der folgende Abschnitt evaluiert die Implementierung der Warenkorbanalyse. In einem ersten Schritt wird der Unterschied zwischen einer mindesthäufigkeitsbasierten Lösung und einer Top-N-Implementierung untersucht. Verwendet wurde hierfür der Anhang vorgestellte SynthC-Datenbestand mit 980.000 Transaktionen in 10 Millionen Zeilen. Abbildung 5.6 zeigt einen Vergleich der Laufzeiten für die Suche von Top-N-Kombinationen in Abhängigkeit von N. Verglichen wird die Laufzeit der Top-N-Implementierung gegen eine Suche mit Hilfe einer nutzerdefinierten Mindesthäufigkeit entsprechend der N-ten Kombination. Diese Mindesthäufigkeit wird in diesem Vergleich als bekannt angenommen, was in der Realität nur selten möglich ist. Sie muss da durch Erfahrung des Nutzers oder eine iterative Annäherung des Wertes bestimmt werden, was viel Zeit in Anspruch nehmen kann. Die Differenz in der Laufzeit zwischen den beiden Graphen zeigen die entstehenden Kosten für die Nutzung von Top-N-Strategien. Der Gewinn an Nutzerfreundlichkeit und die Kosten zur Bestimmung der korrekten Mindesthäufigkeit sind nicht allgemein spezifizierbar und werden daher in dieser Auswertung nicht betrachtet.

Für kleine N kann ein deutlicher Mehraufwand von teilweise über 100% beobachtet werden. Für $N > 7.500$ beträgt der Mehraufwand jeweils etwa 50%, was im Wesentlichen den Beobachtungen bei ähnlichen Verfahren entspricht [ZH99; PHM00]. Die exakten Kosten sind hierbei abhängig von den untersuchten Daten. Die Skalierung des Mehraufwandes wurde aber in ähnlicher Form auf mehreren Datenbeständen beobachtet.

Hinzu kommt, dass bei einer Top-N-Suche der Datenbestand nicht auf final häufige Elemente eingeschränkt werden kann und vollständig betrachtet werden muss. Dieser entste-

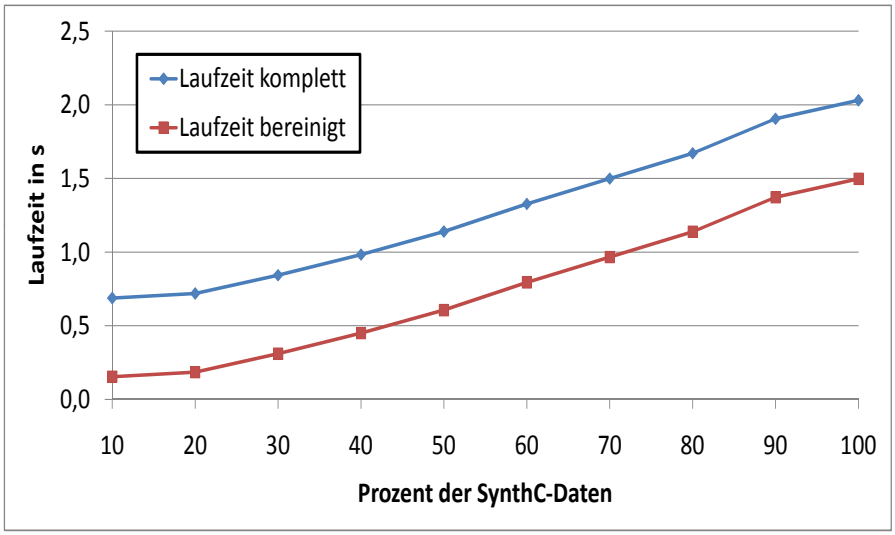
hende Überschuss an Daten, welcher aber zu Beginn der Suche noch nicht erkennbar ist, erzeugt bei der Aufbereitung der Daten einen signifikanten Mehrverbrauch an Speicher und Rechenleistung. Dieser Nachteil tritt jedoch ebenfalls nur bei sehr kleinen N und entsprechend hohem manuell definiertem *minSupport* auf, da nur hier eine signifikante Teilmenge aller Einzelelemente nicht häufig ist und entfernt werden kann. Für große N ist der Aufwand zur Aufbereitung der Daten für beide Verfahren ähnlich.

Die Suche häufiger Kombinationen mit Hilfe der Top-N-Implementierung ist somit sinnvoll, sobald die entsprechende Mindesthäufigkeit nicht bereits bekannt ist. Insbesondere für unbekannte Daten ist eine direkte Wahl der Mindesthäufigkeit schwierig, da hierbei extrem viele Ergebnisse entstehen können. Diese werden vom Nutzer nicht erwartet, benötigen sehr viele Systemressourcen und erzeugen große Algorithmenlaufzeiten. Spontane Analysen ohne die Hilfe eines Experten sind daher oft nur bedingt möglich und können produktive Systeme in ihrer Funktionalität durch ungewollt komplexe Anfragen beeinträchtigen. Einige Nutzer vermeiden aus diesem Grund deren Durchführung. Die Top-N-Strategien sollte somit vor allem von unerfahrenen Anwendern trotz höherem Ressourcenverbrauch bevorzugt verwendet werden.

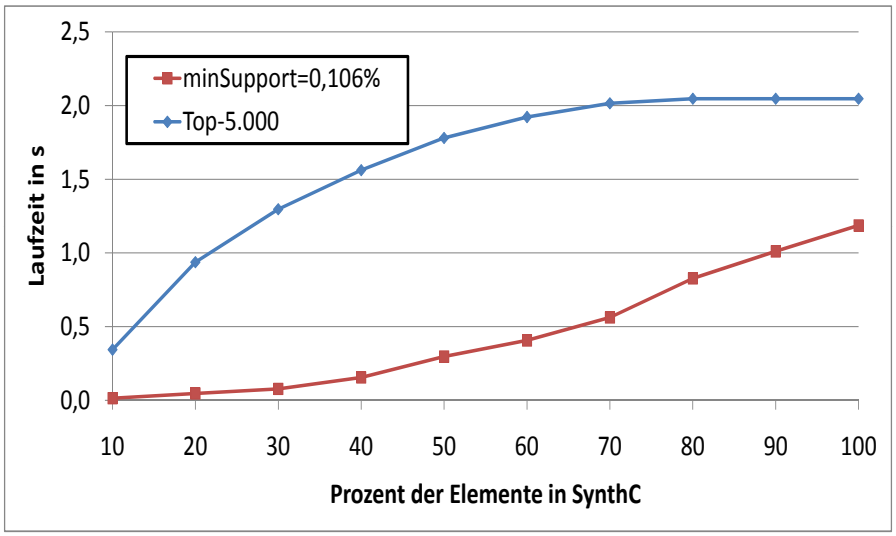
Abbildung 5.7(b) zeigt die Laufzeitskalierung in Abhängigkeit von der zu untersuchenden Datenmenge. Der obere Graph repräsentiert dabei die Laufzeit einer Top-5.000 Anfrage auf dem SynthC-Datenbestand inklusive der Ablage und Verwaltung der Ergebnisse. Der untere Graph ist um diesen konstanten Anteil bereinigt und zeigt die Laufzeit zur Bestimmung der Häufigkeiten. Hierbei kann eine lineare Laufzeitskalierung beobachtet werden.

Hingegen wird in Abbildung 5.7(a) die Skalierung über die Menge der betrachteten Elemente gezeigt. Verglichen wird hierbei für den komplette SynthC-Datenbestand eine Top-5.000 Suche und ihr Gegenstück mit direkter Wahl der entsprechenden Mindesthäufigkeit. Dabei kann gezeigt werden, dass der Mehraufwand einer Top-N-Suche mit steigender Anzahl verfügbarer Elemente in Relation zur Menge der Elemente abnimmt.

Insgesamt kann die Schlussfolgerung gezogen werden, dass eine Warenkorbanalyse für große Datenbestände mit Hilfe des BWA und dessen Ressourcen effizient möglich ist. Außerdem hat sich in der Praxis gezeigt, dass die Nutzung von Top-N-Strategien für den Nutzer gegenüber der schwierigen Wahl von Mindesthäufigkeiten einen signifikanten Vorteil bietet. Das für ungeübte Nutzer unerwartete und schwer zu handhabende Verhalten einer Assoziationsregelanalyse durch schwierige Parameter kann durch Top-N-Suchen mit vertretbaren Kosten sinnvoll vereinfacht werden.



(a) Abhängigkeit von Elementanzahl



(b) Abhängigkeit von der Datenmenge

Abbildung 5.7.: Laufzeitskalierung Warenkorbanalyse

5.7.3. Verteilung mit PARTITION

Die Testergebnisse in Abbildung 5.8 zeigen die Laufzeit der einzelnen Schritte einer verteilten Suche nach häufigen Kombinationen. Während Abbildung 5.8(a) die Messungen für eine Top-20.000-Suche auf dem SynthC-Datenbestand zeigt, wird in Abbildung 5.8(b) die Messung für eine Top-100.000 Suche präsentiert.

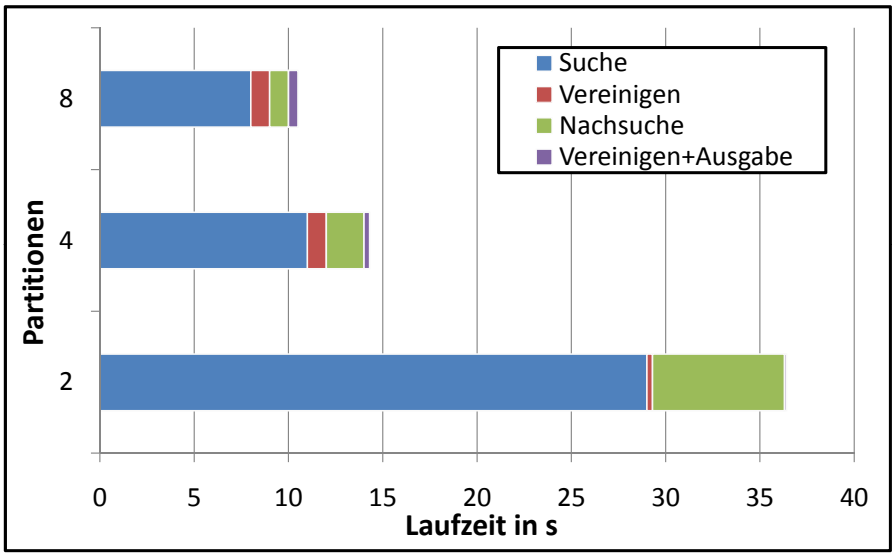
Eine wichtige Erkenntnis aus diesen Darstellungen ist die nicht erreichte lineare Skalierung mit wachsender Partitionierung. Ähnliches Verhalten solcher Algorithmen wurde bereits in anderen Arbeiten beobachtet [CX98; Hid99]. In dieser Darstellung muss außerdem beachtet werden, dass die Laufzeit von zwei auf vier Partitionen überproportional abnimmt. Das ist darin begründet, dass unter *Suche* der komplette Prozess einer lokalen Suche häufiger Kombinationen betrachtet wird. Das schließt den Aufbau der benötigten Datenstrukturen ein. Diese profitieren von kleineren Speicherbereichen und erweiterten Rechenressourcen, wie z.B. Cache pro Rechenknoten. Der Laufzeitanteil zur Bestimmung der lokal häufigen Kombinationen basierend auf der erzeugten Datenstruktur skalierte bei dieser Messung etwa linear.

Gut erkennbar ist ebenfalls, dass die Kosten zur Bestimmung der lokalen Teilergebnisse die Gesamtlaufzeit dominieren. Während die Vereinigung der Teilergebnisse und das Aufbauen der Ergebnisstrukturen vernachlässigbar sind, nimmt lediglich das Nachsuchen fehlender Häufigkeiten noch einen relevanten Laufzeitanteil ein.

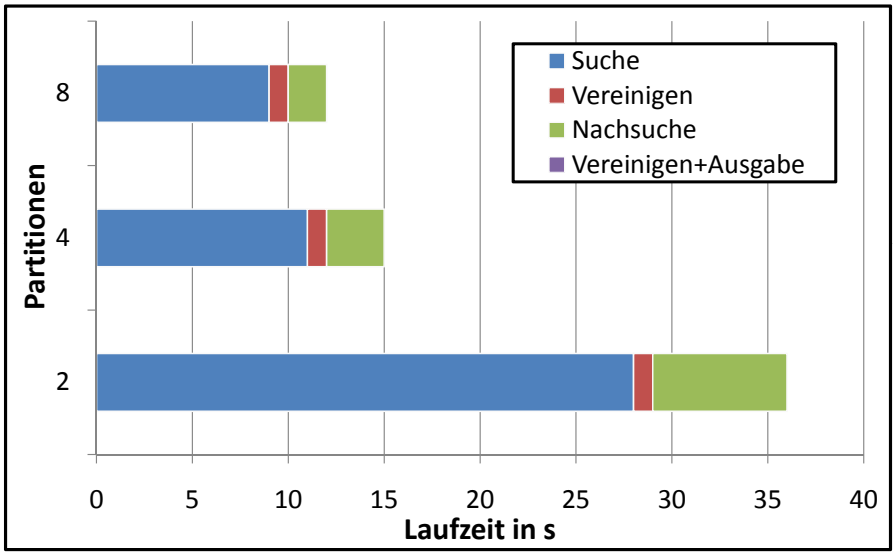
Aus diesen Untersuchungen kann die Erkenntnis abgeleitet werden, dass eine verteilte Suche nach häufigen Kombinationen auf stark verteilten Datenbeständen vermieden werden sollte. Realistische Geschäftsszenarien nutzen Datenverteilung jedoch auch zur Beschleunigung der Berechnung von Geschäftsanalysen. Diese nutzen größtenteils verteilt leicht berechenbare Aggregationsoperationen, welche nahezu mit der Anzahl der Partitionen skalieren. Soll zusätzlich, der Aggregation untergeordnet, eine Assoziationsregelanalyse auf diesen Daten erfolgen, ist auch eine Verarbeitung stark partitionierter Datenbeständen erforderlich und die dabei entstehenden Kosten und ein schlechtes Laufzeitverhalten müssen akzeptiert und behandelt werden.

5.8. Zusammenfassung

Die in dieser Arbeit vorgestellten Verfahren wurden prototypisch in die Plattform des SAP Business Warehouse Accelerator (BWA) der Firma SAP implementiert. Um die effiziente Verarbeitung der kompletten Daten durchführen zu können, werden alle benötigten Informationen im Hauptspeicher gehalten. Der hierfür nötige Speicherbedarf kann durch parallele Datenverarbeitung auf mehreren Blade-Servern abgedeckt werden.



(a) Top-20.000 (SynthC)



(b) Top-100.000 (SynthC)

Abbildung 5.8.: Laufzeitverteilung für verschiedene Partitionierungen

Die dazu erforderliche parallele Verarbeitung der EHK wurde in Form des PARTITION-Algorithmus implementiert. Zur Berechnung der nötigen EHK auf den einzelnen Partitionen sind der Algorithmus BUC für Attributwertkorrelationen und ECLAT für Warenkorbanalysen verfügbar. Beide sind in der Lage, die im BWA vorhandenen Daten direkt zu verarbeiten. Dieses Kapitel bot Details über die verwendeten Datenstrukturen und die benötigten Einzelschritte einer solchen EHK.

Im Abschnitt 5.6 wurde die Implementierung von Verfahren zur optischen Aufbereitung von Assoziationsregeln untersucht und ein Ansatz zur Selektion interessanter Regeln vorgestellt. Hierbei wurde in mehreren gleichzeitigen Sichten eine große Regelmenge effizient gefiltert, was die interaktive Beschreibung subjektiver Suchziele ermöglicht. Insgesamt wurde gezeigt, dass eine Assoziationsregelsuche nicht zwingend technisches Fachwissen des Anwenders erfordern muss. Automatische Steuerungen der Algorithmen erlauben deren robuste Durchführung bei geringem Mehraufwand. Das folgende Kapitel fasst nun die Erkenntnisse dieser Arbeit zusammen und bietet einen Ausblick auf mögliche Erweiterungen.

6. Zusammenfassung und Ausblick

Diese Arbeit befasst sich mit der Alltagstauglichkeit moderner Massendatenverarbeitung. Durch neue Möglichkeiten der Datenspeicherung wachsen die vorhandenen Datenmengen sehr stark an. Diese müssen verarbeitet werden, um Informationen zu gewinnen und Wissen zu bilden.

Diese Arbeit zeigt einen Ansatz, das Verfahren der Assoziationsregelsuchen, das Finden von Korrelationen in einer Datenmenge, auf einen breiten Benutzerkreis zu erweitern und hierfür neue Anwender zu gewinnen. Besonderes Augenmerk liegt auf der einfachen und robusten Anwendung der Verfahren, wodurch komplizierte Parameter und speziell nötiges Fachwissen vermieden werden sollen. Die Assoziationsregelsuche kann damit von der Spezialanwendung zum Alltagswerkzeug gewandelt werden.

Die Assoziationsregelanalyse durchläuft drei wesentliche Schritte: das Extrahieren häufiger Kombinationen (EHK), die Bildung und Bewertung von Assoziationsregeln und deren manuelle Auswertung durch den Nutzer. Ohne tiefes Verständnis der Verfahren kann dem Nutzer nur die Regelauswertung zugemutet werden und die ersten beiden Schritte sollten automatisiert erfolgen. Sie stellen lediglich nötige Vorverarbeitungen dar, welche der Anwender nicht ausdrücklich anfordert.

Hierfür werden im Kapitel 3 Möglichkeiten gezeigt, wie sich eine EHK mit einfacher Parametrisierung mit den Besonderheiten moderner Rechnerarchitekturen und deren Daten vereinigen lassen. Die EHK wird in Form des Algorithmus ECLAT um die Fähigkeit zur Suche von Top-N-Kombinationen erweitert. Dies ist leicht verständlich und datenunabhängig. Der Algorithmus passt sich an die Daten an und kann damit auch von unerfahrenen Anwendern verwendet werden. Dies unterscheidet sich von dem bisher üblichen Wählen einer Mindesthäufigkeit, deren optimaler Wert signifikant von den Eigenheiten der untersuchten Daten abhängt.

Moderne Systemarchitekturen verarbeiten Daten zudem oft parallel auf mehreren Einzelrechnern. Hierfür wird eine Möglichkeit zur verteilten Top-N-EHK vorgestellt. Unter ungünstigen Datenverteilungen kann die Berechnung des Gesamtergebnisses aus den Einzelpartitionen schwierig sein, wodurch die verteilte Top-N-EHK stark verlangsamt wird. Dafür werden durch die Algorithmen STH und MAST laufzeitrobuste, verteilte Top-N-EHKs vorgestellt, welche sich auf ein approximiertes Ergebnis beschränken. Für dieses wird eine Grenze bestimmt, welche garantierte und nicht garantierte Ergebnisse trennt.

Dabei sind die Häufigkeiten aller Kombinationen stets korrekt, wodurch auch die daraus erzeugten Assoziationsregeln korrekt sind. Allerdings werden einige Regeln möglicherweise nicht ermittelt. Dem Anwender können aber einfach sinnvolle Kombinationen zur Verfügung gestellt werden, um Assoziationsregeln zu bilden.

Im Kapitel 4 wird deren automatische Aufbereitung und Bewertung untersucht. Auf Interaktion mit dem Anwender soll auch hier möglichst verzichtet werden und die Auswertung soll stattdessen automatisch erfolgen. Assoziationsregeln können mit Interessantheitsmaßen, wie Konfidenz oder Regelhäufigkeit, versehen werden, welche deren Wert repräsentieren. In der Literatur wurden viele dieser Maße untersucht, die je nach Anforderung des Anwenders dessen subjektiver Regelbewertung entsprechen können. Der automatischen Auswertung ist diese Anforderung nicht bekannt und sie muss aus der Vielzahl möglicher Interessantheitsmaße eine repräsentative Ordnung alle gefundenen Regeln und deren effiziente Auswertung ermöglichen.

In dieser Arbeit werden dafür zwei Ansätze betrachtet. Zuerst werden mehrere Interessantheitsmaße konsolidiert, normalisiert und zu einem globalen Maß vereinigt. Nach diesem kann eine Ordnung der Regeln abgeleitet werden, welche mehrere subjektive Sichtweisen in sich vereinigt. Dem Nutzer wird damit eine Reihenfolge zur sinnvollen Auswertung der Regeln zur Verfügung gestellt, welche Regeln bevorzugt, die aus mehreren Sichtweisen wichtig erscheinen.

Der zweite Ansatz zeigt Möglichkeiten zur Gruppierung großer Assoziationsregelmengen. Werden dem Anwender Regelgruppen statt Einzelregeln präsentiert, kann dieser effizient in der verbleibenden Auswahl navigieren und filtern. Dafür wird vorgeschlagen, Regeln nach den Häufigkeiten der Regelelemente zu vereinigen. Das unterscheidet den Ansatz von anderen Verfahren, welche Regeln nach deren Syntax gruppieren. Regeln gehören dabei zu einer Gruppe, wenn alle äquivalenten relevanten Häufigkeiten zweier Regeln um weniger als einen gewählten Prozentwert voneinander abweichen. Unter der Annahme, dass ähnliche Häufigkeiten vergleichbare Interessantheitsmaße bilden, werden alle Regeln der Gruppe unabhängig von einem konkreten Maß ähnlich bewertet. Auf diese Weise können Gruppen komplett akzeptiert oder verworfen werden, was den Aufwand für deren Auswertung deutlich reduziert.

Abschließend wird in Kapitel 5 die Implementierung der Verfahren dieser Arbeit in einem Prototyp erläutert. Die Grundlage hierfür bildet der SAP BW Accelerator der SAP AG. Unter Nutzung von dessen Ressourcen und Funktionalitäten werden Details und Datenstrukturen der praktischen Umsetzung der Algorithmen vorgestellt.

Insgesamt wird gezeigt, dass die einfache Verwendung von Assoziationsregelsuchen auch bei großen und verteilten Datenmengen ermöglicht werden kann. Dies löst die Beschränkung des Verfahrens, stets Fachleute zur Durchführung von Assoziationsregelanalysen zu benötigen und erlaubt damit die Erschließung von neuen Nutzergruppen, Anwendungen und Wissen.

Ausblick

Im Hinblick auf die einfach Verarbeitung großer Datenbestände ist die Kombination der in dieser Arbeit vorgestellten Verfahren und die Verwendung von Stichproben sinnvoll. Erfolgt deren Anwendung vor dem Anwender verdeckt, robust und zuverlässig, sind Effizienzsteigerungen und damit Kostenersparnisse möglich. Dabei entstehende Konfidenzintervalle können mit den Grenzwerten der garantierten Kombinationen bei STH kombiniert werden. Alternativ kann die Verwendung von Stichproben zur Abschätzung des Laufzeitverhaltens eines EHK-Laufes genutzt werden, um die Güte von deren Parametern zu prüfen. Dies kann die robuste, automatisierte Durchführung der EHK weiter erleichtern.

Nachdem diese Arbeit bereits Möglichkeiten zeigt, den Arbeitsschritt der Extraktion häufiger Kombinationen einfach durchzuführen, werden zur automatischen Bewertung der Regeln immer noch eine Reihe von Parametern benötigt. Dies umfasst sowohl Schwellwerte einzelner Interessantheitsmaße als auch die Kombination der Bedeutungen einzelner Maße zu einer Gesamtbewertung. Hierfür kann die Verwendung verschiedener Vereinigungsstrategien geprüft werden. Außerdem können Gewichtungen für die einzelnen Maße automatisiert bestimmt werden, indem die Maße nach ihrer Aussagekraft im Vergleich zu den anderen Maßen betrachtet werden. Zudem sind Ansätze denkbar, die Auswahl der relevanten Maßen derartig zu optimieren, dass alle Regeln insgesamt bestmöglich platziert werden. Diese verfeinerten Strategien können die Gesamtrangfolge der Regeln potenziell besser repräsentieren als die in dieser Arbeit verwendete Aufsmmierung der Einzelränge. Diese Erweiterung kann die Assoziationsregelbewertung vereinfachen.

Alle Interessantheitsmaße bieten dem Anwender lediglich einen Bewertungsvorschlag. Das eigentliche Ziel ist aber, die subjektiven Wünsche des Anwenders korrekt zu erkennen. Die dazu nötige Anforderungsbeschreibung durch den Nutzer steht dabei im Kontrast zu einer einfachen Verwendung der Verfahren. Hierfür könnte ein Nutzer beispielsweise Regeln als repräsentatives Beispiel wählen, welche durch Lernverfahren helfen, zukünftige Anfragen des Anwenders korrekt zu deuten. Ebenso nützlich kann eine optische Darbietung der Regeln sein, um die Wahl wichtiger Regeln zu erleichtern. Kann diese Lernphase derartig vereinfacht werden, können Assoziationsregeln für viele Anwendern einen deutlichen und nützlichen Mehrwert darstellen.

A. Datenbestände

Name	Transaktionen	Elemente	Zeilen	Partitionen
SynthA	1.000	225	10.000	1
SynthB	800.000	770	16.000.000	1,2,4,6,8,10,40
SynthC	980.000	24.000	10.000.000	1
Connect-4	67.557	130	3.000.000	1
Gazelle	60.000	500	150.000	1
Retail	85.146	16.398	820.414	1, 2
KundeA	134.167	72.252	396.324	1,2,4,10,22
KundeA2	1.341.670	72.252	3.963.240	1
KundeB	33.542.000	72.025	110.485.750	40

Tabelle A.1.: Übersicht genutzter Datenbestände zur Warenkorbanalyse

Name	Zeilen	Attribute	Elemente	Partitionen
Mushroom	8.126	10	2 . . . 13	1
Mushroom2	812.600	10	2 . . . 13	1
Mushroom3	8.126.000	10	2 . . . 13	1,2,4,6,8,10
KundeC	183.492.739	15	13x $\leq$ 1.500	1,2,4,6,8,10
			2x $>$ 20.000	

Tabelle A.2.: Übersicht genutzter Datenbestände zur Attributkorrelationsanalyse

Die für diese Auswertung verwendeten Datenbestände bestehen zum Teil aus transaktionalen Daten und aus relationalen Daten. Typische transaktionale Daten sind z.B. Belegdaten aus dem Einzelhandel, wobei jede Transaktion einen Warenkorb darstellt. Relationale Daten liegen als Attributwertpaare vor und können z.B. beliebige Geschäfts- und Verkaufsdaten beinhalten.

In dieser Arbeit wird versucht, eine möglichst gute Abdeckung denkbarer Szenarien zu erreichen. Dafür werden sowohl synthetische Daten als auch mehrere reale Daten untersucht. Des Weiteren werden in verwandten Publikationen oft verwendete Referenzdatenbestände untersucht. Diese Datenbestände sind öffentlich verfügbar [FIM08].

Zur Verifizierung der implementierten Algorithmen für SAP-Daten werden einige realistischen Kundenszenarien untersucht, wobei diese Datenbestände nicht öffentlich verfügbar

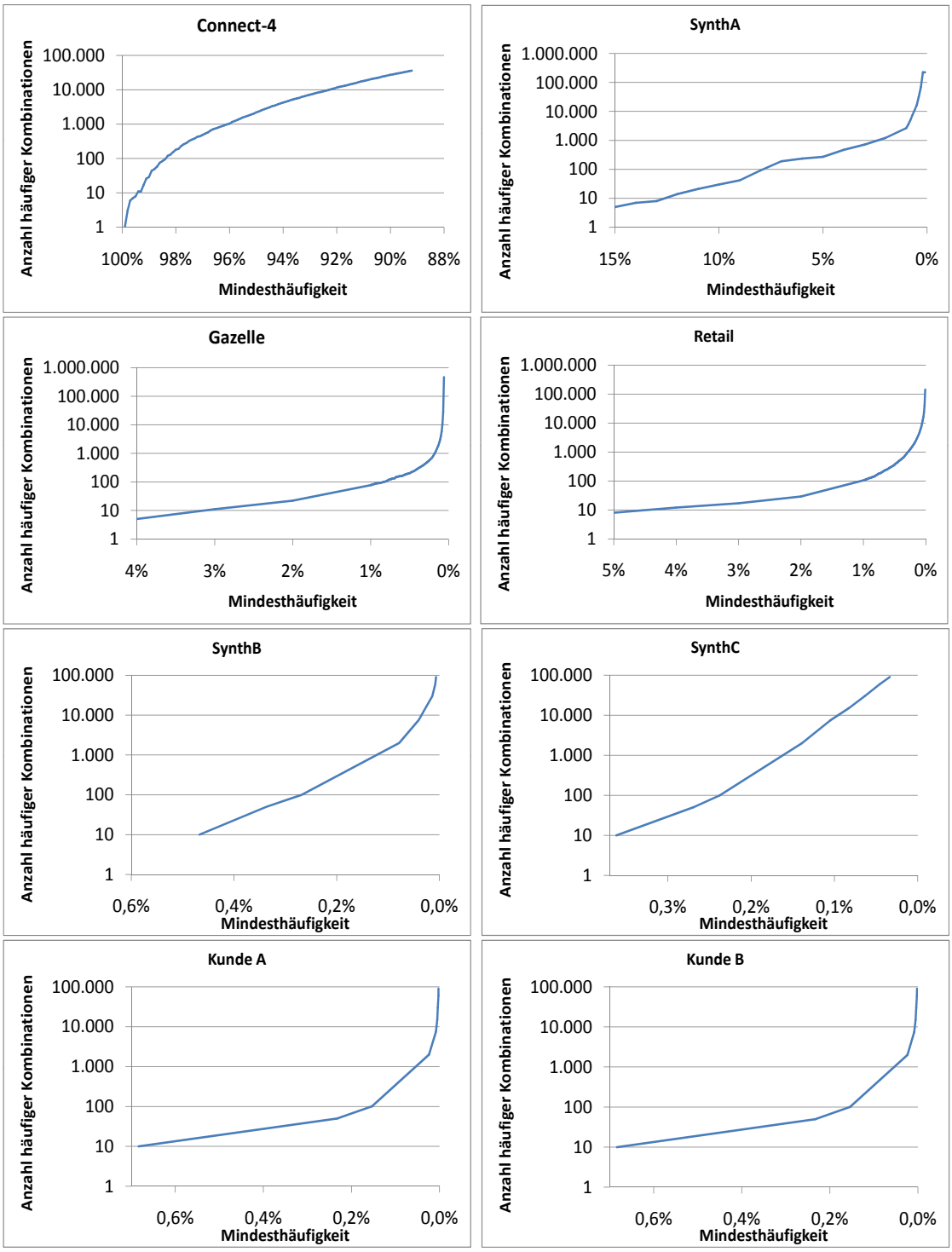


Abbildung A.1.: Kombinationen pro Mindesthäufigkeit (Warenkorbanalyse)

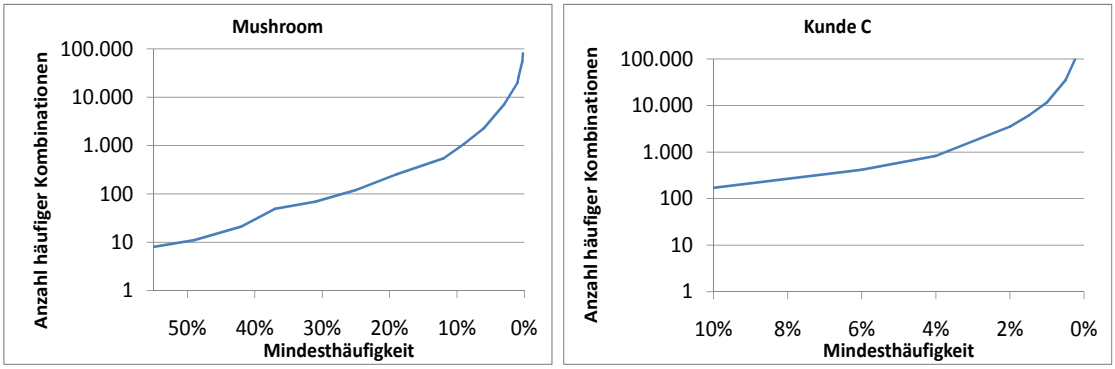


Abbildung A.2.: Kombinationen pro Mindesthäufigkeit (Attributwertkorrelation)

sind. Deren Struktur kann jedoch mit Hilfe der aus Tabelle A erkennbaren Parameter näherungsweise reproduziert werden.

SynthA, SynthB, SynthC

Diese Datenbestände sind mit Hilfe des im IBM Almaden Research Center entwickeltem *Synthetic Data Generation Code for Associations and Sequential Patterns* [Sri] erzeugt. Dieser Generator erstellt Verkaufsdaten, welche realistischen Daten stark ähneln. Die Datenbestände SynthA und SynthB wurden durch den Illimine Data Generator des Data-Mining Projektes der Data-Mining Forschungsgruppe an der Universität Illinois [Ill] erzeugt. Dieser basiert auf dem bereits vorgestellten Generator von IBM Research. Die benutzten Parameter sind aus der Tabelle A zu entnehmen.

Connect-4

Dieser Datenbestand wird bereitgestellt vom UCI Machine Learning Repository [AN07]. Er stellt einen sehr dichten Datenbestand dar, welcher die unterschiedlichen Spielstände sowie den Ausgang eines „Vier-gewinnt“- Spiels beinhaltet. Die Anzahl der Elemente pro Transaktion ist mit ca. 45 Elementen für Assoziationsregeln sehr groß. Dieser Datenbestand wird oft benutzt, um die Effekte von Optimierungen zu verdeutlichen.

Retail

Der Retail-Datenbestand ist ein von Tom Brijs bereit gestellter Datenbestand mit anonymisierten Warenkorbdaten einer anonymen belgischen Handelskette [BSVW99]. Mit ca. 85.000 Transaktionen ist dieser Datenbestand verhältnismäßig klein, ist jedoch mit seiner Datencharakteristik mit von SAP Kunden bekannten Verkaufsdaten vergleichbar. Die zweigeteilte Version dieses Datenbestandes enthält in einer Partition lediglich kurze,

in der zweiten nur lange Transaktionen. Dies simuliert Verteilungen mit unterschiedlichen Datencharakteristiken pro Partition.

Gazelle

Dieser realistische Datenbestand enthält Mausklickabfolgen zu den Bestelldaten eines bereits insolventen Webshops. *Gazelle* ist ein bekannter Referenzdatenbestand des KDD Cups 2000 [ZKM01].

Kunde A, KundeA2, Kunde B

Diese Datenbestände dienen zur Evaluation der Algorithmen auf SAP-Szenarien. Sie bestehen aus realistischen Datenbeständen von SAP-Kunden und müssen entsprechend vertraulich behandelt werden. Sie verhalten sich jedoch ähnlich zu den Datenbeständen *Retail* und *Gazelle*, welche öffentlich verfügbar sind. KundeA2 ist aus den Daten von KundeA erzeugt, indem alle Transaktionen zehnfach unter zehn verschiedenen Bezeichnern eingefügt wurden. Die Datencharakteristiken sind gleich zu KundeA, nur die Datenmenge wurde verzehnfacht.

Kunde C

Dieser Datenbestand besteht aus einem SAP InfoCube, welcher SAP-Kundendaten enthält. Dem entsprechend ist dieser Datenbestand vertraulich zu behandeln. Er dient im Rahmen dieser Arbeit zur Evaluation der Algorithmen zur Attributwertkorrelationsbestimmung.

Mushroom

Der Mushroom-Datenbestand enthält ein Verzeichnis von Pilzen. Dafür ist jeder Pilz mit mehreren Eigenschaften versehen welche jeweils als eigenes Attribut abgelegt sind. Dieser Datenbestand ist ebenfalls Teil des UCI Machine Learning Repository [AN07] und dient im Rahmen dieser Arbeit als Beispieldatenbestand für Assoziationsregelsuchen zwischen Attributausprägungen. Er enthält in der Ausgangsversion 8.125 Einträge mit jeweils 24 Attributen. Diese enthalten jeweils zwischen zwei und dreizehn unterschiedliche Attributausprägungen.

Im Rahmen der Arbeit wurde dieser Datenbestand auch in zwei vergrößerten Formen genutzt (Mushroom2=Mushroom·100, Mushroom3=Mushroom·1000), indem er mehrfach in einen neuen Datenbestand repliziert wurde. Der entstehende Datenbestand weist die gleichen Datencharakteristiken und Abhängigkeiten wie der Ausgangsdatenbestand auf, besitzt jedoch ein Vielfaches an Einträgen.

Top-N-Eclat-Beispiel für approximiertes Ergebnis

Gemäß dem Beispiel aus Abschnitt 3.1 wird im Folgenden das Verhalten von TOP-N-ECLAT erläutert, wenn Kombinationen mit gleicher Häufigkeit wie der seltenste gültige Ergebniskandidat nicht relevant sind. Die Mindesthäufigkeit entspricht dabei im Falle von N verfügbaren Ergebniskandidaten der Häufigkeit der N-ten Kombination plus eins.

Nr.	**Top-Liste**	**sup**
1	[**D:8**]	1
2	[**D:8, C:7**]	1
3	[**D:8, C:7, CD:3**]	1
4	[**D:8, C:7, B:5, CD:3**]	1
5	[**D:8, C:7, B:5, BC:5, CD:3**]	4
6	[**D:8, C:7, B:5, BC:5, A:4**, CD:3]	5
7	[**D:8, C:7, B:5, BC:5, A:4**, CD:3]	5

Tabelle A.3.: Beispiel Ablauf TOP-N-ECLAT mit erhöhter Mindesthäufigkeit

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Symbol	Beschreibung	Erstmals in Abschnitt
ACHARM	Approximiert arbeitendes CHARM	3.4.1
APRIORI	Grundlegendes Verfahren zur EHK	2.2.1
BUC	Bottom Up Computation, EHK (Attributkorr.)	2.2.4
CFP	Condensed FP-Base, Verfahren für unscharf-geschlossene Kombinationen	3.4.2
CHARM	ECLAT für geschlossene Kombinationen	2.2.7
CLOSET	FP-GROWTH für geschlossener Kombinationen	2.2.7
DFCIM	Distributed Frequent Closed Itemset Mining, Verfahren zur Bestimmung verteilter geschlossener häufiger Kombinationen	3.4.2
dECLAT	Diffset ECLAT, Optimierung von ECLAT	2.2.3
EASE	EHK mit adaptiven Stichproben	2.2.9
ECLAT	Equivalence CLAss Transformation, EHK-Verfahren	2.2.3
EHKSIM	Simulierte EHK mittels Elementhäufigkeiten	3.4.3
EXPSUP	Verfahren zur Schätzung einer Mindesthäufigkeit	3.4.3
FAST	EHK mit adaptiven Stichproben	2.2.9
FASTINC	Optimierung von TOP-N-ECLAT	3.1.1
FCFIM	Fuzzy-Closed Frequent Itemset Mining, Verfahren für unscharf-geschlossene häufige Kombinationen	3.4.2
FP-GROWTH	Kombinationsbaumbasierte EHK	2.2.2
MT-CLOSED	Verfahren zur Bestimmung verteilter geschlossener häufiger Kombinationen	3.4.2
STH	Save-ThresHold-Algorithmus, Erweiterung von TOP-N-ECLAT für verteilte Datenbestände	3.2.1
PARTITION	Verfahren zur EHK auf verteilten Datenbeständen	2.2.5
TCFI	Tolerant Closed Frequent Itemsets, Verfahren für unscharf-geschlossene häufige Kombinationen	3.4.2
TFP	Top-K Frequent Patterns, Verfahren zur Top-N-EHK	2.2.6
TOP-N-ECLAT	Top-N-Implementierung von ECLAT	3.1
TPARTITION	Vereinigung von STH und PARTITION, Top-N-EHK auf verteilten Datenbeständen	3.2.4

Abkürzungsverzeichnis

Abkürzung	Bedeutung
BI	Business Intelligence
BWA	Business Warehouse Accelerator
FCI	geschlossene Kombinationen (engl. Frequent Closed Itemsets)
EHK	Extraktion Häufiger Kombinationen
engl.	englisch
GB	Gigabyte
GHz	Gigahertz
ID	Identifier - eindeutige Kennzeichnung
Mio.	Millionen
Mrd.	Milliarden
OLAP	OnLine-Analytical-Processing, ein Verfahren der Datenanalyse
P2P	Peer-to-Peer, Rechnerverbindungskonzept
SAP	Firmenname, ehemals Akronym für *Systeme, Anwendungen und Produkte in der Datenverarbeitung*
SIMD	Single Instruction-Multiple Data, Datenverarbeitungskonzept
SQL	Structured Query Language, Datenanfragesprache
TA	Threshold Algorithms (deutsch: Schwellwertalgorithmen)
TREX	Text Retrival and EXtraction, Textsuchmaschine
vgl.	Vergleiche

Variablenverzeichnis

Variable	Bedeutung
$\triangle$	symmetrische Differenz
A	Antecedent, Bedingung einer Assoziationregel; vgl. Conclusion C
a	Knotenbezeichnung im Kombinationsbaum
α	Erweiterungsfaktor für N
b	Größtmögliche relative Mindesthäufigkeit kleiner als $support(Q)$
β	Unschärfefaktor bei FCFIM
C	Conclusion, Schlussfolgerung einer Assoziationsregel; vgl. Antecedent A
c	Korrelationskoeffizient
δ	Mindestanteil im Verhältnis zum nächsten Einzelelement
DB	Datenbestand, in dieser Arbeit eine Menge von Transaktionen mit Elementen aus E
DDB	Verteilter Datenbestand, vgl. DB
E	Menge der Einzelelemente in DB
E'	Menge der auftretenden Einzelelemente in J
e	Einzelelement, $e \in E$
ϵ	Maximaler absoluter Fehler
expSup	Geschätzte sinnvolle Mindesthäufigkeit
f	Anzahl der betrachteten häufigsten Elemente e
γ	Qualifizierender Anteil der möglichen Platzierungen
G	Menge von Ergebniskandidaten für häufige Kombinationen
gs	Garantierter relativer Grenzwert, vgl. s
H	Menge der geschlossenen häufigen Kombinationen, $H \subset I$
I	Menge der häufigen Kombinationen
J	Menge ermittelten Assoziationsregeln
K	Kardinalität von DB, $K = \|DB\|$
k	Nummer eines Durchlaufes
L	Zeile bzw. Transaktion in DB
lb	Untere Grenze der Häufigkeit des N-ten Ergebnisses einer Top-N-Suche bei FASTINC
m	$m = \|J\|$
mams	Minimale Mindesthäufigkeit aller Partitionen (minimum absolute minsupport)

mpl	Maximal erlaubte Anzahl an Elementen in X
μ	Maximale Abweichung einer durch Stichproben geschätzten Häufigkeit
N	Anzahl gesuchter Ergebnisse einer Top-N-Suche
n	$n = \|I\|$
$\mathcal{O}$	Obere Komplexitätsschranke
o	$o = \|V\|$ eines Vektors V, Anzahl selektierter Einträge
$\mathcal{P}$	Potenzmenge
ϕ	Nachbarschaftsradius eines Regelclusters
P	Partition, $\bigcup_{i=1}^{p} P_i = DDB$, bei p Partitionen
p	Anzahl Partitionen in DDB
Q	Seltensten ermittelten Kombination einer Ergebnismenge
ρ	Gewichtungsparametern für Regelteile bei einer Gruppierung
R	Anzahl gesuchter Assoziationsregeln
r	Nachbarschaftsradius
S	Stichprobenumfang vgl. K
s	Save-Threshold Häufigkeit
SDB	Auf häufige Elemente reduziertes DB
si	Save-Threshold Häufigkeit in Kombination mit $mams > 1$
γ	Wahrscheinlichkeit, dass eine auf Stichproben bestimmte Häufigkeit maximal um μ abweicht
t	Anzahl lokal betrachteter Transaktionen
u	Untere Grenze der Häufigkeit des N-ten Ergebnisses einer Top-N-Suche bei TPARTITION
v	Platzierungsvektor
W	Häufigste Kombination, welche nicht ermittelt wurde
w	Platzierungsvektor
X	Kombination, bestehend aus Elementen e
X'	$X' \subset X$
Y	Assoziationsregel, $Y \in J$
Z	Repräsentant eines Clusters

Literaturverzeichnis

[AAB+08] Agrawal, Rakesh ; Ailamaki, Anastasia ; Bernstein, Philip ; Brewer, Eric ; Carey, Michael ; Chaudhuri, Surajit ; Doan, AnHai ; Florescu, Daniela ; Franklin, Michael ; Garcia-Molina, Hector ; Gehrke, Johannes ; Gruenwald, Le ; Haas, Laura ; Halevy, Alon ; Hellerstein, Joseph ; Ioannidis, Yannis ; Korth, Hank ; Kossmann, Donald ; Madden, Samuel ; Magoulas, Roger ; Ooi, Beng C. ; O'Reilly, Tim ; Ramakrishnan, Raghu ; Sarawagi, Sunita ; Stonebraker, Michael ; Szalay, Alexander ; Weikum, Gerhard: The Claremont Report on Database Research. In: *SIGMOD Record* 37 (2008), Nr. 3, S. 9–19

[AIS93] Agrawal, Rakesh ; Imieliński, Tomasz ; Swami, Arun: Mining association rules between sets of items in large databases. In: *Proc. of the 1993 ACM SIGMOD Intl. Conf. on Management of Data*, 1993, S. 207–216

[AL99] Aumann, Yonatan ; Lindell, Yehuda: A statistical theory for quantitative association rules. In: *Proc. of the 1999 Intl. Conf. on Knowledge Discovery and Data Mining*, 1999, S. 261–270

[AN07] Asuncion, A. ; Newman, D.J.: *UCI Machine Learning Repository*. `http://www.ics.uci.edu/mlearn/MLRepository.html`. Version: 2007

[AS94] Agrawal, Rakesh ; Srikant, Ramakrishnan: Fast Algorithms for Mining Association Rules. In: *Proc. of the 1994 Intl. Conf. on Very Large Data Bases*, 1994, S. 487–499

[AZJ01] Ammoura, Ayman ; Zaïane, Osmar ; Ji, Yuan: Immersed Visual Data Mining: Walking the Walk. In: *Proc. of the 2001 British National Conf. on Databases*, 2001, S. 202–218

[BAG99] Bayardo, Roberto ; Agrawal, Rakesh ; Gunopulos, Dimitrios: Constraint-based rule mining in large, dense databases. In: *Proc. of the 1999 Intl. Conf. on Data Engineering*, 1999, S. 188–197

[Bay98] Bayardo, Roberto: Efficiently mining long patterns from databases. In: *Proc. of the 1998 ACM SIGMOD Intl. Conf. on Management of Data*, 1998, S. 85–93

[BCD+03] Brönnimann, Hervé ; Chen, Bin ; Dash, Manoranjan ; Haas, Peter ; Scheuermann, Peter: Efficient data reduction with EASE. In: *Proc. of the 2003 Intl. Conf. on Knowledge Discovery and Data Mining*, ACM, 2003, S. 59–68

[BCD+04] Brönnimann, Hervé ; Chen, Bin ; Dash, Manoranjan ; Qiao, Yi ; Haas, Peter ; Scheuermann, Peter: Efficient Data-Reduction Methods for On-Line Association Rule Discovery. In: *Data Mining: Next Generation Challenges and Future Directions*, AAAI Press, 2004, S. 190–208

[BD06] Burns, Rick ; Dorin, Robert: *The SAP NetWeaver BI Accelerator: Transforming Business Intelligence*. 2006. – White Paper, Winter Corporation

[BKK97] Brunk, Cliff ; Kelly, James ; Kohavi, Ron: MineSet: An Integrated System for Data Mining. In: *Proc. of the 1997 Intl. Conf. on Knowledge Discovery and Data Mining*, 1997, S. 135–138

[BKS01] Börzsönyi, Stephan ; Kossmann, Donald ; Stocker, Konrad: The Skyline Operator. In: *Proc. of the 2001 Intl. Conf. on Data Engineering*, 2001, S. 421–430

[Blo70] Bloom, Burton H.: Space/time trade-offs in hash coding with allowable errors. In: *Communications of the ACM* 13 (1970), Nr. 7, S. 422–426

[BM00] Bernardino, Jorge ; Madeira, Henrique: A new technique to speedup queries in data warehousing. In: *Symposium on Advances in Databases and Information Systems - 4th East-European Conf. on Advances in Databases and Information Systems*, 2000, S. 21–32

[BMS97] Brin, Sergey ; Motwani, Rajeev ; Silverstein, Craig: Beyond Market Baskets: Generalizing Association Rules to Correlations. In: *Proc. of the 1997 ACM SIGMOD Intl. Conf. on Management of Data*, 1997, S. 265–276

[BMUT97] Brin, Sergey ; Motwani, Rajeev ; Ullman, Jeffrey ; Tsur, Shalom: Dynamic itemset counting and implication rules for market basket data. In: *Proc. of the 1997 ACM SIGMOD Intl. Conf. on Management of Data*, 1997, S. 255–264

[Bon05] Boncz, Peter: *MonetDB: A high performance database kernel for query-intensive applications*. 2005. – `http://monetdb.cwi.nl/projects/monetdb/Development/Assets`

[Bor03] Borgelt, Christian: *Efficient Implementations of Apriori and Eclat*. 2003

[BP99] Bay, Stephen D. ; Pazzani, Michael J.: Detecting change in categorical data: mining contrast sets. In: *Proc. of the 1999 Intl. Conf. on Knowledge Discovery and Data Mining*, 1999, S. 302–306

[BPT97] Baralis, Elena ; Paraboschi, Stefano ; Teniente, Ernest: Materialized Views Selection in a Multidimensional Database. In: *Proc. of the 1997 Intl. Conf. on Very Large Data Bases*, 1997, S. 156–165

[BR99] Beyer, Kevin ; Ramakrishnan, Raghu: Bottom-up computation of sparse and Iceberg CUBE. In: *Proc. of the 1999 ACM SIGMOD Intl. Conf. on Management of Data*, 1999, S. 359–370

[Bro07] Brodie, Michael: Computer Science 2.0: a new world of data management. In: *Proc. of the 2007 Intl. Conf. on Very Large Data Bases*, 2007, S. 1161

[BSVW99] Brijs, Tom ; Swinnen, Gilbert ; Vanhoof, Koen ; Wets, Geert: Using Association Rules for Product Assortment Decisions: A Case Study. In: *Knowledge Discovery and Data Mining*, 1999, S. 254–260

[CCR78] Charnes, Abraham ; Cooper, William ; Rhodes, Edwardo: Measuring the efficiency of decision making units. In: *European Journal of Operational Research* 2 (1978), November, Nr. 6, S. 429–444

[CCS] Codd, Edgar ; Codd, Sharon ; Salley, Clynch: *Providing OLAP (On-line Analytical Processing) to User Analysts: An IT Mandate, White Paper, Arbor Software Corporation, 1993.* `http://www.fpm.com/refer/codd.html`

[CH97] Cabena, Peter ; Hadjinian, Pablo: *Discovering Data Mining: From Concept to Implementation.* London : Prentice Hall PTR, 1997. – S. 12

[Che04] Cheung, Yin-Ling: Mining Frequent Itemsets without Support Threshold: With and without Item Constraints. In: *IEEE Trans. on Knowl. and Data Eng.* 16 (2004), Nr. 9, S. 1052–1069

[CHS02] Chen, Bin ; Haas, Peter ; Scheuermann, Peter: A new two-phase sampling based algorithm for discovering association rules. In: *Proc. of the 2002 Intl. Conf. on Knowledge Discovery and Data Mining*, 2002, S. 462–468

[CKN06] Cheng, James ; Ke, Yiping ; Ng, Wilfred: δ-Tolerance Closed Frequent Itemsets. In: *Proc. of the 2006 IEEE Intl. Conf. on Data Mining*, 2006, S. 139–148

[CMS99] Card, Stuart ; Mackinlay, Jock ; Shneiderman, Ben: *Readings in information visualization: using vision to think.* Erste Edition. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1999. – S. 93-142

[Con01] Cong, Shi: *Mining The Top-K Frequent Itemset With Minimum Length M.* 2001 School of Computing Science, Simon Fraser University, Master

[CTX+04] Cong, Gao ; Tung, Anthony K. H. ; Xu, Xin ; Pan, Feng ; Yang, Jiong: FARMER: finding interesting rule groups in microarray datasets. In: *Proc. of the 2004 ACM SIGMOD Intl. Conf. on Management of Data*, 2004, S. 143–154

[CW04] Cao, Pei ; Wang, Zhe: Efficient top-k query calculation in distributed networks. In: *Proc. of the Intl. Symposium on Principles Of Distributed Computing*, 2004, S. 206–215

[CX98] Cheung, David Wai-Lok ; Xiao, Yongqiao: Effect of Data Skewness in Parallel Mining of Association Rules. In: *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 1998, S. 48–60

[CYH06] Cheng, Hong ; Yu, Philip ; Han, Jiawei: AC-Close: Efficiently Mining Approximate Closed Itemsets by Core Pattern Recovery. In: *Proc. of the 2006 Intl. Conf. on Data Mining*, 2006, S. 839–844

[DL98] Dong, Guozhu ; Li, Jinyan: Interestingness of Discovered Association Rules in terms of Neighborhood-Based Unexpectedness. In: *Research and Development in Knowledge Discovery and Data Mining, Proc. of the 2nd Pacific-Asia Conf. Knowledge Discovery and Data Mining, PAKDD*, 1998, S. 72–86

[FIM08] *Frequent Itemset Mining Dataset Repository.* 2008. – `http://fimi.cs.helsinki.fi/data`

[FKT00] Fu, Ada Wai-Chee ; Kwong, Renfrew ; Tang, Jian: Mining N-most Interesting Itemsets. In: *Proc. of the 2000 Intl. Symposium on Foundations of Intelligent Systems*, 2000, S. 59–67

[FLN01] Fagin, Ronald ; Lotem, Amnon ; Naor, Moni: Optimal aggregation algorithms for middleware. In: *Proc. of the 2001 ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems*, 2001, S. 102–113

[Fri92] Friendly, Michael: Mosaic displays for loglinear models. In: *ASA, Proc. of the Statistical Graphics Section.*, 1992, S. 61–68

[GBK00] Güntzer, Ulrich ; Balke, Wolf-Tilo ; Kiessling, Werner: Optimizing Multi-Feature Queries for Image Databases. In: *Proc. of the 2000 Intl. Conf. on Very Large Data Bases*, 2000, S. 419–428

[GBY07] Gedik, Buğra ; Bordawekar, Rajesh ; Yu, Philip: CellSort: high performance sorting on the cell processor. In: *Proc. of the 2007 Intl. Conf. on Very Large Data Bases*, 2007, S. 1286–1297

[GCB+97] Gray, Jim ; Chaudhuri, Surajit ; Bosworth, Adam ; Layman, Andrew ; Reichart, Don ; Venkatrao, Murali ; Pellow, Frank ; Pirahesh, Hamid: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. In: *Journal of Data Mining and Knowledge Discovery* 1 (1997), Nr. 1, S. 29–53

[GGKM06] Govindaraju, Naga ; Gray, Jim ; Kumar, Ritesh ; Manocha, Dinesh: GPUTeraSort: high performance graphics co-processor sorting for large database management. In: *Proc. of the 2006 ACM SIGMOD Intl. Conf. on Management of Data*, 2006, S. 325–336

[GGP98] Guillaume, Sylvie ; Guillet, Fabrice ; Philippe, Jacques: Improving the Discovery of Association Rules with Intensity of Implication. In: *Proc. of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, 1998, S. 318–327

[GH06] Geng, Liqiang ; Hamilton, Howard: Interestingness measures for data mining: A survey. In: *ACM Computational Survey* 38 (2006), Nr. 3, S. 9

[GHH+08] Guo, Hongyu ; He, Bingsheng ; He, Yifan ; Luo, Qiong ; Peng, Bo ; Xiao, Xiangye: *Frequent Pattern Mining on Graphics Processors*. 2008

[GHQ95] Gupta, Ashish ; Harinarayan, Venky ; Quass, Dallan: Aggregate-query processing in data warehousing environments. In: *Proc. of the 1995 Intl. Conf. on Very Large Data Bases*, 1995, S. 358–369

[GRM05] Govindaraju, Naga ; Raghuvanshi, Nikunj ; Manocha, Dinesh: Fast and approximate stream mining of quantiles and frequencies using graphics processors. In: *Proc. of the 2005 ACM SIGMOD Intl. Conf. on Management of Data*, 2005, S. 611–622

[GSG99] Gupta, Gunjan K. ; Strehl, Alexander ; Ghosh, Joydeep: Distance based clustering of association rules. In: *Intelligent Engineering Systems Through Artificial Neural Networks*, ASME Press, 1999, S. 759–764

[GZ03a] Goethals, Bart ; Zaki, Mohammed: Advances in Frequent Itemset Mining Implementations: Report on FIMI'03. In: Goethals, Bart (Hrsg.) ; Zaki, Mohammed (Hrsg.): *Proc. of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations* Bd. 90, 2003 (CEUR Workshop Proceedings)

[GZ03b] Grahne, Gösta ; Zhu, Jianfei: Efficiently using prefix-trees in mining frequent itemsets. In: *Proc. of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations*, 2003

[GZ03c] Grahne, Gösta ; Zhu, Jianfei: High Performance Mining of Maximal Frequent Itemsets. In: *Proc. of the 2003 SIAM Intl. Workshop on High Performance Data Mining*, 2003, S. 135–143

[Haa97] Haas, Peter: Large-Sample and Deterministic Confidence Intervals for Online Aggregation. In: *Statistical and Scientific Database Management*, 1997, S. 51–63

[HCC+97] Han, Jiawei ; Chiang, Jenny ; Chee, Sonny ; Chen, Jianping ; Chen, Qing ; Cheng, Shan ; Gong, Wan ; Kamber, Micheline ; Koperski, Krzysztof ; Liu, Gang ; Lu, Yijun ; Stefanovic, Nebojsa ; Winstone, Lara ; Xia, Betty ; Zaiane, Osmar ; Zhang, Shuhua ; Zhu, Hua: DBMiner: a system for data mining in relational databases and data warehouses. In: *Proc. of the 1997 Conf. of the Centre for Advanced Studies on Collaborative research*, 1997, S. 249–260

[He05] He, Zengyou: Mining Top-k Approximate Frequent Patterns. 2005 (TR-2005-0315). – Forschungsbericht

[HGN00] Hipp, Jochen ; Güntzer, Ulrich ; Nakhaeizadeh, Gholamreza: Algorithms for association rule mining — a general survey and comparison. In: *SIGKDD Exploration Newsletter* 2 (2000), Nr. 1, S. 58–64

[HH03] Hilderman, Robert ; Hamilton, Howard: Measuring the interestingness of discovered knowledge: A principled approach. In: *Intelligent Data Analysis* 7 (2003), Nr. 4, S. 347–382

[Hid99] Hidber, Christian: Online Association Rule Mining. In: *Proc. of the 1999 ACM SIGMOD Intl. Conf. on Management of Data*, 1999, S. 145–156

[HK05a] Han, Jiawei ; Kamber, Micheline: *Data Mining: Concepts and Techniques.* Zweite Edition. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2005. – S. 7

[HK05b] Han, Jiawei ; Kamber, Micheline: *Data Mining: Concepts and Techniques.* Zweite Edition. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2005. – S. 160

[HKSZ05] Hose, Katja ; Karnstedt, Marcel ; Sattler, Kai-Uwe ; Zinn, Daniel: Processing top-N queries in P2P-based web integration systems with probabilistic guarantees. In: *Proc. of the 2005 Intl. Workshop on Web and Databases (WebDB)*, 2005, S. 109–114

[HPY00] Han, Jiawei ; Pei, Jian ; Yin, Yiwen: Mining frequent patterns without candidate generation. In: *Proc. of the 2000 ACM SIGMOD Intl. Conf. on Management of Data*, 2000, S. 1–12

[HWLT02] Han, Jiawei ; Wang, Jianyong ; Lu, Ying ; Tzvetkov, Petre: Mining Top-K Frequent Closed Patterns without Minimum Support. In: *Proc. of the 2002 IEEE Intl. Conf. on Data Mining*, 2002, S. 211–218

[Ill] *IlliMine Project.* – `http://illimine.cs.uiuc.edu`

[IS56] Isaac, E. J. ; Singleton, R. C.: Sorting by Address Calculation. In: *Journal of the ACM* 3 (1956), Nr. 3, S. 169–174

[JYP03] Jia, Lei ; Yao, Jun ; Pei, Renqing: Mining Association Rules with Frequent Closed Itemsets Lattice. In: *Knowledge-Based Intelligent Information and Engineering Systems*, 2003, S. 469–475

[KGLB00] Kuntz, Pascale ; Guillet, Fabrice ; Lehn, Rémi ; Briand, Henri: A User-Driven Process for Mining Association Rules. In: *Proc. of the 2000 European Conf. on Principles of Data Mining and Knowledge Discovery*, 2000, S. 483–489

[KMR+94] Klemettinen, Mika ; Mannila, Heikki ; Ronkainen, Pirjo ; Toivonen, Hannu ; Verkamo, Inkeri: Finding interesting rules from large sets of discovered association rules. In: *Proc. of the 1994 Conf. on Information and Knowledge Management*, 1994, S. 401–407

[KP03] Kosters, Walter ; Pijls, Wim: Apriori, A Depth First Implementation. In: *Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003

[KRTR98] Kimball, Ralph ; Reeves, Laura ; Thornthwaite, Warren ; Ross, Margy: *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses*. Zweite Edition. New York, NY : John Wiley & Sons, Inc., 1998. – S. 139-218

[LD60] Land, A. H. ; Doig, A. G.: An automatic method of solving discrete programming problems. In: *Econometrica* 28 (1960), S. 497–520

[LFZ99] Lavrac, Nada ; Flach, Peter ; Zupan, Blaz: Rule Evaluation Measures: A Unifying View. In: *Intl. Workshop on Inductive Logic Programming*, 1999, S. 174–185

[LLR06] Legler, Thomas ; Lehner, Wolfgang ; Ross, Andrew: Data mining with the SAP NetWeaver BI accelerator. In: *Proc. of the 2006 Intl. Conf. on Very Large Data Bases*, 2006, S. 1059–1068

[LLR07] Legler, Thomas ; Lehner, Wolfgang ; Ross, Andrew: Der Einfluss der Datenverteilung auf die Performanz eines Data Warehouse. In: *BTW 2007: Datenbanksysteme in Business, Technologie und Web*, 2007, S. 502–513

[LLZT07] Liu, Li ; Li, Eric ; Zhang, Yimin ; Tang, Zhizhong: Optimization of frequent itemset mining on multiple-core processor. In: *Proc. of the 2007 Intl. Conf. on Very Large Data Bases*, 2007, S. 1275–1285

[LOP07] Lucchese, Claudio ; Orlando, Salvatore ; Perego, Raffaele: Parallel Mining of Frequent Closed Patterns: Harnessing Modern Computer Architectures. In: *Proc. of the 2007 IEEE Intl. Conf. on Data Mining*, 2007, S. 242–251

[LOPS05] Lucchese, Claudio ; Orlando, Salvatore ; Perego, Raffaele ; Silvestri, Claudio: Mining Frequent Closed Itemsets from Highly Distributed Repositories. In: *Knowledge and Data Mangement in GRIDs - Proc. of the First Worskshop on Knowledge and Data Management*, 2005, S. 62–65

[LSF09] Lemke, Christian ; Sattler, Kai-Uwe ; Färber, Franz: Kompressionstechniken für spaltenorientierte BI-Accelerator-Lösungen. In: *BTW 2009: Datenbanksysteme in Business, Technologie und Web*, 2009, S. 486–497

[LZBX06] Liu, Bing ; Zhao, Kaidi ; Benkler, Jeffrey ; Xiao, Weimin: Rule interestingness analysis using OLAP operations. In: *Proc. of the 2006 Intl. Conf. on Knowledge Discovery and Data Mining*, 2006, S. 297–306

[LZCZ07] Liu, Chun ; Zheng, Zheng ; Cai, Kai-Yuan ; Zhang, Shichao: Distributed Frequent Closed Itemsets Mining. In: *Proc. of the 2007 Intl. IEEE Conf. on Signal-Image Technologies and Internet-Based System*, 2007, S. 43–50

[Mac94] MacQueen, J.: Some methods of classification and analysis of multivariate observations. In: *Proc. of the 5th Berkley Symp. on Mathematical Statistics and Probability*, 1994, S. 281–297

[MCSW06] Miao, YuQing ; Chen, Guoliang ; Song, Bin ; Wang, ZhiHao: TP+Close: Mining Frequent Closed Patterns in Gene Expression Datasets. In: *VDMB, Data Mining and Bioinformatics, First Intl. Workshop*, 2006, S. 120–130

[MFM+98] Morimoto, Yasuhiko ; Fukuda, Takeshi ; Matsuzawa, Hirofumi ; Tokuyama, Takeshi ; Yoda, Kunikazu: Algorithms for Mining Association Rules for Binary Segmentations of Huge Categorical Databases. In: *Proc. of the 1998 Intl. Conf. on Very Large Data Bases*, 1998, S. 380–391

[Mil56] Miller, George: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. In: *Information. The Psychological Review* 63 (1956), S. 81

[MM02] Manku, Gurmeet S. ; Motwani, Rajeev: Approximate frequency counts over data streams. In: *Proc. of the 2002 Intl. Conf. on Very Large Data Bases*, 2002, S. 346–357

[MTW05] Michel, Sebastian ; Triantafillou, Peter ; Weikum, Gerhard: KLEE: a framework for distributed top-k query algorithms. In: *Proc. of the 2005 Intl. Conf. on Very Large Data Bases*, 2005, S. 637–648

[MWDI02] McDonald, Kevin ; Wilmsmeier, Andreas ; Dixon, David ; Inmon, William: *Mastering the SAP Business Information Warehouse*. Wiley Publishing Inc., 2002. – S. 100

[NLHP98] Ng, Raymond ; Lakshmanan, Laks ; Han, Jiawei ; Pang, Alex: Exploratory mining and pruning optimizations of constrained associations rules. In: *Proc. of the 1998 ACM SIGMOD Intl. Conf. on Management of Data*, 1998, S. 13–24

[NR99] Nepal, Surya ; Ramakrishna, M. V.: Query Processing Issues in Image (Multimedia) Databases. In: *Proc. of the 1999 Intl. Conf. on Data Engineering*, 1999, S. 22–29

[PBTL99] Pasquier, Nicolas ; Bastide, Yves ; Taouil, Rafik ; Lakhal, Lotfi: Discovering Frequent Closed Itemsets for Association Rules. In: *Lecture Notes in Computer Science* 1540 (1999), S. 398–416

[PCT+03] Pan, Feng ; Cong, Gao ; Tung, Anthony K. H. ; Yang, Jiong ; Zaki, Mohammed: Carpenter: finding closed patterns in long biological datasets. In: *Proc. of the 2003 Intl. Conf. on Knowledge Discovery and Data Mining*, 2003, S. 637–642

[PDZH04] Pei, Jian ; Dong, Guozhu ; Zou, Wei ; Han, Jiawei: Mining condensed frequent-pattern bases. In: *Knowledge and Information Systems* 6 (2004), Nr. 5, S. 570–594

[PHM00] Pei, Jian ; Han, Jiawei ; Mao, Runying: CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. In: *Proc. of the 2000 ACM SIGMOD Intl. Conf. on Management of Data*, 2000, S. 21–30

[PSF92] Piatetsky-Shapiro, Gregory ; Frawley, William J.: *Knowledge Discovery in Databases*. AAAI/MIT Press, 1992. – S. 229-249

[PTB+05] Pasquier, Nicolas ; Taouil, Rafik ; Bastide, Yves ; Stumme, Gerd ; Lakhal, Lotfi: Generating a Condensed Representation for Association Rules. In: *J. Intell. Inf. Syst.* 24 (2005), Nr. 1, S. 29–60

[Ros09] Ross, Andrew: *SAP NetWeaver BI Accelerator*. Galileo Press, 2009

[RSQ+08] Raman, Vijayshankar ; Swart, Garret ; Qiao, Lin ; Reiss, Frederick ; Dialani, Vijay ; Kossmann, Donald ; Narang, Inderpal ; Sidle, Richard: Constant-Time Query Processing. In: *Proc. of the 2008 Intl. Conf. on Data Engineering*, 2008, S. 60–69

[SA96] Srikant, Ramakrishnan ; Agrawal, Rakesh: Mining quantitative association rules in large relational tables. In: *SIGMOD Rec.* 25 (1996), Nr. 2, S. 1–12

[SA97] Srikant, Ramakrishnan ; Agrawal, Rakesh: Mining generalized association rules. In: *Future Generation Computer Systems* 13 (1997), Nr. 2–3, S. 161–180

[SAB+05] Stonebraker, Mike ; Abadi, Daniel ; Batkin, Adam ; Chen, Xuedong ; Cherniack, Mitch ; Ferreira, Miguel ; Lau, Edmond ; Lin, Amerson ; Madden, Sam ; O'Neil, Elizabeth ; O'Neil, Pat ; Rasin, Alex ; Tran, Nga ; Zdonik, Stan: C-store: a column-oriented DBMS. In: *Proc. of the 2005 Intl. Conf. on Very Large Data Bases*, 2005, S. 553–564

[SB75] Shortliffe, Edward ; Buchanan, Bruce: A model of inexact reasoning in medicine. In: *Mathematical Biosciences* 23 (1975), S. 351–379

[SG92] Smyth, Padhraic ; Goodman, Rodney: An Information Theoretic Approach to Rule Induction from Databases. In: *IEEE Transactions on Knowledge and Data Engineering* 4 (1992), Nr. 4, S. 301–316

[SON95] Savasere, Ashoka ; Omiecinski, Edward ; Navathe, Shamkant: An Efficient Algorithm for Mining Association Rules in Large Databases. In: *The VLDB Journal*, 1995, S. 432–444

[Spe04] Spearman, Charles: The Proof and Measurement of Association between Two Things. In: *The American Journal of Psychology* 15 (1904), Nr. 1, S. 72–101

[Sri] Srikant, Ramakrishnan: *Synthetic Data Generation Code for Associations and Sequential Patterns.* – `www.almaden.ibm.com/cs/quest`

[SS05a] Shang, Xuequn ; Sattler, Kai-Uwe: Depth-first frequent itemset mining in relational databases. In: *Proc. of the 2005 Annual ACM Symp. on Applied Computing*, 2005, S. 1112–1117

[SS05b] Shang, Xuequn ; Sattler, Kai-Uwe: Frequent Itemset Mining with Parallel RDBMS. In: *Advances in Knowledge Discovery and Data Mining*, 2005, S. 539–544

[ST96] Silberschatz, Avi ; Tuzhilin, Alexander: What makes patterns interesting in knowledge discovery systems. In: *IEEE Trans. On Knowledge And Data Engineering* 8 (1996), S. 970–974

[SV08] Schmidt-Volkmar, Pascal: *Betriebswirtschaftliche Analyse auf operationalen Daten.* Gabler Verlag, 2008. – S. 193

[SVA97] Srikant, Ramakrishnan ; Vu, Quoc ; Agrawal, Rakesh: Mining association rules with item constraints. In: *Proc. of the 1997 Intl. Conf. on Knowledge Discovery and Data Mining*, 1997, S. 67–73

[TCRB06] Toon Calders, T. ; Rigotti, Christophe ; ; Boulicaut, Jean-Francois: A Survey on Condensed Representations for Frequent Sets. In: Boulicaut, J-F. (Hrsg.) ; de Raedt, L. (Hrsg.) ; Mannila, H. (Hrsg.): *Constraint-Based Mining* Bd. 3848. Springer, 2006

[TKR+95] Toivonen, Hannu ; Klemettinen, Mika ; Ronkainen, Pirjo ; Hatonen, Kimmo ; Mannila, Heikki: Pruning and grouping of discovered association rules. In: *In Workshop Notes of the ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases*, 1995, S. 47–52

[TKS02] Tan, Pang-Ning ; Kumar, Vipin ; Srivastava, Jaideep: Selecting the right interestingness measure for association patterns. In: *Proc. of the 2002 Intl. Conf. on Knowledge Discovery and Data Mining*, 2002, S. 32–41

[Toi96] Toivonen, Hannu: Sampling Large Databases for Association Rules. In: *Proc. of the 1996 Intl. Conf. on Very Large Data Bases*, 1996, S. 134–145

[TSK06a] Tan, Pang-Ning ; Steinbach, Michael ; Kumar, Vipin: *Introduction to Data Mining.* Addison Wesley, 2006. – S. 359-363

[TSK06b] Tan, Pang-Ning ; Steinbach, Michael ; Kumar, Vipin: *Introduction to Data Mining.* Addison Wesley, 2006. – S. 370-382

[TWS04] Theobald, Martin ; Weikum, Gerhard ; Schenkel, Ralf: Top-k query evaluation with probabilistic guarantees. In: *Proc. of the 2004 Intl. Conf. on Very Large Data Bases*, 2004, S. 648–659

[UHB01] Unwin, Antony ; Hofmann, Heike ; Bernt, Klaus: The TwoKey Plot for Multiple Association Rules Control. In: *Proc. of the 2001 European Conf. on Principles of Data Mining and Knowledge Discovery*, 2001, S. 472–483

[VDNV08] Vlachou, Akrivi ; Doulkeridis, Christos ; Nørvåg, Kjetil ; Vazirgiannis, Michalis: On efficient top-k query processing in highly distributed environments. In: *Proc. of the 2008 ACM SIGMOD Intl. Conf. on Management of Data*, 2008, S. 753–764

[Ver07] Vertica Systems: *The Vertica Database. Technical Overview Whitepaper. http://www.vertica.com.* 2007

[WHP03] Wang, Jianyong ; Han, Jiawei ; Pei, Jian: CLOSET+: searching for the best strategies for mining frequent closed itemsets. In: *Proc. of the 2003 Intl. Conf. on Knowledge Discovery and Data Mining*, 2003, S. 236–245

[WMB99a] Witten, Ian ; Moffat, Alistair ; Bell, Timothy: *Managing Gigabytes. Compressing and Indexing Documents and Images.* Zweite Edition. Academic Press, 1999. – S. 431-450

[WMB99b] Witten, Ian ; Moffat, Alistair ; Bell, Timothy: *Managing Gigabytes. Compressing and Indexing Documents and Images.* Zweite Edition. Academic Press, 1999. – S. 105-153

[XHYC05] Xin, Dong ; Han, Jiawei ; Yan, Xifeng ; Cheng, Hong: Mining compressed frequent-pattern sets. In: *Proc. of the 2005 Intl. Conf. on Very Large Data Bases*, 2005, S. 709–720

[YHN06] Yahia, S. B. ; Hamrouni, T. ; Nguifo, E. M.: Frequent closed itemset based algorithms: a thorough structural and analytical survey. In: *SIGKDD Exploration Newsletter* 8 (2006), Nr. 1, S. 93–104

[YZ99] Yao, Yiyu ; Zhong, Ning: An Analysis of Quantitative Measures Associated with Rules. In: *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 1999, S. 479–488

[Zak99] Zaki, Mohammed: Parallel and Distributed Association Mining: A Survey. In: *IEEE Concurrency* 7 (1999), Nr. 4, S. 14–25

[Zak04] Zaki, Mohammed: Mining Non-Redundant Association Rules. In: *Data Mining Knowledge Discovery* 9 (2004), Nr. 3, S. 223–248

[ZG03] Zaki, Mohammed ; Gouda, Karam: Fast vertical mining using diffsets. In: *Proc. of the 2003 Intl. Conf. on Knowledge Discovery and Data Mining*, 2003, S. 326–335

[ZH99] Zaki, Mohammed ; Hsiao, Ching-Jui: CHARM: An efficient algorithm for closed association rule mining / Computer Science, Rensselaer Polytechnic Institute. 1999 (TR99-10). – Forschungsbericht

[ZKM01] Zheng, Zijian ; Kohavi, Ron ; Mason, Llew: Real World Performance of Association Rule Algorithms. In: *Proc. of the 2001 Intl. Conf. on Knowledge Discovery and Data Mining*, 2001, S. 401–406

[ZPOL97] Zaki, Mohammed ; Parthasarathy, Srinivasan ; Ogihara, Mitsunori ; Li, Wei: New Algorithms for Fast Discovery of Association Rules. 1997 (TR651). – Forschungsbericht

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Dissertation zum Thema

Datenzentrierte Bestimmung von Assoziationsregeln in parallelen Datenbankarchitekturen

selbstständig und unter Angabe aller Zitate und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Ich bin mir bewusst, dass eine unwahre Erklärung rechtliche Folgen haben wird.

Walldorf, den 8. April 2009