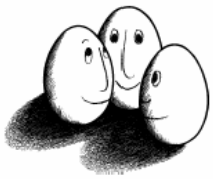
Diplomarbeit

Beschreibung von Web-Nutzungsverhalten unter Verwendung von Data Mining Techniken

Irina Alesker



Diplomarbeit
am Fachbereich Informatik
der Universität Dortmund

23. Juni 2005

Betreuer:

Prof. Dr. Katharina Morik
Dipl.-Inf. Stefan Rüping

Inhaltsverzeichnis

1 Einleitung 1

1.1 Problemstellung 1
1.2 Web Usage Mining 2
1.2.1 Übersicht 2
1.2.2 Arbeiten im Bereich Web Usage Mining 6
1.3 Ziele 6
1.4 Gliederung 9

2 Theoretische Grundlagen 11

2.1 Assoziationsregeln (AR) 11
2.2 Apriori 12
2.2.1 Entdeckung von häufigen Itemsets 12
2.3 Repräsentative Assoziationsregeln (*RAR*) 14
2.4 Abgeschlossene Itemsets (Closed Itemsets) 16
2.5 Algorithmen 20
2.5.1 Generierung von häufigen abgeschlossenen Itemsets 20
2.5.2 Generierung des Abschlusses 21
2.5.3 Generierung von Kandidaten-Itemsets 22
2.5.4 Generierung von Repräsentativen Assoziationsregeln (*RAR*) 24
2.6 WINEPI 26
2.6.1 Ereignisfolgen 26
2.6.2 Episoden 27
2.6.3 Algorithmen 28
2.7 Entdeckung von Interaktionsmustern 32
2.7.1 Aufgabenstellung 33
2.7.2 Algorithmusbeschreibung 34

3 Datenanalyse 41

3.1 Ansatz 41
3.2 Daten 41
3.3 Preprocessing 42
3.4 Statistische Analyse 43
3.4.1 Besuchstatistik/Bestellstatistik 44
3.4.2 Startseiten-Statistik 48
3.4.3 Abbruchstatistik 49

3.4.4 Verweilstatistik 52
3.5 Gemeinsame Seitenmengen 55
3.5.1 Apriori 55
3.5.2 WINEPI 57
3.5.3 Frequent Closed Sequence Itemsets 58
3.5.4 IPM2 (Interaction-Pattern Mining) 67
3.5.5 Einteilung der Benutzer in Gruppen 72
3.6 Vergleich mit der erwarteten Nutzung 82
3.6.1 Unerwartete Nutzung 82
3.6.2 Visualisierung der Website-Nutzung 88
3.6.3 Nicht geplante Links 92

4 Zusammenfassung 95

4.1 Fazit 95
4.2 Ausblick 97

Literaturverzeichnis 99

Abbildungsverzeichnis

Abb. 2.2.1: *Apriori*-Algorithmus [2] 13
Abb. 2.4.1: häufige abgeschlossene und maximale Itemmengen [49] 19
Abb. 2.5.1: Algorithmus *Close-FCI*(*D*) [42] 21
Abb. 2.5.2: Algorithmus *Generate-Closures*(FCC_k) [42] 22
Abb. 2.5.3: Algorithmus *Generate-Candidates*(FC_k) [42] 23
Abb. 2.5.4: Algorithmus *Generate-RAR*(all frequent closed Itemsets FC) [42] 24
Abb. 2.6.1: Beispiel einer Ereignisfolge 26
Abb. 2.6.2: Hauptalgorithmus *WINEPI* [28] 29
Abb. 2.6.3: Algorithmus 2 Generierung von häufigen Episoden [28] 29
Abb. 2.6.4: Der Algorithmus *Kandidatengenerierung* [28] 30
Abb. 2.7.1: Preprocessing von Interaktionsfolgen [14] 35
Abb. 2.7.2: *Prozedur 1* von *IPM2* [14] 36
Abb. 2.7.3: *Prozedur 2* von *IPM2* [14] 37
Abb. 2.7.4: *Prozedur 3* von *IPM2* [14] 39
Abb. 3.4.1: 30 Top-Seiten der Statistik “#User pro Seite“ 46
Abb. 3.4.2: 30 Top-Seiten der Statistik “#Sessions pro Seite“ 47
Abb. 3.4.3: Bestellstatistik 48
Abb. 3.4.4: 7 Startseiten 48
Abb. 3.4.5: Ausschnitt aus der Abbruchstatistik(#User pro Abbruchseite) 50
Abb. 3.4.6: Ausschnitt aus der Abbruchstatistik(#Sessions pro Abbruchseite) 51
Abb. 3.4.7: Verweilstatistik (Fort.) 53
Abb. 3.5.1: Beispiele der Supermengen 56
Abb. 3.5.2: Beispiele der gemeinsamen Seitenmengen mit minimalem Support 5 60
Abb. 3.5.3: Beispiele der abgeschlossenen Itemmengen (keine Teilmengen der Bestellmengen) 62
Abb. 3.5.4: Beispiele von Items, die in “keiner“ Bestellmenge vorhanden sind 62
Abb. 3.5.5: Beispiel einer Itemmenge in zeitlicher Abfolge 64
Abb. 3.5.6: Nutzungsmodelle beim Bestellen des Produktes A 64
Abb. 3.5.7: Nutzungsmodelle mit Besuchzahl und Verweildauer 66
Abb. 3.5.8: Nutzungsmodelle der Nichtbesteller mit Besuchzahl und Verweildauer 66
Abb. 3.5.9: Beispiele von sequentiellen Mustern mit (in oberer Zeile) und ohne Wiederholungen (in unterer Zeile) 69
Abb. 3.5.10: Bespiele der (Teil)Sequenzen (keine “Bestellsequenzen“) 69

Abb. 3.5.11: (Teil)Sequenzen mit Bestellseite "*43*" aus der Musteranalyse 70
Abb. 3.5.12: Hauptpfade aus der "abgeschlossenen Itemmengen"–Analyse 71
Abb. 3.5.13: Hauptpfade aus der sequentiellen Analyse 71
Abb. 3.6.1: Pfade zu selten genutzten Seiten 84
Abb. 3.6.2: Beispiele der seltenen abgeschlossenen Itemsets 84
Abb. 3.6.3: Sitemap (Ausschnitt) 86
Abb. 3.6.4: Beispiele der Abbruchmuster 87
Abb. 3.6.5: Seitenmengen, die häufiger vorkommen als die Bestellung 87
Abb. 3.6.6: Beispiele der (Teil)Pfade mit sich wiederholenden Seitenbesuchen 88
Abb. 3.6.7: Sitemap mit Übergangswahrscheinlichkeiten (Ausschnitt) 89
Abb. 3.6.8: Sitemap mit "wahrscheinlichen" Pfaden (Ausschnitt) 89
Abb. 3.6.9: Pfaddarstellung in einer html-Datei 90
Abb. 3.6.10: Ausschnitt aus einem Präfixbaum 91
Abb. 3.6.11: Präfixbaum mit Übergangswahrscheinlichkeiten (Ausschnitt) 92
Abb. 3.6.12: nicht geplanter Link c → d 93
Abb. 3.6.13: Beispiele der nicht geplanten Links 93

Tabellenverzeichnis

Tabelle 2.4.1: Beispiel eines *Data Mining Kontext* [42] 17
Tabelle 2.4.2: Beispiel einer Datenbank mit häufigen Itemmengen [49] 18
Tabelle 3.5.1: Ergebnisse von Apriori 55
Tabelle 3.5.2: häufige abgeschlossene Itemmengen 58
Tabelle 3.5.3: Ergebnisse nach dem IPM2-Algorithmus 68
Tabelle 3.5.4: Benutzereinteilung nach bestellten Produkten 72
Tabelle 3.5.5: aus den häufigen abgeschlossenen Itemmengen abgeleitete Regeln 73
Tabelle 3.5.6: Klassifikation der Sessions nach jeweils einer Regel 74
Tabelle 3.5.7: Klassifikation der Sessions nach mehreren Regeln 75
Tabelle 3.5.8: Klassifikation der Besucher nach mehreren Regeln 76
Tabelle 3.5.9: Durchschnittsfehler nach 10-facher Kreuzvalidierung 76
Tabelle 3.5.10: Klassifikationsbewertung 77
Tabelle 3.5.11: Klassifikationsbewertung nach 10-facher Kreuzvalidierung 77
Tabelle 3.5.12: Klassifikation der Sessions nach einer Regel 79
Tabelle 3.5.13: Klassifikation der Besucher nach einer Regel 79
Tabelle 3.5.14: Durchschnittsfehler nach 10-facher Kreuzvalidierung 80
Tabelle 3.5.15: Klassifikationsbewertung 81
Tabelle 3.5.16: Klassifikationsbewertung nach 10-facher Kreuzvalidierung 81

Kapitel 1

Einleitung

1.1 Problemstellung

Das World Wide Web enthält eine enorme Menge von Informationen in der Form eher unstrukturierter Sammlung verlinkter Dokumente, was das Auffinden von relevanten Dokumenten mit nützlicher Information zunehmend zu einer wichtigen Aufgabe macht [26]. Die Übermenge von Daten wird als Situation des Datenreichtums und der Informationsarmut bezeichnet [31]. Es wird immer schwieriger, nützliche Information zu finden. Die Nutzer wollen die relevante Information schnell und exakt finden. Zu jedem Zeitpunkt kann jede Website von mehreren Benutzern mit verschiedenen Zielen besucht werden, die an unterschiedlichen Informationen oder Präsentationsarten interessiert sind. Selbst dieselbe Person besucht zu einem anderen Zeitpunkt die gleiche Website mit anderem Vorhaben [31]. Als Ergebnis kann eine einfache Organisation der Website verschiedenen Bedürfnissen nicht gerecht werden [47]. Die Nutzer bevorzugen die Websites, die entsprechend ihren Bedürfnissen vorbereitet werden. Deshalb wird das Problem der dynamischen Gestaltung der Website, um an die individuellen Präferenzen der Besucher anzupassen, zu einer Herausforderung für die Anbieter und zu einem interessanten Forschungsobjekt [32].

Die Unternehmen, die ihre Leistungen über Internet anbieten, wollen, dass die Anzahl der Besucher und die Zeit, die Kunden auf ihrer Website verweilen, ständig steigen.

So entsteht eine Kluft zwischen Besuchern und Anbietern der Website. Nutzer wissen nicht, wie man an die nützlichen und genauen Informationen gelangt und dabei schnell und intuitiv dem Pfad durch die Website folgt [31]. Die Anbieter müssen aufgrund des zunehmenden Wettbewerbs Kunden immer wieder aufs Neue gewinnen. Um Kunden an sich zu binden, müssen sie ein attraktives Produkt bereitstellen. Und um

die Website attraktiver darzustellen, brauchen Webdesigner und Anbieter die Bedürfnisse und Gewohnheiten der potentiellen Kunden zu kennen, um die Website entsprechend der Erwartungen der zukünftigen Besucher und, soweit es möglich ist, entsprechend der individuellen Vorlieben zu gestalten [32].

Traditionelle Methoden der Datensammlung über die Softwarebenutzer, wie Befragungen und Begutachten sind nicht geeignet, um die Websitekunden zu verstehen. Die Menge der potentiellen Kunden ist zu groß und variiert ständig, und Menschen besuchen die Website in der Regel, bevor sie endlich zu regelmäßigen Nutzern werden. Die Alternative ist, Daten über ihre Besuche zu sammeln und zu analysieren, um zu verstehen, was die Kunden erwarten, und so die Website anzupassen und den gewünschten Inhalt auf eine einfache und leichtere, übersichtlichere Weise bereitzustellen [32].

Die Analyse des Benutzerverhaltens liefert wichtige Erkenntnisse, wie man Website rekonstruieren kann, um effektivere Unternehmenspräsenz zu schaffen, bzw. welche Maßnahmen sinnvoll und/oder notwendig sind, um den Ansprüchen der Kunden gerecht zu werden.

1.2 Web Usage Mining

1.2.1 Übersicht

Web Usage Mining hat das Ziel, Techniken und Werkzeuge zu entwickeln, um das Verhalten und die Nutzungsmodelle der Besucher zu studieren [32]. Mit diesen Methoden werden die auf die Nutzung der Seiten[1] bezogenen Daten wie IP-Adresse, Seitenreferenzen, Datum und Zeit des Zugriffs untersucht [12], um häufige Zugriffsmodelle zu entdecken. Diese Zugriffsdaten werden aus den Logfiles gewonnen.

Anhand der Logfiles-Daten kann man z.B. Statistiken über Anzahl der Zugriffe innerhalb einer Zeitspanne oder über Zugriffe auf Web-Seiten erstellen, sie geben einen Überblick, wie und von welchen Personengruppen Web-Applikationen genutzt werden.

Statistiken über Besuche der einzelnen Seiten bieten erste Hilfe zur Erforschung des Userverhaltens. So werden z.B. Hit-Seiten ermittelt, d.h. Seiten mit der großen Anzahl der Besucher, und so stellt man schnell fest, ob die Seiten mit dem wichtigen Inhalt von Kunden angeschaut werden. Ist die Anzahl solcher Besucher relativ groß, kann der Betreiber

[1] engl. page

sicher sein, dass Kunden wenigstens Zugriff auf die wichtigen Informationen finden, weiß aber nicht, wie Personen auf diese Seiten kommen; der Anbieter kann nur vermuten - wenn auch Zwischenseiten dieses Pfades oft besucht wurden - dass Kunden dem kürzesten, vom Web-Designer geplanten Pfad gefolgt haben. Sollten die Seiten mit wichtigen Informationen nur von wenigen Personen angeschaut werden, möchte der Anbieter den Grund wissen, warum dies der Fall ist. Die Statistik liefert nur den Hinweis darauf, wie groß ist der Benutzeranteil, der die Seiten angeklickt hat; anhand der Statistik kann man nicht feststellen, ob Besucher tatsächlich zu diesen Informationen gelangen wollten oder es sich um zufällige Zugriffe handelt.

Insbesondere wenn man den Grund erfahren muss, warum Benutzer zu den wichtigen Informationen nicht gelangen, ist es notwendig zu wissen, wie sie durch die Website navigieren, speziell an welchen Stellen und warum sie ihre Sitzungen abbrechen. Es ist leider unmöglich zu unterscheiden, ob der Kunde schließlich zu gesuchten Informationen gelangen konnte und deswegen die Sitzung beendet hat oder er keine ihn interessierenden Informationen finden konnte und aus diesem Grund die Suche abgebrochen hat. Man kann aber durch Analyse des von ihm gefolgten Pfades annehmen, welche Seiten anscheinend das Interesse des Kunden geweckt haben.

Es ist deswegen von großer Bedeutung, die Zugriffsmodelle der Kunden zu untersuchen. So findet man heraus, welche Seiten sich als weiterführend erweisen und von Besuchern als hilfreich bei der Suche nach dem wichtigen Inhalt eingestuft werden und welche nur als störend empfunden werden, wenn man die gewünschten Informationen nicht finden konnte und angewiesen ist, zurück zu gehen und einen anderen Weg einzuschlagen. Beim Vergleich der Klickfolge und der Sitetopologie kann der Web-Designer solche hilfreiche und störende Seiten finden.

Das Studieren der Zugriffsfolge jedes einzelnen Benutzers wird aber zu einem unlösbaren Problem, deswegen werden diese Folgen zuerst zu den Zugriffsmodellen zusammengefasst und dann mit der Sitetopologie verglichen. Dadurch kann Web-Designer sehen, welchen Pfaden Kunden folgen, und solche Seiten finden, wo User vom geplanten Pfad abweichen oder welche Klicks zu den Fehlermeldungen führen.

Ein Zugriffsmodell ist ein sich wiederholendes sequentielles Muster unter den Einträgen im Logfile. Wenn z.B. verschiedene Benutzer immer wieder auf dieselbe Reihe von Seiten zugreifen, kommen entsprechende Folgen in den Logfiles vor, und so können sie als Zugriffsmodelle betrachtet werden [32].

In ihren täglichen Aktivitäten sammeln Unternehmen eine große Menge von Daten, die automatisch vom Webserver generiert werden, und speichern sie in den Serveraccesslogs. Andere Quellen für die Benutzerinformationen sind spezielle Logfiles, die Informationen über jede angezeigte Seite, über Benutzerregistrierung und durch Skripte ausgelesene Prüfdaten enthalten.

Die Analyse der Logfiles liefert dem Webdesigner Kenntnisse, wie Benutzer üblicherweise durch die Website navigieren, d.h. über das Verhalten jedes einzelnen Benutzers. Dies ist ein Schlüssel bei der Optimierung der Website und somit ein Schlüssel zum Erfolg in der Befriedigung der Bedürfnisse der Kunden.

Verschiedene Arten der Entdeckung von Zugriffsmodellen werden in Abhängigkeit von Zielen des Erforschers, wie Pfadanalyse, Erforschung der Assoziationsregeln und sequentiellen Mustern, und Gruppeneinteilung und Klassifikation, dargestellt.

Es gibt verschiedene Arten von Graphen, die zur Darstellung der Pfadanalyse konstruiert werden können, weil durch einen Graphen Beziehungen zwischen den Web Seiten anschaulich repräsentiert werden können. Meist einleuchtend ist ein Graph, der das physische Layout der Website repräsentiert, mit Web Seiten als Knoten und Links zwischen Seiten als gerichteten Kanten [27]. Andere Graphen können basierend auf den Typen der Web-Seiten mit Kanten, die Ähnlichkeit zwischen den Seiten repräsentieren, konstruiert werden oder erzeugen Kanten mit Angabe der Benutzeranzahl, die von einer Seite zu anderen gehen [40]. Die Pfadanalyse kann zur Ermittlung der meist besuchten Pfade in der Website genutzt werden. Außerdem liefert diese Analyse Regeln, die darauf hinweisen, durch welche Seiten User häufig auf die nützlichen Informationen kommen, welche Seiten sie als Startseite nutzen oder wie viele Seiten die meisten Kunden im Schnitt auf der Website anklicken.

Methoden der Entdeckung der Assoziationsregeln [2,41] werden generell auf die Datenbanken der Transaktionen angewandt, wo jede Transaktion aus einer Menge von Items besteht. Man findet hier alle Assoziationen und Korrelationen zwischen den Items, wo das Vorhandensein einer Menge von Items in einer Transaktion das Vorhandensein anderer Items (mit einem bestimmten Sicherheitsgrad) impliziert. Im Kontext der Web Usage Mining bedeutet dies die Entdeckung der Korrelationen zwischen Referenzseiten; jede Transaktion besteht aus einer Menge der URLs, auf die in einer Sitzung zugegriffen wurde [41]. Und so findet man Korrelationen zwischen einzelnen Web Seiten.

Die Entdeckung der Assoziationsregeln kann für Unternehmen, die sich mit dem e-Commerce beschäftigen, bei der Entwicklung der effektiven

Marketingstrategien hilfreich sein. Zusätzlich können Assoziationsregeln Hinweise geben, wie man die Webpräsenz am besten organisiert, d.h. wo es sinnvoller wäre, bestimmte Inhalte zu platzieren.

Mit Methoden der Entdeckung der sequentiellen Muster [28,45] werden Regeln gefunden, die strenge sequentielle Abhängigkeiten zwischen verschiedenen Ereignissen vorhersagen. Es werden z.B. die Zugriffsfolgen der Kunden, die ein oder mehrere Produkte bestellt haben, analysiert und oft auftretende Muster identifiziert. Die Anbieter wollen die Gemeinsamkeiten der Besucher aufspüren, die eine bestimmte Seite innerhalb der vorgegebenen Zeitspanne anklicken. Oder umgekehrt, Anbieter können an einem Zeitintervall interessiert sein, währenddessen die meisten Zugriffe auf eine bestimmte Seite stattfinden [27]. So kann das zukünftige Kaufmodell vorhersagt werden, was sehr hilfreich z.B. beim Plazieren der an bestimmte Benutzergruppen gerichtete oder der zeitabhängigen Werbung sein.

Ähnlich kann die Analyse der Seitenfolgen zur Vorhersage des zukünftigen Verhaltens des Kunden angewandt und Vorhersagen als Empfehlungen angeboten werden. Anhand der häufigen Nutzungsmodelle können Usergruppen mit entsprechenden Nutzungsmustern verknüpft werden, so dass Benutzer entsprechend dem Muster durch die Website geführt werden.

Auf der Basis der demographischen Charakteristiken oder der Zugriffsmodelle ermöglichen Klassifikationstechniken den Anbietern Profile ihrer Kunden zu bilden, die auf bestimmte Seiten zugreifen. So können als Klassifikationskriterien das Alter oder die Branchenzugehörigkeit der Kunden gewählt werden.

Durch die Cluster-Analyse [19,34] können User oder Items, die ähnliche Charakteristiken haben, gruppiert werden. So erleichtert man die Entwicklung und Realisierung der zukünftigen Marktstrategien, sowohl online als auch offline, wie automatische Post an die Besucher, die bereits in bestimmte Gruppen eingeteilt worden, oder dynamische Weiterleitung an eine Seite anhand der letzten Klassifikation des Besuchers.

Die Nutzung der Website kann durch einen Präfix-Baum dargestellt werden. Jeder Knoten trägt in sich Information, wie viele Benutzer die assoziierte Seite genau durch den zu diesem Knoten führenden Pfad erreicht haben und jede Kante zeigt, wie viele Benutzer auf dem entsprechenden Weg genau diese 2 durch die Kante verbundenen Seiten hintereinander besucht haben. So kann man problematische Stellen der Website erkennen.

Durch Analyse der Klicks kann außerdem ein Graph mit Übergangswahrscheinlichkeiten erstellt werden; ebenfalls wie Präfix-Baum dient dieser dem Website-Designer als Werkzeug zur Erforschung des Nutzverhaltens der Kunden.

1.2.2 Arbeiten im Bereich Web Usage Mining

In den letzten Jahren ist die ganze Reihe von Forschungsarbeiten auf dem Gebiet Web Usage Mining erschienen [8,9,44,3,5,16]. Die Hauptmotivation dieser Arbeiten war, Aktivitäten und Motivationen von Benutzern besser zu verstehen.

WEBMINER [26,12] entdeckt Assoziationsregeln und sequentielle Muster in den Serveraccesslogs. Manche Studien haben ihre Ergebnisse zur Verbesserung des Website-Designs angewendet [24,38,39] oder stellen Empfehlungssysteme vor, die HTML-Dokumente anhand der Userprofile dynamisch generieren [30]. WebWatcher [18], SiteHelper [35], Letizia [22] und Arbeiten von Mobasher et. al. [25] und Yan et. al. [46] konzentrieren sich alle auf der Bereitstellung von Website-Personalisierung basierend auf den Nutzungsdaten.

Büchner und Mulvenna [6] haben einen Wissens Entdeckungsprozess präsentiert, mit dem sich für das Marketing relevante Informationen gewinnen lassen. Padmanabhan et. al. [36]. nutzen Web Server Logs, um eine Vorstellung über die Zugriffsmuster von Webkunden der bestimmten Website zu beschaffen.

Die in [7] gesammelten Ergebnisse liefern detaillierte Information über Interaktion der Benutzer mit dem Browser sowie über Navigationsstrategie, die beim Zugriff auf eine bestimmte Website genutzt wird. Chi et. al. [10] beschreiben ein System(Web Ecology and Evolution Visualization), ein Visualisierungstool, um die Beziehungen zwischen Webnutzung, Inhalt und Sitetopologie in Bezug auf eine Zeitperiode zu studieren.

Einige Arbeiten zeigen, wie Nutzungsinformationen zur Entwicklung der Webcaching-Strategie, der Übertragung in Netzwerken [11] oder zu Informationsverteilung genutzt werden können.

1.3 Ziele

Ziel dieser Arbeit ist, die Einsetzbarkeit der verschiedenen Lernmethoden für die Analyse der Nutzung einer Web-Applikation zu testen.

Dafür werden die mir zu Verfügung gestellten Logfiles einer Web-Applikation eines Dienstleistungsunternehmens ausgewertet.

Als erster Schritt bei der Analyse der Website-Nutzung werden Statistiken erstellt:

- Besucherstatistik zu jeder einzelnen Web-Seite
- Bestellungsstatistik
- Statistik zu Web-Seiten, die als Startseiten genutzt wurden
- Statistik zu Web-Seiten, die als letzte in der Sitzung aufgerufen wurden
- Verweilstatistik

Die Statistiken beschreiben die Häufigkeit der Seitennutzung, liefern jedoch relativ wenige Informationen über das Navigations- bzw. Nutzungsverhalten der Kunden (z.B. welche Seiten werden häufig zusammen aufgerufen werden) und erlauben keine Schlüsse über die Benutzermerkmale. Deswegen werden weitere Analyseschritte benötigt.

Es wird untersucht, welche Charakteristiken Kunden haben, die Bestellungen vornehmen und welche Charakteristiken diejenigen haben, die sich für ein Produkt Interesse gezeigt, aber trotzdem nicht bestellt haben. Die Gemeinsamkeiten werden durch Analyse der Klickfolgen festgestellt, d.h. es werden für beide Gruppen (Besteller und Nichtbesteller) gemeinsame URLs gefunden. Diese Korrelationen zwischen Referenzseiten werden zuerst mit Hilfe des Apriori-Algorithmus ermittelt. Da diese Methode die gemeinsamen URLs als ungeordnete Menge von Objekten liefert, aber ihre Darstellung in Form einer Seitensequenz bessere Interpretierbarkeit erlaubt, werden sie mit weiteren Verfahren wie Close-FCI-Algorithmus[2], WINEPI und IPM(Intertaction-Pattern Mining)-Algorithmus gelernt und mit der erwarteten Nutzung verglichen. Es ist sehr wahrscheinlich, dass es keine Abhängigkeiten gefunden werden, die dem Webentwickler ein neues, interessantes Wissen von nicht geplanten Modellen vermitteln, weil die Bewegung jedes Kunden durch die Website beim Bestellen eines Produktes extrem eingeschränkt ist. Man kann aber versuchen, solche Sequenzen herauszufinden, die ein typisches Abbruchmuster darstellen. Die Ergebnisse beschreiben dann – in Abhängigkeit von dem Verfahren – Klickpfade bzw. eine Menge der gemeinsam auftretenden URLs für beide Gruppen, so wird die Einteilung der Kunden in die Cluster möglich. Man

[2] Frequent Closed Itemsets

sollte bei der Klassifikation solche Kriterien finden, die möglichst frühe und relativ sichere Vorhersage erlauben. Jeder Cluster stellt dann ein Nutzungsmodell dar, das ein bestimmtes Kundenverhalten beschreibt. Die Methoden unterscheiden sich in der Art, in welcher Form sie die gefundenen Seitenmengen wiedergeben und welche Einschränkungen sie für die Eingabesequenzen fordern. Deswegen dienen als Gütekriterien für diese Verfahren die Eignung der Methoden zur Analyse der Klickfolgen, d.h. sie sollten möglichst unempfindlich gegen eventuelle Störungen in der Eingabemenge und für verschiedene Eingabeparameter wie Häufigkeit und Konfidenz möglichst gleich genau und gut sein, sowie Interpretierbarkeit der Ergebnismenge (d.h. Datenmengegröße, Redundanz) der entsprechenden Algorithmen. Außerdem werden Methoden in der Hinsicht bewertet, wie gut die Ergebnisse die tatsächlichen Klickfolgen widerspiegeln und inwiefern und wie genau die Einteilung mit den Korrelationsmengen als Kriterien in disjunktive Cluster möglich ist.

Außerdem wird dem Web-Designer in einem Visualisierungstool vorgestellt, wie die Website von den Kunden genutzt wird. Dies geschieht durch das Präsentieren der Website in Form eines Graphen, wobei der Graph selbst die Websitestruktur mit Links als gerichteten Kanten mit den Übergangswahrscheinlichkeiten widerspiegelt. Es wird auch analysiert, welche Pfade sowie Teilsequenzen und wie oft genutzt wurden. Pfade (bzw. Teilsequenzen) werden mit Methoden zur Entdeckung der sequentiellen Muster ermittelt. Besonders interessant für den Web-Designer ist, Pfade, die nicht genutzt wurden, und direkte Links, die nicht geplant wurden zu finden. Diese werden durch den Vergleich der sequentiellen Muster mit der vorhandenen Sitemap ermittelt. Die genutzten bzw. nicht genutzten Pfade und nicht geplanten Links werden auf 2 Arten dargestellt: in Form eines Präfix-Baumes und einer Tabelle, wo alle (Teil)Sequenzen aufgelistet sind. Um dem Web-Entwickler die “nicht genutzte Pfade“ bzw. “nicht geplante Links“ aufzuzeigen, wird nach bestimmten Kriterien gesucht, wie z.B. von welchem Anteil der Kunden diese genommen werden, so dass eine möglichst gute Klassifikation (d.h. mit einem relativ hohen Genauigkeitsgrad) erreicht wird. Die Darstellungsarten werden in Bezug auf bessere Interpretierbarkeit bewertet.

Man sollte bei den Darstellungen aller Ergebnisse möglichst auf die Informationen, die dem Web-Entwickler bereits bekannt sind, verzichten oder ihm eine Möglichkeit geben, diese rauszufiltern.

Das Ziel ist erreicht, wenn es Charakteristiken für die Besteller bzw. „Nicht-Besteller“ sowie Benutzergruppen gefunden und Empfehlungen bzw. problematische Stellen der Website dem Web-Designer präsentiert

werden können, die er beim Rekonstruieren der Website zwecks der besseren Kundenakzeptanz berücksichtigen sollte.

1.4 Gliederung

Die Diplomarbeit ist wie folgt aufgebaut:

Im folgenden Kapitel werden theoretische Grundlagen zur Entdeckung von Nutzungsmodellen und Assoziationsregeln wie Apriori, WINEPI und andere vorgestellt.

Im nächsten Kapitel wird gezeigt, wie die im Kapitel 2 beschriebenen Verfahren zur Analyse einer Web-Applikation eingesetzt werden und Ergebnisse präsentiert und bewertet. Ferner wird vorgestellt, wie verschiedene interessante Informationen visuell dargestellt werden können.

Im letzten Abschnitt der Diplomarbeit werden Ergebnisse zusammengefasst. Da werden Bewertungen des Dienst-Anbieters und Ausblick beschrieben.

Kapitel 2

Theoretische Grundlagen

In diesem Kapitel werden Data Mining Methoden zur Entdeckung der gemeinsam auftretenden URLs und der Interaktionsmuster und Abhängigkeiten zwischen den URLs in Form von Assoziationsregeln vorgestellt.

2.1 Assoziationsregeln (AR)

Das Problem der Entdeckung von Assoziationsregeln wurde zuerst in [1] vorgestellt und kann formal so beschrieben werden [1,2]: Sei $I = \{i_1, i_2, \ldots i_m\}$ eine Menge von m Literalen, genannt Items. Sei $D = \{t_1, t_2, \ldots t_n\}$ eine Datenbank von Transaktionen, wobei jede Transaktion eine Teilmenge von I ist. Eine Teilmenge von Items X wird k-Itemset genannt, wenn die Anzahl von Items in X gleich k ist. Der *Support* (Häufigkeit) von Itemset X, bezeichnet $sup(X)$, ist der Anteil von Transaktionen in der Datenbank D, die X enthalten. Ein Itemset wird *häufiges Itemset* genannt, wenn sein Support größer oder gleich dem vom Benutzer spezifizierten Grenzwert ist [42]. Ein häufiges Itemset nennt man *maximal*, wenn es kein Subset eines anderen häufigen Itemsets ist [49].

Eine Assoziationsregel r ist eine Regel der Form $X \Rightarrow Y$, wo beide X und Y nicht leere Teilmengen von I sind und $X \cap Y = \varnothing$. X wird *Regelprämisse* und Y *Regelfolgerung* genannt. Der *Support* und *Konfidenz* einer Regel r: $X \Rightarrow Y$ werden $sup(r)$ und $conf(r)$ bezeichnet und definiert als $sup(r) = sup(X \cup Y)$ und $conf(r) = sup(X \cup Y) / sup(X)$. Support von r: $X \Rightarrow Y$ ist einfacher Maß für statistische Signifikanz und Konfidenz von r ist ein Maß für bedingte Wahrscheinlichkeit, dass eine Transaktion Y enthält, wenn gegeben ist, dass sie X enthält [42].

Die Menge aller Assoziationsregeln mit minimalem Support s und minimaler Konfidenz c wird $AR(s, c)$ oder AR bezeichnet

Das Problem der Entdeckung von Assoziationsregeln kann in 2 Teilprobleme zerlegt werden [1]:

1. Das Finden aller Itemmengen, deren Häufigkeiten über den minimalen Support liegen. Itemmengen mit dem minimalen Support werden *häufige Itemmengen* genannt. Im Folgenden wird der Algorithmus *Apriori* vorgestellt, der dieses Problem löst.

2. Das Generieren von gewünschten Assoziationsregeln unter Einsatz von *häufigen Itemmengen*. Der unkomplizierte Algorithmus funktioniert wie folgt. Für jedes häufige Itemset l finde alle nichtleeren Subsets von l. Für jede solche Teilmenge a gib die Regel $a \Rightarrow (l - a)$ aus, wenn der Quotient von $sup(l)$ und $sup(a)$ größer oder gleich *minconf* ist. Man muss alle Teilmengen von l betrachten, um die Regeln mit zusammengesetzter Regelfolgerung zu generieren.

2.2 Apriori

Mit diesem Algorithmus lassen sich häufige Itemmengen und Abhängigkeiten zwischen einzelnen oder mehreren Items in Form von Assoziationsregeln entdecken.

2.2.1 Entdeckung von häufigen Itemsets

Algorithmen zur Entdeckung von häufigen Itemsets laufen mehrfach durch die Datenbank. Beim ersten Durchlauf zählt man die Vorkommnisse der einzelnen Items und bestimmt, welche von ihnen häufig sind, d.h. den minimalen Support haben. In jedem folgenden Durchlauf k startet man mit der Menge von häufigen Itemsets (der Länge (k-1)), die im letzten Durchlauf gefunden wurden. Man nutzt diese Menge zur Generierung von neuer potentiell häufigen Itemsets der Länge k, genannt Kandidaten-Itemsets, und zählt man die Häufigkeit dieser Kandidaten-Itemsets während des Durchlaufs durch die Datenbank. Nach dem Datenbankdurchlauf stellt man fest, welche von diesen Kandidaten-Itemsets häufig sind, sie werden im folgenden Durchlauf zur Bildung der Itemsets der Länge k+1 genutzt. Der Prozess läuft, bis keine neuen Itemmengen gefunden werden können [2].

Der Apriori–Algorithmus generiert die Kandidaten-Itemsets unter Einsatz von häufigen Itemsets aus dem letzten Durchlauf, ohne einzelne Transaktionen aus der Datenbank zu betrachten. Das Hauptprinzip ist, dass jede Teilmenge einer häufigen Menge auch häufig ist (*Monotonie-*

Eigenschaft). Deshalb können Kandidaten-Itemsets, die *k* Items haben, durch die Vereinigung von häufigen Itemmengen der Länge *k*-1 und die Entfernung derjenigen, die irgendeine nichthäufige Teilmenge enthalten, generiert werden [2].

```
L1 ={häufige 1-Itemsets};
for (k = 2; Lk-1≠ ∅; k++) do begin
   Ck = apriori-gen(Lk-1); // neue Kandidaten
   forall transaction t ∈ D do begin
      Ct = subset (Ck, t) // in t enthaltene Kandidaten
      forall candidates c ∈ Ct do
         c.count++;
   end
   Lk = { c ∈ Ck | c.count ≥ minsup}
end
Return ∪k Lk;
```

Abb. 2.2.1: *Apriori*-Algorithmus [2]

In Abbildung 2.2.1 wird der *Apriori–*Algorithmus vorgestellt. Im ersten Durchlauf werden einfach die Itemvorkommnisse gezählt, um die häufigen 1-Itemsets festzustellen. Jeder folgende Durchlauf *k* besteht aus 2 Phasen. In der ersten Phase werden die häufigen Itemmengen L_{k-1}, die im *k*-1-ten Durchlauf gefunden wurden, zur Generierung von Kandidaten-Itemsets unter Einsatz der *Apriori–Gen–*Funktion verwendet. Die *Apriori–Gen–*Funktion erhält als Parameter L_{k-1}, die Menge häufiger (*k*-1)-Itemsets. Sie liefert die Supermengen aller häufigen *k*-Itemsets. Die Funktion arbeitet wie folgt. Zuerst werden im Vereinigungsschritt L_{k-1} und L_{k-1} zusammengefügt [2]:

Insert into C_k
Select $p.\text{item}_1$, $p.\text{item}_2$, . . . , $p.\text{item}_{k-1}$, $q.\text{item}_{k-1}$
from L_{k-1} *p*, L_{k-1} *q*
where $p.\text{item}_1 = q.\text{item}_1$, $p.\text{item}_2 = q.\text{item}_2$, ..., $p.\text{item}_{k-2} = q.\text{item}_{k-2}$, $p.\text{item}_{k-1} < q.\text{item}_{k-1}$

Im nächsten "Abschneiden"-Schritt werden alle Itemsets $c \in C_k$, so dass (*k*-1)-Teilmengen von *c*, die nicht in L_{k-1} sind, entfernt:

```
forall itemsets c ∈ Ck do
    forall (k-1)-subsets s of c do
        if (s ∉ Lk-1) then
            delete c from Ck;
```

Als Nächstes wird die Datenbank durchgelaufen und der Support von den Kandidaten in C_k berechnet. Zur schnellen Berechnung braucht man nur,

Kandidaten in C_k, die in der Transaktion t vorkommen, effizient zu bestimmen.

Damit die *subset*-Funktion effizient arbeitet, werden Kandidaten-Itemsets in einem Hashbaum gespeichert. Ein Knoten im Hashbaum enthält entweder eine Liste von Itemsets (ein *Blatt*) oder eine Hashtabelle (ein *innerer Knoten*). In einem *inneren Knoten* verweist jeder Behälter von der Hashtabelle auf einen anderen Knoten. Der Wurzel von dem Hashbaum hat die Tiefe 1. Ein *innerer Knoten* der Tiefe d verweist auf die Knoten der Tiefe d+1. Itemsets werden in *Blättern* gespeichert. Wenn man ein Itemset hinzufügt, startet man von der Wurzel und geht weiter in die Tiefe des Baumes, bis man das *Blatt* erreicht. An einem *inneren Knoten* der Tiefe d entscheidet man, welchem Ast man weiter folgt bei der Anwendung der Hashfunktion für den d-ten Item der Itemmenge. Alle Knoten sind anfangs als *Blätter* initialisiert. Wenn die Anzahl der Itemmengen in einem *Blatt* einen gewissen Grenzwert überschreitet, wird das *Blatt* in einen *inneren Knoten* umgewandelt.

Beginnend von einem *Wurzelknoten* findet die *subset*-Funktion alle Kandidaten, die in einer Transaktion enthalten sind, wie folgt. Wenn man an einem *Blatt* steht, findet man, welche Itemmengen in dem *Blatt* in t enthalten sind und fügt man die Referenzen zu ihm zu Ergebnismenge. Wenn man an einem *inneren Knoten* steht und man hat ihn durch Hashing des Items i erreicht, wendet man die Hashfunktion für jedes Item nach i in t an und rekursiv wendet man diese Prozedur zu dem *Knoten* in dem entsprechenden Behälter an. Für den *Wurzelknoten* wendet man die Hashfunktion für jedes Item in t an [2].

2.3 Repräsentative Assoziationsregeln (*RAR*)

Da die Anzahl der entdeckten Assoziationsregeln üblicherweise riesig groß ist, ist es sehr schwer für einen Experten diese Regeln zu analysieren und ihn interessierende zu identifizieren. Das besondere Interesse der Experten liegt in der Entdeckung der für einen Benutzer wichtigen Regeln und in der Reduzierung der Anzahl der gefundenen Assoziationsregeln [4,33,23,43]. Viele dieser Methoden führen zusätzliche Kriterien für Interessantsein einer Regel ein und filtern solche Regeln heraus, die zusätzlichen Kriterien nicht genügen. Eine Menge der repräsentativen Assoziationsregeln (*RAR*) ist die minimale Regelnmenge, aus der alle Assoziationsregeln generiert werden können. Die Anzahl der repräsentativen Assoziationsregeln ist viel kleiner als die Anzahl aller Assoziationsregeln. Außerdem braucht man keine zusätzlichen Kriterien zur Bestimmung der repräsentativen Assoziationsregeln [42], d.h

minimaler Support und minimale Konfidenz sind auch hier die notwendigen Kriterien.

Algorithmen zur Entdeckung von repräsentativen Assoziationsregeln sind in [20,21] vorgestellt. Diese Algorithmen nutzen alle häufigen Itemmengen, um *RAR* zu finden. Im Folgenden wird eine andere Methode zur Generierung der repräsentativen Assoziationsregeln aus [42] präsentiert. Diese Methode nutzt nur eine Teilmenge der Menge häufiger Itemsets, die man *häufige abgeschlossene Itemsets* genannt hat. Dies ergibt eine Reduktion der Eingabegröße und daher auch schnellere Algorithmen zur Entdeckung von Assoziationsregeln [42]. Dabei wurden die Ideen des formalen Analysekonzepts genutzt, um häufige abgeschlossene Itemmengen zu finden.

Um die Menge von repräsentativen Assoziationsregeln zu definieren, muss man zuerst den Begriff *Cover Operator* (Deckungsoperator) einführen.

Informell ist *Cover* von der Regel r: $X \Rightarrow Y$, bezeichnet $C(r)$, die Menge von Assoziationsregeln, die aus r generiert werden können. Formal [42]:

$$C(r: X \Rightarrow Y) = \{ X \cup U \Rightarrow V \mid U, V \subseteq Y, U \cap V = Æ \text{ und } V \neq Æ \}.$$

Eine wichtige Eigenschaft von Cover Operator ist, dass, wenn eine Assoziationsregel r Support s und Konfidenz c hat, dann hat jede Regel $r' \in C(r)$ den Support mindestens s und Konfidenz mindestens c [21].

Unter Einsatz von Cover Operator lässt sich die Menge repräsentativen Assoziationsregeln(*RAR*) mit minimalem Support s und minimaler Konfidenz c so definieren [42]:

$$RAR(s, c) = \{ r \in AR(s, c) \mid \text{es gibt keinen } r' \in AR(s, c), r \neq r' \text{ und } r \in C(r') \}.$$

So ist die Menge der repräsentativen Assoziationsregeln die kleinste Menge von Assoziationsregeln, die alle Assoziationsregeln deckt und aus denen alle Assoziationsregeln generiert werden können. Klar,

$$AR(s, c) = \cup\{C(r) \mid r \in RAR(s, c) \}.$$

Sei die Länge von $X \Rightarrow Y$ die Anzahl von Items in $X \cup Y$. Zur Entdeckung von *RAR* sind einige Eigenschaften von *RAR* von großer Bedeutung [20,21].

Eigenschaft 1. Sei r: $X \Rightarrow Y$ und r': $X' \Rightarrow Y'$ 2 verschiedene Assoziationsregeln, dann gelten:

1. *Wenn r länger ist als r', dann $r \notin C(r')$.*
2. *Wenn r kürzer ist als r', dann $r \in C(r')$, wenn $X \cup Y \subset X' \cup Y'$ und $X \supseteq X'$.*
3. *Wenn r und r' gleicher Länge sind, dann $r \in C(r')$, wenn $X \cup Y = X' \cup Y'$ und $X \supset X'$.*

Eigenschaft 2. Sei r: $X \Rightarrow Z \backslash X \in AR(s, c)$ und sei $maxSup = max(\{sup(Z') | Z \subset Z' \subseteq I\} \cup \{0\})$. Dann gilt $r \in RAR(s, c)$, wenn folgende Bedingungen erfüllt sind:

i. $maxSup < s$ oder $maxSup/sup(X) < c$,
ii. es gibt keinen X', $Æ \subset X' \subset X$, so dass $X' \Rightarrow Z \backslash X' \in AR(s, c)$,

Eigenschaft 3. Sei $Æ \neq X \subset Z \subset Z' \subseteq I$ und $sup(Z) = sup(Z')$. Es gibt dann keine Regel r: $X \Rightarrow Z \backslash X \in AR(s, c)$, so dass $r \in RAR(s, c)$.

Diese Eigenschaften führen zu Entwicklung von Algorithmen *GenAllRepresentatives* und *FastGenAllRepresentatives* zur Entdeckung von *RAR* [20, 21]. Beide Algorithmen nutzen alle häufigen Itemmengen, die sich durch Anwendung von Apriori auf der Datenbank D ergeben. Die im Folgenden vorgestellte Methode nutzt nur eine Teilmenge von häufigen Itemmengen, die *häufige abgeschlossene Itemsets* genannt werden. Häufige abgeschlossene Itemsets werden mit Methoden aus der formalen Begriffsanalyse gefunden.

2.4 Abgeschlossene Itemsets (Closed Itemsets)

In diesem Kapitel werden theoretische Grundlagen präsentiert, die zur Entwicklung der Algorithmen zur RAR–Generierung geführt haben. Diese Ergebnisse beziehen sich direkt auf die formale Begriffsanalyse [48]. Der Begriff des abgeschlossenen Itemsets ist sehr ähnlich dem eines Verbandes. Informell ist ein Verband ein Paar von zwei Mengen: Menge von Objekten (Transaktionen oder Itemsets) und Menge von Eigenschaften (Items) gemeinsamen für alle Objekte. Unter Einsatz der Struktur einer formalen Begriffsanalyse werden Begriffe in Form eines Verbandes aufgebaut, genannt Begriffsverband. Der Begriffsverband hat sich als sehr nützliches Werkzeug zur Wissensrepräsentation und Wissensentdeckung bewährt [15].

Definition 1. *Data Mining Kontext* ist definiert als Tripel (T, I, R), wobei T die Transaktionsmenge, I die Itemmenge und $R \subseteq T \times I$.

Ein *Data Mining Kontext* ist die formale Definition der Datenbank. Die Menge T ist die Transaktionsmenge in der Datenbank und die Menge I

ist die Itemmenge in der Datenbank. Für $t \in T$ und $i \in I$ schreibt man $(t, i) \in R$ und meint, dass Transaktion t das Item i enthält. Ein Beispiel von einem Data Mining Kontext ist in der Tabelle 2.4.1 zu sehen, wo x in der t-ter Zeile und i-ter Spalte bedeutet, dass $(t, i) \in R$.

	A	B	C	D	E	F	G	H
t_1	x	x	x	x	x			
t_2	x	x	x	x	x	x		
t_3	x	x	x	x	x		x	x
t_4	x	x			x			
t_5		x	x	x	x		x	x

Tabelle 2.4.1: Beispiel eines *Data Mining Kontext* [42]

Definition 2. Sei (T, I, R) ein Data Mining Kontext, $X \subseteq T$, und $Y \subseteq I$. Die Abbildungen a, b sind folgendermaßen definiert [42]:

$b : 2^T \rightarrow 2^I$, $b(X) = \{ i \in I \mid (t, i) \in R \ \forall t \in X \}$,

$a : 2^I \rightarrow 2^T$, $a(Y) = \{ t \in T \mid (t, i) \in R \ \forall i \in Y \}$.

Die Abbildung $b(X)$ assoziiert mit X die Menge von Items, die gemeinsam in allen Transaktionen aus X sind. Ähnlich assoziiert die Abbildung $a(Y)$ mit Y die Menge aller Transaktionen, die alle Items aus Y haben. Intuitiv ist $b(X)$ die maximale Menge von Items, die in allen Transaktionen in X sind, und $a(Y)$ die maximale Menge von Transaktionen, die alle Attribute aus Y besitzen. Am folgenden Beispiel werden diese Begriffe erklärt.

Beispiel 1 [42]. *Gegeben sei die in der Tabelle 2.4.1 dargestellte Datenbank und sei* $X = \{ t_1, t_2 \}$ *und* $Y = \{ A,B,C \}$, *dann sind* $b(X) = \{ A,B,C,D,E \}$, $a(Y) = \{ t_1, t_2, t_3 \}$, $a(b(X)) = a(\{ A,B,C,D,E \}) = \{ t_1, t_2, t_3 \}$, *und* $b(a(Y)) = b(\{ t_1, t_2, t_3 \}) = \{ A,B,C,D,E \}$.

Generell gilt: $b(a(Y)) \neq Y$, wobei Y ein Itemset ist.

Definition 3. Ein Itemset Y, das die Bedingung $b(a(Y)) = Y$ erfüllt, wird *abgeschlossenes Itemset* genannt.

Abgeschlossene Itemsets sind von besonderer Bedeutung, weil alle Glieder eines Begriffsverbandes eines Data Mining Kontext die Bedingung $b(a(Y)) = Y$ erfüllen. Dieser Schritt der Berücksichtigung nur abgeschlossener Itemsets kann als erste Maßnahme in der Verringerung des Itemsetverbandes betrachtet werden.

Beispiel 2 [42]. *Sei* $Y = \{ A,B,C,D,E \}$, *dann gilt* $b(a(Y)) = Y$. *Daher ist* Y *ein abgeschlossenes Itemset. Im Gegensatz ist das Itemset* $\{ A,B,C \}$ *nicht abgeschlossen, weil* $b(a(\{ A,B,C \})) = \{ A,B,C,D,E \} \neq \{ A,B,C \}$.

Der Begriffsverband kann noch weiter verkleinert werden, indem man nur abgeschlossene Itemsets mit größerem Support als minimaler Support betrachtet. Dies führt zur folgenden Definition.

Definition 4. Ein *häufiges abgeschlossenes Itemset* ist ein abgeschlossenes Itemset, das außerdem häufig ist. Es hat Support größer oder gleich dem vom Benutzer spezifizierten Wert für den minimalen Support.

DISTINCT DATABASE ITEMS

Jane Austin	*Agatha Christie*	*Sir Arthur Conan Doyle*	*Mark Twain*	*P.G. Wodehouse*
A	*C*	*D*	*T*	*W*

DATABASE

Transaction	*Items*
1	*A C T W*
2	*C D W*
3	*A C T W*
4	*A C D W*
5	*A C D T W*
6	*C D T*

ALL FREQUENT ITEMSETS

MINIMUM SUPPORT = 50%

Support	*Itemsets*
100% (6)	*C*
83% (5)	*W, CW*
67% (4)	*A, D, T, AC, AW CD, CT, ACW*
50% (3)	*AT, DW, TW, ACT, ATW CDW, CTW, ACTW*

Tabelle 2.4.2: Beispiel einer Datenbank mit häufigen Itemmengen [49]

Beispiel 3 [49]. *Gegeben sei die in Tabelle 2.4.2 dargestellte Datenbank mit I = { A,C,D,T,W } und T = { 1,2,3,4,5,6 }. Die Tabelle rechts zeigt alle 19 häufigen Itemsets, die mindestens in 3 Transaktionen enthalten sind (minsup=50%). Abbildung 2.4.1 zeigt die 19 häufigen Itemsets, die als Subsetverband organisiert sind. Zu jeder häufigen Itemmenge werden Transaktionen, wo sie vorkommen, angegeben. Alle häufigen Itemsets $X_1, X_2, \ldots X_n$ mit $a(X_1)=a(X_2)=\ldots= a(X_n)$ werden zusammengefügt (in der Abbildung durch 7 abgeschlossene Bereiche dargestellt). So gehören Itemsets A,AC,AW,ACW zu t = { 1,3,4,5 } (in der Abbildung (1345) geschrieben), sie bilden eine separate Gruppe. Die 7 häufigen abgeschlossenen Itemsets { C,CD,CT,CW,ACW,CDW,ACTW } erhält man durch Vereinigung aller Itemmengen, die jeweils zu einer Gruppe gehören (d.h. die gleiche a(X) haben). Betrachtet man den Verband der abgeschlossenen Itemsets, so findet man darunter auch 2 maximale häufige Itemsets ACTW und CDW (durch Kreise markiert).*

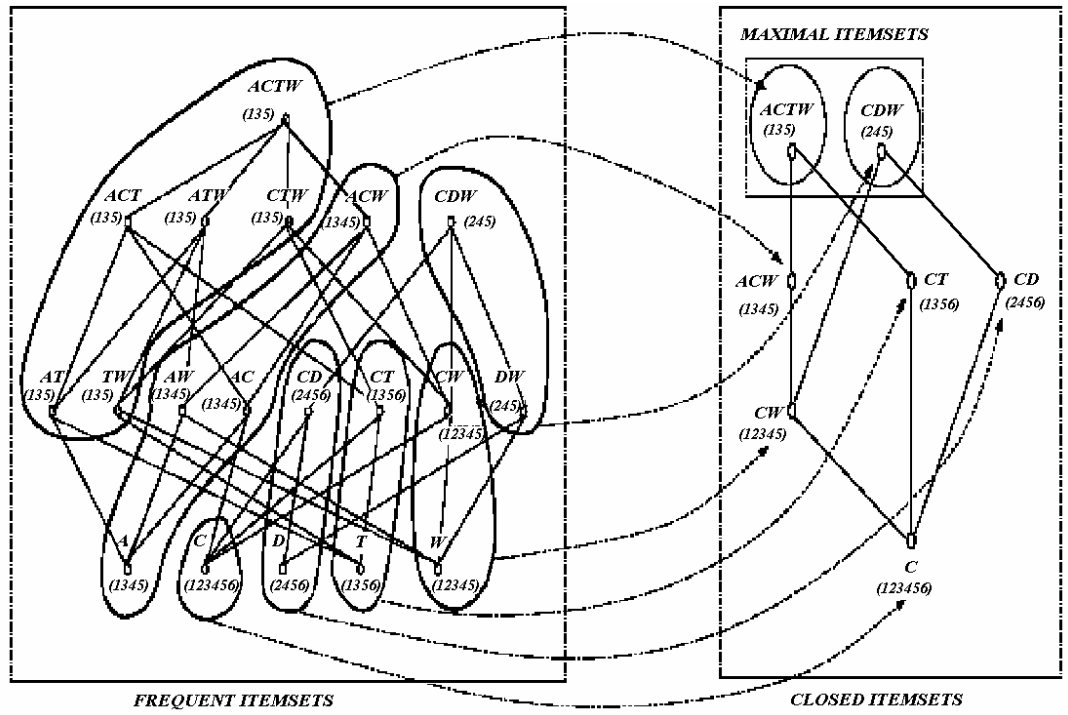


Abb. 2.4.1: häufige abgeschlossene und maximale Itemmengen [49]

Wie das Beispiel 3 zeigt, wenn $\boldsymbol{F}$ die Menge aller häufigen Itemsets, $\boldsymbol{C}$ die Menge der abgeschlossenen Itemsets und $\boldsymbol{M}$ die Menge der maximalen häufigen Itemsets ist, gilt im Allgemeinen $\boldsymbol{M} \subseteq \boldsymbol{C} \subseteq \boldsymbol{F}$. Jedoch sind die abgeschlossenen Mengen verlustfrei in dem Sinne, dass die genaue Häufigkeit aller häufigen Itemsets aus $\boldsymbol{C}$ bestimmt werden kann, währenddessen die Menge $\boldsymbol{M}$ zu einem Informationsverlust führt. Um festzustellen, ob ein Itemset *X* häufig ist, findet man das kleinste abgeschlossene Itemset, das auch Superset von *X* ist. Wenn kein Superset existiert, ist *X* nicht häufig. Z.B. um zu prüfen, ob *ATW* (aus Beispiel 3)

häufig ist, findet man, dass *ACTW* das kleinste abgeschlossene Itemset ist, das *ATW* enthält; folglich ist *ATW* häufig und hat dieselbe Häufigkeit wie *ACTW*. Andererseits ist *DT* nicht häufig, weil es kein häufiges abgeschlossenes Itemset gibt, das es enthält.

In [37] ist gezeigt worden, dass $support(X) = support(closure(X))$. Dieses Lemma besagt, dass alle häufigen Itemsets aus den häufigen abgeschlossenen Itemsets eindeutig bestimmt werden können. Es wird im Folgenden gezeigt, dass es nicht nötig ist, die Assoziationsregeln aus allen häufigen Itemsets abzuleiten, weil die meisten von ihnen redundant sind. Tatsächlich genügt es, lediglich die Regeln aus den häufigen abgeschlossenen Itemsets zu betrachten, was folgendes Theorem besagt [49]:

Theorem: Die Regel $X \Rightarrow Y$ mit Konfidenz p ist äquivalent der Regel $closure(X) \Rightarrow closure(Y)$ mit Konfidenz q, wobei $q = p$.

Gegeben seien 2 Itemsetmengen: S_1 mit $|S_1| = n$ und $closure(S_1) = C_1$ und S_2 mit $|S_2| = m$ und $closure(S_2) = C_2$. Dann sind $n*m-1$ Regeln zwischen 2 nicht abgeschlossenen aus S_1 und S_2 abgeleiteten Mengen redundant. Sie sind alle äquivalent der Regel $C_1 \Rightarrow C_2$ mit Konfidenz p. Ferner sind $m*n-1$ Regeln zwischen 2 nicht abgeschlossenen aus S_2 und S_1 abgeleiteten Mengen auch redundant und äquivalent der Regel $C_2 \Rightarrow C_1$ mit Konfidenz q [49]. Im Beispiel aus Tabelle 2.4.1 wurde gefunden, dass Items *B* und *E* auf das abgeschlossene Itemset *BE* und Itemsets *A* und *BE* auf das *ABE* abgebildet wurden. Die entdeckten Regeln $B \Rightarrow A$, $B \Rightarrow AE$, $E \Rightarrow A$, $E \Rightarrow AB$, $BE \Rightarrow A$ mit Konfidenzen 0.8 sind alle äquivalent der Regel $BE \Rightarrow ABE$ mit Konfidenz 0.8.

2.5 Algorithmen

2.5.1 Generierung von häufigen abgeschlossenen Itemsets

Sei $D = (T, I, R)$ ein Data Mining Kontext. Der Algorithmus, der im Folgenden vorgestellt wird, ist eine geringfügige Modifikation des *Close*-Algorithmus, der in [37] erwähnt wird, und wird *Close-FCI* genannt [42]. Beide Algorithmen sind ähnlich zu Apriori.

Es wird angenommen, dass Items in I lexikographisch sortiert sind. Die genutzten Datenstrukturen bestehen aus 2 Mengen: die Menge der Kandidaten der häufigen abgeschlossenen Itemmengen FCC und die Menge der häufigen abgeschlossenen Itemmengen FC. Die Notationen

FCC_i und FC_i werden genutzt, um entsprechend Kandidaten der häufigen abgeschlossenen Itemmengen und die häufigen abgeschlossenen Itemmengen der Größe i zu bezeichnen. Jedes Element in FCC_i und FC_i hat 3 Komponenten: eine Itemsetkomponente, eine *Abschluss*–Komponente[3] und eine *Support*-Komponente [42].

Der *Abschluss* eines Itemsets $X \subseteq I$, bezeichnet durch *closure*(X), ist das kleinste abgeschlossene Itemset, das X enthält, und ist gleich der Schnittmenge aller Itemsets, die X enthalten. In [37] ist gezeigt worden, dass *support*(X) = *support*(*closure*(X)).

```
FCC1.itemsets = {1-itemsets};
for ( k=1; FCCk ¹ Æ ; k++ ) do begin
    forall X ∈ FCCk do begin
        X.closure =Æ;
        X.support =0;
    end forall
    FCCk = Generate-Closures(FCCk);
    forall candidate closed itemsets X ∈ FCCk do begin
        if ( (X.support>=minSupport) and (X.closure ∉ FCk) ) then
            FCk ← FCk ∪ { X };
        end if
    FCCk+1 = Generate-Candidates(FCk);
    endfor
return ∪ k-1 i=1{ FCi.closure and FCi.support };
```

Abb. 2.5.1: Algorithmus *Close-FCI*(*D*) [42]

Als Erstes initialisiert der Algorithmus aus Abbildung 2.5.1 die Itemsets in FCC_1 zu den Items aus der Datenbank, dazu ist kein Datenbankdurchlauf erforderlich. Danach werden in der Iteration k der Hauptschleife des Algorithmus *closure* und *support* jedes Itemsets in FCC_k initialisiert. Dann findet der Algorithmus Kandidaten der häufigen abgeschlossenen Itemsets der Größe k, FCC_k (in der Zeile 7). Häufige abgeschlossene Itemsets FC_k werden im nächsten Schritt gefunden, wobei der Grenzwert für den minimalen Support *minSupport* zum Ausfiltern der nicht häufigen Itemsets genutzt wird. Anschließend generiert der Algorithmus Kandidaten FCC der Größe k +1, FCC_{k+1} (Zeile 12). Als

[3] engl. Closure-Component

Ausgabe liefert der Algorithmus alle häufige abgeschlossene Itemmengen FC_i der Größe i mit i=1, ... k-1 [42].

2.5.2 Generierung des Abschlusses

Der *Abschluss* eines Itemsets $X \subseteq I$ ist nach Definition das kleinste abgeschlossene Itemset, das X enthält, und ist gleich der Schnittmenge aller häufigen Itemsets, die X enthalten. Daher ist *closure*(X) nach folgender Formel zu finden:

$$closure(X) = \cap_{t \in T}\{b(t) \mid X \in b(t)\},$$

was im folgenden Algorithmus (s. Abb. 2.5.2) genutzt wird. Analog kann der *Abschluss* einer Episode (s. Definition 7) definiert werden. Sie ist die kleinste abgeschlossene Episode, die die Episode a enthält und ist gleich der Schnittmenge aller Episoden, die a enthalten. Dieser Algorithmus erfordert einen Datenbankdurchlauf, um den *Abschluss* von allen Elementen aus FCC_k zu finden.

forall transactions $t \in T$ **do begin**
  $b_t = \{ X \in FCC_k \mid X \subseteq b(t) \}$; // alle Itemsets, die in $b(t)$ enthalten sind
  forall itemsets $X \in b(t)$ **do begin**
    if(*X.closure* =Æ) **then**
      X.closure ←$b(t)$;
    else
      X.closure ← *X.closure* ∩ $b(t)$;
    end if
    X.support++ ;
  end forall
end forall
return ∪ { $X \in FCC_k$ | *X.closure* ≠Æ };

Abb. 2.5.2: Algorithmus *Generate-Closures*(FCC_k) [42]

2.5.3 Generierung von Kandidaten-Itemsets

Der Algorithmus *Generate-Candidates*(FC_k) (s. Abb. 2.5.3), der Kandidaten der häufigen abgeschlossenen Itemsets der Größe k +1 findet, nutzt das Ergebnis der folgenden Eigenschaft [37].

Eigenschaft 4. Sei X ein Itemset der Länge k und $X = \{ X_1, X_2, \ldots X_m \}$ *sei die Menge von (k-1)-Teilmengen von X, wo* $\cup_{Xi \in X} = X$. *Wenn* $\exists X_i \in X$, *so dass* $X \subseteq closure(X_i)$, *dann* $closure(X) = closure(X_i)$.

Die Eigenschaft besagt, dass das Itemset X sich nach redundanten Berechnungen von häufigen abgeschlossenen Itemsets ergibt, weil $closure(X)$, die gleich der $closure(X_i)$ ist, bereits generiert ist.

Der *Close-FCI*-Algorithmus erfordert nur einen Datenbankdurchlauf in jeder Iteration. Dieser Durchlauf ist für den *Generate-Closures*-Algorithmus notwendig.

insert into $FCC_{(k+1)}$.itemset
  select X.itemset$_1$, X.itemset$_2$, … X.itemset$_k$, Y.itemset$_k$
  from FC_k.itemset X, FC_k.itemset Y
  where X.itemset$_1$ = Y.itemset$_1$ ∧ X.itemset$_2$ = Y.itemset$_2$ ∧ … ∧ X.itemset$_{k-1}$ = Y.itemset$_{k-1}$ ∧ X.itemset$_k$ < Y.itemset$_k$;
  // schneide alle Supersets von nicht häufigen Itemsets ab
  delete all itemsets $X \in FCC_{(k+1)}$.itemset
  where some k-subset of X is not in FC_k;
  // schneide alle Itemsets mit bereits generiertem Abschluss ab
  delete all itemsets $X \in FCC_{(k+1)}$.itemset
  where the closure of some k-subset of X contains X;
return $\cup \{ X \in FCC_{(k+1)} \}$

Abb. 2.5.3: Algorithmus *Generate-Candidates*(FC_k) [42]

Beispiel 4 [42]. *Wendet man den Close-FCI-Algorithmus an der Datenbank aus der Tabelle 2.4.1 mit dem minimalen Support s = 3/5 = 0.6 und minimaler Konfidenz c = 0,75 an, werden folgende häufige abgeschlossene Itemsets gefunden:*

$$FC = \{ ABCDE, BCDE, ABE, BE \}$$

mit entsprechenden Supports 3/5, 4/5, 4/5 und 5/5. Mit Apriori werden mit den gleichen Werten für s und c 31(!) verschiedene häufige Itemsets (nicht leere Teilmengen von ABCDE) erzeugt. Diese Mengen lassen sich auch ganz anders als häufige Itemsets von Apriori interpretieren. Man kann daraus schließen, dass B und E immer zusammen auftreten, dass A immer mit B und E zusammen auftritt usw.

Daraus werden entsprechend auch viel weniger Assoziationsregeln abgeleitet als dies bei Apriori der Fall ist.

Andere Möglichkeit, Kandidaten für häufige Itemsets zu generieren, wird in Kapitel 2.6.3.3 vorgestellt. Dabei werden Itemsets als geordnete Mengen betrachtet.

2.5.4 Generierung von Repräsentativen Assoziationsregeln (*RAR*)

Im Folgenden wird der Algorithmus zur Generierung von repräsentativen Assoziationsregeln präsentiert (s. Abb. 2.5.4), genannt *Generate-RAR* [42]. Als Eingabe erhält er alle häufige abgeschlossenen Itemmengen (*FCI*) und liefert die Menge von allen *RAR*. *Generate-RAR* ist die Modifikation von *FastGenAllRepresentatives* [20] und er nutzt Eigenschaften 1 bis 3 aus dem Abschnitt 2.3. Außerdem nutzt er die Eigenschaft des Itemsets X, $support(X) = support(closure(X))$.

```
1.  c ← minConfidence; // spezifischer Wert für minimale Konfidenz
2.  RAR ← Æ; // initialiieree set von RAR
3.  k ← 0;
    // trenne häufige abgeschlossene Itemsets entsprechend ihrer Größe
4.  forall X ∈ FC do begin
5.      FC|X| ← FC|X| ∪ {X};
6.      if ( k < |X| ) then k ← |X|;
7.  end forall
8.  for (i ← k ; i >1; i --) do
9.      forall Z ∈ FCi do begin
10.         maxSup = max({sup(Z') | Z ⊂ Z' ∈ FC} ∪ {0});
11.         if (Z.support ≠ maxSup) then begin // Eigenschaft 3
12.             A1 = {{ Z[1]}, { Z[2]},… ,{ Z[i]}} // erzeuge Regelprämissen
13.             for (j = 1; (Aj ≠ Æ) and (j < i) ; j++)
14.                 forall X ∈ Aj do begin
15.                     Y ← smallest closed itemset containing X
16.                     X.support = Y.support;
17.                     // prüfe, ob X ⇒ Z\X  eine RAR ist
18.                     if (Z.support / X.support ≥ c and
                        maxSup / X.support < c ) then
19.                         RAR ← RAR ∪ {X ⇒ Z\X };
20.                     Aj = Aj \{X};
21.                 end if
22.             end forall
23.             Aj+1 ← AprioriGen(Aj);
24.         end for
25.     end if
26.   end forall
27. return RAR;
```

Abb. 2.5.4: Algorithmus *Generate-RAR*(all frequent closed Itemsets FC) [42]

Zuerst wird *RAR*-Menge initialisiert. Dann wird jedes häufige abgeschlossene Itemset X der Länge i zu FC_i hinzugefügt und die Länge des größten abgeschlossenen Itemsets k wird gefunden. Die Zeile *8* "kontrolliert" die Generierung von repräsentativen Assoziationsregeln. Zuerst werden die längsten Regeln (der Länge k) generiert und zu *RAR* hinzugefügt. Danach werden die Regeln der Länge (k-1) generiert und zu *RAR* hinzugefügt usw. Zum Schluss werden die Regeln der Länge *2* erzeugt und an die *RAR* angeschlossen. Das Generieren von Assoziationsregeln der Länge i wird von Zeile *9* bis *26* durch den Einsatz von oben beschriebenen Eigenschaften kontrolliert. In der Zeile *11* wird die Eigenschaft *3* überprüft, laut der keine repräsentative Assoziationsregel aus Z generiert werden kann, wenn $Z.support$ gleich *maxSup* ist. Anderenfalls kann der Prozess der Generierung starten. Als Erstes wird die Menge A_1 als Menge aller 1-elementigen Teilmengen von Z initialisiert. Die Schleife der Zeile *13* bis *24* steuert das Erzeugen von repräsentativen Assoziationsregeln mit den Regelprämissen der Länge j. Alle möglichen Regelprämissen $X \in A_j$ werden betrachtet. Support von X (ist gleich dem $support(closure(X))$) ist bekannt. $X \Rightarrow Z \backslash X$ ist eine gültige repräsentative Assoziationsregel, wenn Konfidenz größer oder gleich c ist und die zweite Bedingung der *Eigenschaft 2* auch erfüllt ist. In diesem Fall wird X aus A_j entfernt. Nachdem alle repräsentativen Assoziationsregeln mit Regelprämissen der Länge j aus Z generiert wurden, A_j kann nicht leer sein. Die *Apriori–Gen*–Funktion wird mit dem Argument A_j aufgerufen, um die Regelprämissen der Länge j+1 zu finden. Die *Eigenschaft 2.ii* ist erfüllt, indem man sichergestellt wird, dass kein Itemset in A_{j+1} die Obermenge der Regelprämissen der bereits generierten repräsentativen Assoziationsregel ist (dafür wird in der Zeile *20* gesorgt).

Beispiel 5 [*42*]. *Lässt man den Generate-RAR-Algorithmus mit häufigen abgeschlossenen Itemsets aus dem Beispiel 4 als Eingabe laufen, werden folgende repräsentative Assoziationsregeln generiert:*

RAR = { $A \Rightarrow BCDE$, $C \Rightarrow ABDE$, $D \Rightarrow ABCE$, $B \Rightarrow CDE$, $E \Rightarrow BCD$, $B \Rightarrow AE$, $E \Rightarrow AB$ }.

Es sind dieselben Regeln, die vom *FastGenAllRepresentatives*-Algorithmus in [20] generiert wurden.

2.6. WINEPI

Mit Hilfe des *WINEPI*-Algorithmus lassen sich die Ereignisfolgen analysieren und periodisch auftretende Ereigniskombinationen (*häufige Episoden*) entdecken. Zuerst wird das Konzept der Ereignisfolgen formuliert, danach wird detaillierter auf die Episoden eingegangen.

2.6.1 Ereignisfolgen

Man betrachtet die Eingabe als eine Folge der Ereignisse, wobei jedem Ereignis die zugehörige Zeit des Auftretens entspricht. Gegeben sei eine Menge E von *Ereignistypen*, ein Ereignis ist ein Paar (A, t), wobei $A \in E$ ein Ereignistyp und t eine ganze Zahl, die Zeit des Auftretens. Der Ereignistyp kann einige Attribute beinhalten; aus Einfachheitsgründen wird im Weiteren unter dem Ereignistyp eine Zahl gemeint [28].

Definition 5. Eine *Ereignisfolge* s über E ist ein Tripel (s, T_s, T_e), wobei $s = \langle (A_1, t_1),(A_2, t_2),\ldots,(A_n, t_n) \rangle$ eine geordnete Folge von Ereignissen ist, so dass $A_i \in E$ für alle $i = 1,\ldots,n$ und $t_i \leq t_{i+1}$ für alle $i = 1, \ldots, n-1$. Weiter sind $T_s < T_e$ ganze Zahlen, T_s nennt man die Startzeit und T_e die Endzeit, und $T_s \leq t_i < T_e$ für alle $i=1,\ldots,n$.

Das Beispiel aus Abbildung 2.6.1 zeigt eine Ereignisfolge $s = (s, 49, 62)$, wo $s = \langle$ (B, 50),(C,51),(A,53),(D,53),(E, 54),(D,55),(C,57),...,(B,60),(F,60) $\rangle$, wobei einzelne Ereignisse teilweise zeitgleich auftreten.

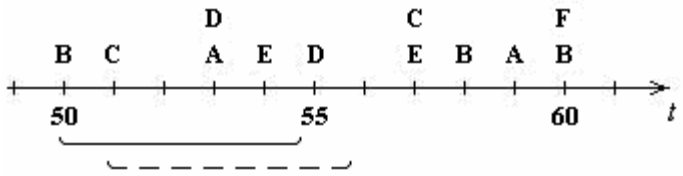


Abb. 2.6.1: Beispiel einer Ereignisfolge

In der Sequenzanalyse interessiert man sich für das Auffinden aller häufigen Episoden aus der Episodenmenge. *WINEPI*–Algorithmus betrachtet diejenigen Ereignisse einer Episode als interessant, die innerhalb einer bestimmten Zeitspanne auftreten. Der User definiert durch Angabe der Breite des Zeitfensters[4], innerhalb welcher Zeitspanne die Ereignisse auftreten müssen, um als interessant eingestuft zu werden. Für *WINEPI* wird Fenster als ein Teil der Ereignisfolge definiert und man betrachtet eine Ereignisfolge als Folge teilweise überlappender

[4] engl. time window

Fenster. Außer der Fensterbreite gibt User an, in wie vielen Fenster eine Episode auftreten muss, um als häufig betrachtet zu werden [28].

Definition 6. Formal ist ein *window* über Ereignisfolge $\boldsymbol{s} = (s, T_s, T_e)$ eine Ereignisfolge w = (w, t_s, t_e), wobei $t_s < T_e$, $t_e > T_s$ und w enthält solche Paare (A, t), wo $t_s \leq t < t_e$. Die Zeitspanne $t_e - t_s$ nennt man Breite des Fensters w und wird bezeichnet als *width*(w). Gegeben sei eine Ereignisfolge $\boldsymbol{s}$ und eine Zahl *win*. Man bezeichnet mit $W(\boldsymbol{s}, win)$ die Menge aller Fenster **w** über $\boldsymbol{s}$ mit *width*(**w**) = *win*. Das erste und das letzte Fenster, die sich außerhalb der Sequenz ausdehnen, werden so definiert, dass das erste Fenster nur den ersten und das letzte Fenster nur den letzten Zeitpunkt decken. Mit dieser Definition betrachtet man Ereignisse an beiden Enden der Folge in genau so vielen Fenstern wie die Ereignisse in der Sequenzmitte.

In der Abbildung 2.6.1 sind 2 Fenster der Breite 5 auf der Folge $\boldsymbol{s}$ gezeigt. Das zum Zeitpunkt 50 gestartete Fenster ist

(⟨(B,50),(C,51),(A,53),(D,53),(E,54)⟩,50,55).

2.6.2 Episoden

Informell ist eine Episode eine partiell geordnete Sammlung von zusammen auftretenden Ereignissen. Eine Episode kann als gerichteter azyklischer Graph beschrieben werden. Man unterscheidet *serielle* und *parallele* Episoden.

Formal werden Episoden folgendermaßen definiert [28]:

Definition 7. Eine Episode α ist ein Tripel $(V, \leq, g)$, wobei V eine Knotenmenge, $\leq$ partielle Ordnung über V sind und g: $V \rightarrow E$ als Abbildung, die jeden Knoten mit einem Ereignistyp assoziiert. Die Bedeutung der Episode ist, dass Ereignisse in $g(V)$ in der mit $\leq$ beschriebener Reihenfolge auftreten. Die Größe von α, bezeichnet $|\alpha|$, ist $|V|$. Episode α ist *parallel*, wenn partielle Ordnung $\leq$ trivial ist (d.h. $x \not\leq y$ für alle $x, y \in V$ und $x \neq y$). Episode β ist *seriell*, wenn Beziehung $\leq$ eine totale Ordnung darstellt (d.h. $x \leq y$ oder $y \leq x$ für alle $x, y \in V$). Episode α ist *injektiv*, wenn g eine injektive Abbildung ist, d.h. kein Ereignistyp tritt zweifach in der Episode auf.

Als nächstes wird definiert, wann eine Episode eine Subepisode von der anderen ist; diese Beziehung wird benutzt bei Entdeckung aller häufigen Episoden [28].

Definition 8. Eine Episode $b = (V', \leq', g')$ ist eine Subepisode der Episode $a = (V, \leq, g)$, bezeichnet $b \leq a$, wenn es eine injektive Abbildung $f : V \rightarrow V'$ existiert, so dass $g'(v) = g(f(v))$ für alle $v \in V'$ und für alle $v, w \in V'$ mit $v \leq' w$ also $f(v) \leq f(w)$ gilt. Eine Episode a ist eine Superepisode der Episode b nur dann und nur dann, wenn $b \leq a$.

Definition 9. Man definiert *frequency* von der Episode als Anteil der Fenster, in denen diese Episode aufgetreten ist. Bei gegebenen Ereignisfolge **s** und Fensterbreite *win* wird *frequency* der Episode a folgendermaßen definiert:

$$fr(a, \mathbf{s}, win) = \frac{|\{w \in W(\mathbf{s}, win) \mid a \text{ tritt in w auf}\}|}{|W(\mathbf{s}, win)|}$$

Gegeben sei Grenzwert für *frequency* min_fr, a ist häufig, wenn $fr(a, \mathbf{s}, win) \geq min\_fr$. Die Aufgabe ist es, alle häufige Episoden aus der gegebenen Klasse E der Episoden zu finden. Die Klasse kann z.B. alle parallele oder alle serielle Episoden sein. Man bezeichnet die Menge der häufigen Episoden mit Bezug auf **s**, *win*, min_fr mit $F(\mathbf{s}, win, min\_fr)$ [28].

Sind die häufigen Episoden bekannt, dann können sie zur Ermittlung der Regeln, die Zusammenhänge zwischen Ereignissen in der gegebenen Sequenz beschreiben, verwendet werden.

2.6.3 Algorithmen

2.6.3.1 Hauptalgorithmus

Der Algorithmus aus Abbildung 2.6.2 zeigt, wie Regeln und ihre Konfidenzen aus den Häufigkeiten der Episoden ermittelt werden können. Nachdem die häufigen Episoden bekannt sind, vergleicht man Quotienten der *frequency* der Episoden und ihrer Subepisoden mit der *minimalen frequency* min_fr: ist der Quotient größer als min_fr, wird die Regel ausgegeben.

***Input**: Eine Menge E von Ereignistypen, eine Ereignisfolge **s** über E, eine Menge E von Episoden, eine Fensterbreite win, eine minimale Häufigkeit min_fr, und minimale Konfidenz min_conf.*
***Output**: Die Episodenregeln, die in **s** mit win, min_fr und min_conf gelten.*
Method:
/**Finde häufige Episoden* (s. Abschnitt 2.6.3.2);*/
compute ***F***(**s**, *win*, *min_fr*);
/**Generiere Regeln**/
for all $a \in \boldsymbol{F}$(**s**, *win*, *min_fr*) **do**
for all $b < a$ **do**
if $fr(a)/fr(b) \geq min\_conf$ **then**
output the rule $b \rightarrow a$ and the confidence $fr(a)/fr(b)$;

Abb. 2.6.2: Hauptalgorithmus *WINEPI* [28]

2.6.3.2 Generierung der häufigen Episoden

***Input**: Eine Menge E von Ereignistypen, eine Ereignisfolge **s** über E, eine Menge E von Episoden, eine Fensterbreite win, eine minimale Häufigkeit min_fr.*
***Output**: Die Menge **F**(**s**, win, min_fr).von häufigen Episoden..*
Method:
berechne $C_1 := \{ a \in E \mid |a|=1 \}$;
l:=1;
while $C_l \neq \emptyset$ **do**
/***Datenbankdurchlauf**/
berechne **F** := $\{ a \in C_l \mid fr(a, \mathbf{s}, win) \geq min\_fr \}$;
$l := l+1$;
Kandidatengenerierung (s. Abschnitt 2.6.3.3)***/***
berechne $C_l := \{ a \in E \mid |a|=l$ und für alle $b \in E$, so dass $b \leq a$ und $|b| < l$ haben wir $b \in \mathbf{F}_{|b|} \}$;
forall l **do** output $\mathbf{F}_l$

Abb. 2.6.3: Algorithmus 2 Generierung von häufigen Episoden [28]

Der in Abbildung 2.6.3 vorgestellte Algorithmus 2 läuft mehrfach die Eingabesequenz durch. Im ersten Schritt werden alle Ereignisvorkommnisse (Episoden der Länge 1) gezählt. Mit anderen Worten, für jedes Ereignis wird die Anzahl der *window*, die diesen Event enthalten, ermittelt. Der Durchlauf k startet mit der Generierung der k-langen Episodenkandidaten C_k aus den häufigen Episoden der Länge k-1 aus dem vorigen Durchlauf. Diese Methode basiert auf der Subset-Eigenschaft von Apriori, die besagt, dass eine Episode nur dann häufig

sein kann, wenn alle ihre Subepisoden häufig sind. Der Algorithmus terminiert, wenn nach einem Durchlauf keine häufigen Episoden generiert werden konnten.

2.6.3.3 Generierung von Kandidaten-Episoden

Der Algorithmus aus Abbildung 2.6.4 berechnet Kandidaten für parallele Episoden. Eine Episode $a = (V, \leq, g)$ wird als ein lexikographisch sortiertes Array von Ereignissen repräsentiert. Das Array wird durch den Episodennamen bezeichnet und jedes Ereignis kann als ein Arrayelement abgefragt werden. Die Menge der Episoden ist durch ein lexikographisch sortiertes Array repräsentiert, d.h. die i-te Episode der Menge F wird durch $F[i]$ bezeichnet. Da Episoden und Episodenmengen sortiert sind,

***Input**: Ein sortiertes Array F_l von häufigen parallelen Episoden der Länge l.*
***Output**: Ein sortiertes Array von Kandidaten für parallele Episoden der Länge l+1.*
Method:

```
1.  C_{l+1}:=∅;
2.  k:=0;
3.  if l=1 then for h:=1 to |F_l| do F_l.block_start[h] :=1;
4.  for i:=1 to |F_l| do
5.          current_block_start:=k+1;
6.          for (j = i; F_l.block_start[j]= F_l.block_start[i]; j:= j+1) do
7.                  /* F_l [i]  und F_l [j] haben l-1 erste Ereignistypen gemeinsam,
8.                  bilde einen potentiellen Kandidaten a als ihre Kombination*/
9.                   for x:=1 to l do a[x]:= F_l[i][x];
10.                 a[l+1]:= F_l[j][l];
11.                 /*Bilde und teste Subepisoden b, die in a[y] nicht enthalten
    sind*/
12.                 for y:=1 to l-1 do
13.                     for x:=1 to y-1 do b[x]:=a[x];
14.                     for x:=y to l do b[x]:=a[x+1];
15.                     if b is not in F_l then continue with the next j at the line 6:
16.                 /*Alle Episoden sind in F_l , speichere a als Kandidaten*/
17.                 k := k +1;
18.                 C_{l+1}[k]:= a;
19.                 C_{l+1}.block_start[k]:= current_block_start;
20. output C_{l+1};
```

Abb. 2.6.4: Der Algorithmus *Kandidatengenerierung* [28]

stehen alle Episoden, die die gleichen ersten Ereignisse gemeinsam haben, unmittelbar hintereinander in dem Episodenarray. Das bedeutet, dass, wenn Episoden $F_l[i]$ und $F_l[j]$ der Länge l die ersten l-1 Ereignisse gemeinsam haben, dann hat $F_l[k]$ für alle k mit $i \leq k \leq j$ die gleichen ersten l-1 Ereignisse. Die maximale Sequenz der hintereinander stehenden Episoden der Länge l, die die gleichen ersten l-1 Ereignisse haben, wird als *block* bezeichnet. Potentielle Kandidaten werden durch Kombinationen aller Episodenpaaren im selben *block* gebildet. Zum effizienten Zugriff auf diese *blocks*, wird in $F_l.block\_start[j]$ für jede Episode $F_l[j]$ i, wo $F_l[i]$ die erste Episode im *block* ist, gespeichert.

Der Algorithmus kann leicht zu Generierung der seriellen Episoden modifiziert werden. Die Zeile 6 wird folgendermaßen ersetzt:

6. **for** (j:= $F_l.block\_start[i]$; $F_l.block\_start[i]$= $F_l.block\_start[j]$; j:=j+1) **do**

Wenn es gefordert wird, dass Episoden *injektiv* sein müssen, muss noch eine weitere Zeile nach Zeile 6 hinzugefügt werden:

6b. **if** j=i **then** continue with the next j at line 6;

Nach dem Generieren von Kandidaten-Episoden muss man überprüfen, ob diese die Bedingung der minimalen Häufigkeit erfüllen. Dazu muss man die Datenbank durchlaufen und die Episodenvorkommnisse wieder zählen. Die Erkennung der Episoden in den Sequenzen geschieht auf der inkrementellen Art. Für zwei Fenster w = (w, t_s, t_s + *win*) und w' = (w', t_s+1, t_s+ *win* +1) sind Ereignisfolgen $w$und w' ähnlich zu einander. Man macht dieses Prinzip der Ähnlichkeit zu Nutze: nach dem Erkennen von Episoden in w aktualisiert man inkrementell die Datenstrukturen, um die Verschiebung des Fensters zu w' zu erreichen.

Wenn bei der Analyse der Klickfolgen alle Aktivitäten jeweils einem bestimmten Benutzer zugeordnet werden können und Klicks verschiedener User als von einander unabhängige Ereignisse betrachtet werden können (und müssen), kann man als *window* das entsprechende Zeitfenster der ganzen Session wählen. Die Ereignisfolge, die WINEPI als Eingabe erhält, zerfällt dann in mehrere unabhängige Folgen, und die Häufigkeit einer Episode entspricht dann der Anzahl der Sessions(=Folgen), in denen diese aufgetreten ist. Somit entfällt auch der Schritt der Fensterverschiebung bei der Erkennung von Episoden.

Im Unterschied zu Apriori wird bei dieser Methode die Reihenfolge der Ereignisse berücksichtigt. Allerdings hat er den gleichen Nachteil wie Apriori, dass man im Falle der langen sich stark ähnelnden Ereignisfolgen zu einer enorm großen Menge der häufigen Episoden

kommt. So bringt die Anwendung des Verfahrens keine Vorteile gegenüber Apriori bei der Entdeckung häufiger Episoden. Das Verfahren arbeitet deswegen mit solchen Einschränkungen wie Zeitfenster, um einerseits die Länge der "interessanten" Episoden im Voraus zu reduzieren und andererseits um den Zusammenhang zwischen verschiedenen Ereignissen festzustellen, da die Zuordnung zu einem so genannten "Verursacher" nicht immer eindeutig ist.

2.7 Entdeckung von Interaktionsmustern

Das Problem der Entdeckung von Interaktionsmustern (Interaction-Pattern Mining) ist ähnlich zu dem Problem der Entdeckung von häufigen Episoden in Ereignisfolgen. Mit dieser Methode findet man entweder wahrscheinliche Muster oder bestimmte Muster mit Rauschen. Bei der Entdeckung von sehr langen Sequenzen braucht man eine effiziente Strategie, die sich in der Entdeckung von kurzen oder weniger verschwommenen Muster unter Einsatz flächendeckender Suche wieder findet. Dann werden Muster, die einen ausreichend großen Support-Wert haben, erweitert, um längere Muster zu bilden. Dieser Prozess wird fortgesetzt, bis keine Muster mehr entdeckt werden können [14].

Mit dieser Methode werden nur geordnete Muster entdeckt, auch wenn die Eingabesequenzen Fehler enthalten, die Fehler werden aber nur dann verworfen, wenn ihre Anzahl kleiner als eine vorgegebene Zahl ist. *IPM2* besteht aus 3 Schritten: Preprocessing der Eingabesequenzen, Musterentdeckung und Musteranalyse. Im Gegensatz zu Apriori meidet *IPM2* mehrfache Durchläufe über die Datenbank durch das Beibehalten der Adressenliste der Kandidatenmuster, die auch zu Erzeugung der Adressenliste der längeren zusammengesetzten Muster genutzt werden. *IPM2* ist eine Erweiterung des früheren Algorithmus *IPM* [13]. *IPM2* verwendet die Tiefensuche statt der vom *IPM* genutzten Breitensuche. Dadurch braucht man nicht mehr, alle Muster der Länge l und ihre Adressenlisten in einer Matrix $|A| \times |A|$ abzuspeichern, was sehr viel Speicherplatz verbrauchen kann. Aber er ist aufwendiger durch die Generierung von mehr Kandidatenmuster als *IPM*. *IPM2* erweitert die Kandidatenmuster durch das Ankleben der Muster der Länge *2*, bis das Muster keinen ausreichend großen Support aufweist, läuft dann zurück, und berichtet alle gefundene Muster, die vorgegebene Kriterien erfüllen [14].

2.7.1 Aufgabenstellung

Bevor die Beschreibung des Algorithmus vorgestellt wird, müssen alle notwendigen Notationen eingeführt werden [14].

1. Sei A ein Ereignisalphabet.
2. Sei $S = \{ s_1, s_2, \ldots s_n \}$ eine Sequenzmenge. Jede Sequenz s_i ist eine geordnete Menge von Ereignissen aus A.
3. Eine Episode e ist eine geordnete Menge von Ereignissen, die zusammen in einer bestimmten Sequenz auftreten.
4. Ein Muster p ist eine geordnete Menge von Ereignissen, das in jeder Episode $e \in E$ existiert, wobei E eine Menge von interessanten Episoden ist, die das vom User definierte Kriterium c erfüllen. Mit $e[i]$ wird das i-te Ereignis in der Episode e angesprochen. Mit $|e|$ und $|p|$ werden die Längen, d. h. die Anzahl der Ereignisse in e und p bezeichnet.
5. Wenn eine Menge von Episoden E das Muster p enthält, dann müssen das erste und das letzte Ereignis in p entsprechend das erste und das letzte Ereignis jeder Episode $e \in E$ sein und alle Ereignisse in p müssen in der gleichen Reihenfolge in e stehen, aber e darf auch andere Ereignisse enthalten, d.h. $|p| \leq |e|$ $\forall\, e \in E$. Formal,
   - $p[1] = e[1], \forall\, e \in E$
   - $p[\,|p|\,] = e[\,|e|\,], \forall\, e \in E$ und
   - $\forall\, (i,j)\ 1 \leq i \leq |p|$ und $1 \leq j \leq |p|$ und $i < j, \exists\, e[k] = p[i]$ und $e[l] = p[j]$, so dass $k < l$.
6. Die Adressenliste des Musters p, bezeichnet $loclist(p)$, ist eine Liste von Tripels $(seqnum, startLoc, endLoc)$, jeder ist die Adresse der Episode $e \in E$, wobei s_{seqnum} die Sequenz, die e enthält, $startLoc$ unf $endLoc$ Adressen von $e[1]$ und $e[\,|e|\,]$ in s_{seqnum}.
7. Die Häufigkeit des Musters p, bezeichnet $support(p)$, ist die Anzahl der Episoden in S, die p beinhalten. Somit gilt: $support(p) = loclist(p).length$, die Länge der $loclist(p)$.
8. Die Dichte des Musters p, das von der Episodenmenge E gedeckt ist, bezeichnet $density(p)$ ist der Quotient von $|p|$ und der durchschnittlichen Episodenlänge der Episoden $\in E$:

   $density\ (p) = |p| * support(p) / \sum_{e \in E} |e|$

9. Ein Qualifikationskriterium *c*, oder einfach Kriterium, ist ein benutzerdefiniertes Quadrupel (*minLen*, *minSupp*, *maxError*, *minScore*). Gegeben ist das Muster *p*, die minimale Länge *minLen* ist der Grenzwert für $|p|$. Die minimale Häufigkeit *minSupp* ist der Grenzwert für *support*(*p*). Der maximale Fehler *maxError* ist die maximale Anzahl der Eingangsfehler (evtl. Störungen, Ereignisse die nicht beachtet werden), die für jede Episode $e \in E$ zulässig sind. Die minimale Wertung *minScore* ist der Grenzwert für Auswertungsfunktion, die zur Bestimmung des Rangs für die entdeckten Muster genutzt wird. Diese Funktion kann z.B. lauten:

   $score(p) = log_2 |p| * log_2 support(p) * density(p)$

   Diese Funktion hat sich nach vielen Experimenten von Entwicklern des *IPM2* als angemessen erwiesen, weil sie Musterlänge, Musterhäufigkeit und Musterdichte berücksichtigt und diese Werte gewichtet.

10. Ein *maximales Muster* ist ein Muster, das kein Teilmuster eines anderen Muster mit dem gleichen Support ist.
11. Ein *zugelassenes Muster* ist ein Muster, für das die benutzerdefinierten Kriterien zutreffen.
12. Ein *Kandidatenmuster* ist ein Muster in der Analyse, das Bedingungen *minSupp* und *maxError* erfüllt.

Nach diesen Definitionen kann das Problem der Entdeckung von Interaktionsmustern folgendermaßen definiert werden:

Gegeben :
1) ein Alphabet *A*,
2) eine Sequenzmenge *S* und
3) das Kriterium *c*.

Finde alle maximalen zugelassenen Muster in *S*.

2.7.2 Algorithmusbeschreibung

2.7.2.1 Preprocessing

Die Interaktionsfolge wird üblicherweise als Sequenz von Zahlen dargestellt. Diese Repräsentation wird als *R0* bezeichnet. *R0* enthält oft Wiederholungen, die auf die Besonderheiten der Eingabedaten zurückzuführen sind. Diese Wiederholungen können die Entdeckung von

Mustern beträchtlich hindern, deswegen werden die unmittelbaren Wiederholungen durch ein Ereignis ersetzt. Diese Repräsentation wird *R1* bezeichnet. In der Abbildung 2.7.1 wird ein Beispiel gezeigt.

R0 : {1,2,3,4,5,6,6,6,6,6,6,7,7,8,8,8,9,7,7,7}
R1 : {1,2,3,4,5,6,7,8,9,7}

Abb. 2.7.1: Preprocessing von Interaktionsfolgen [14]

2.7.2.2 Musterentdeckung mit IPM2

Die Eingabe für *IPM2* ist die Sequenzmenge *S* und Kriterium *c*. Die Ausgabe von *IPM2* sind alle maximale zugelassene Muster in *S*. *IPM2* besteht aus 2 Phasen [14].

Erstens, sucht der Algorithmus flächendeckend in den Eingabesequenzen alle Kandidatenmuster der Länge 2, die *minSupp*- und *maxError*-Bedingungen erfüllen (Prozedur 1). Für jedes solche Muster wird eine Adressenliste gebildet. Die Muster werden in einem Vektor der Länge $|A|$ *ptListVec* gespeichert, dessen Zellen nach den Ereignissen aus A bezeichnet werden. Jede Zelle *ptListVec*[*i*] enthält alle Muster *p*, so dass *p*[1]=*i*. Z.B. das Muster {1,3} ist in *ptListVec*[1] gespeichert [14].

In der zweiten Phase (Prozedur 2) erweitert der Algorithmus rekursiv jedes Kandidatenmuster aus *ptListVec* nach dem "depth-first"-Printip. Falls eine Zusammensetzung des Kandidatenmusters *p1* mit einem anderen Muster *p2* ein neues Kandidatenmuster *p3* = *p1*+ *p2*[2] erzeugt, wird *p3* entsprechend erweitert. *p1* kann nur mit Muster in *ptListVec*[*p1*[|*p1*|]] zusammengefügt werden, z.B. Muster der Länge 2, dessen erstes Ereignis dasselbe ist wie das letzte von *p1*. Die Adressenlisten von *p1* und *p2* werden bei der Konstruktion der Adressenliste von *p3* genutzt (Prozedur 3). Die Adressen der Episoden, die *p3* abdecken, aber mehr Eingangsfehler als *maxError* enthälten, werden entfernt. Wenn *support*(*p3*) ≥ *minSupp*, dann wird *p3* noch Mal durch Muster in *ptListVec*[*p3*[|*p3*|]] erweitert, anderenfalls wird *p3* ignoriert und der Algorithmus speichert *p1*, falls es zugelassen wird, und läuft dann zurück. Während der Zurückverfolgung und nach dem Berichten von *p1* überprüft der Algorithmus das Elternmuster *p0* von *p1*. Weil *p0* das Teilmuster von *p1* ist, ist es auch ein Kandidatenmuster. Falls *p0* zugelassen ist und *support*(*p0*) > *support*(*p1*), d.h. es ist nicht nichtmaximal in Bezug auf *p1*, dann wird es auch gespeichert. Nachdem alle Muster aus *ptListVec* erweitert wurden, werden nichtmaximale Muster gelöscht und nur maximale zugelassene Muster berichtet [14].

Der Algorithmus in der Abbildung 2.7.2 beschreibt die Prozedur 1 von IPM2. In der Zeile 1 wird ein Vektor *ptListVec* initialisiert. *PatternList* ist eine Hashtabelle-ähnliche Datenstruktur, die eine Liste von Mustern enthält. Zeilen 3 bis 14 werden für jede Eingabesequenz $s_k \in S$ ausgeführt. Die Schleife, die in der Zeile 3 beginnt, iteriert über alle Ereignisse von s_k, von $s_k[1]$ bis $s_k[|s_k| - maxError - 1]$. In Zeilen 4 und 5 wird jedes Ereignis zum Generieren eines Musters genutzt, beginnend mit dem nächst folgenden und bis zu *maxError* +1. Z.B. wenn s_k = {1,3,2,3,4,3} und *maxError* = 2, das erste Ereignis wird mit jedem von den nächsten 3 kombiniert, so dass die Muster {1,3}, {1,2}, {1,3} sich ergeben. In Zeilen 6 und 7 wird das neue Muster zu *ptListVec* hinzugefügt, falls es noch nicht da ist. Die Adresse der Episode, die dieses Muster enthält, wird zu Adressenliste in Zeile 8 hinzugefügt. Zeilen 9 bis 14 führen das gleiche wie Zeilen 2 bis 8 für die letzten *maxError* Ereignisse der s_k. Schritte 15 bis 18 entfernen alle nicht zugelassene Muster p, Muster mit $support(p) < minSupp$.

***Prozedur 1**: Erzeugung der Kandidatenmenge von Anfangsmustern*

***Input**: Ein Alphabet A, ein Kriterium c und eine Menge von Sequenzen S*

***Ouput**: Alle Kandidatenmuster der Länge 2*

```
1.PatternList  ptListVec[|A|]
2.    For every trace s_k ∈ S, 1 ≤ k ≤ | S |
3.        For i = 1 to |s_k| - maxError -1
4.            For j = i +1 to i + maxError + 1
5.                Construct new pattern p = s_k[i] + s_k[j]
6.                If p NOT in ptListVec[s_k[i]]
7.                    then Add p to ptListVec[s_k[i]]
8.                Add (k,i,j) to locList(p)
9.        For i= |s_k| - maxError  to |s_k| -1
10.           For j = i+1 to |s_k|
11.               Construct new pattern p = s_k[i] + s_k[j]
12.               If p NOT in ptListVec[s_k[i]]
13.                  then Add p to ptListVec[s_k[i]]
14.               Add (k,i,j) to locList(p)
15.   For every id ∈ A
16.       For every pattern p in ptListVec[id]
17            If loclist(p).length < minSupp
18.               then Remove p from  ptListVec[id]
```

Abb. 2.7.2: *Prozedur 1* von *IPM2* [14]

***Prozedur 2**: Generierung von Langen Kandidatenmustern aus den Kurzen*

***Input**: Ein Vektor von Pattern Lists, der mit allen Kandidatenmustern der Länge 2 und ihren Location Lists initialisiert ist, und ein Kriterium c.*

***Ouput**: Alle maximale Muster, die entsprechend c qualifiziert sind.*

```
1.  PatternList resultsIPM2
2.  For every id ∈ A
3.      For every pattern p in ptListVec[id]
4.          PatternList tempResults = Extend(p)
5.          Merge tempResults with resultsIPM2
6.          Remove non-maximal patterns from resultsIPM2
7.          Report resultsIPM2
```

PatternList **Extend**(*p1*)

```
1.  PatternList extensionResults
2.  For every pattern p2 in ptListVec[p1[|p1|]]
3.      Construct new pattern p3 = p1 + p2 [|p2|]
4.      Construct the location list of p3 (Prozedur 3)
5.      If support(p3) ≥ minSupp then
6.          PatternList tempResults = Extend(p3)
7.          Merge tempResults with extensionResults
8.          If support(p1) > support (p3)
9.              If |p1| ≥ minLen AND score(p1) ≥ minScore
10.                 If p1 is NOT in extensionResults
11.                     Add p1 to extensionResults
12.     Else
13.         If |p1| ≥ minLen AND score (p1) ≥ minScore
14.             If p1 is NOT in extensionResults
15.                 Add p1 to extensionResults
16. Return extensionResults
```

Abb. 2.7.3: *Prozedur 2* von *IPM2* [14]

Der Algorithmus in der Abbildung 2.7.3 beschreibt die Prozedur 2 von *IPM2*. In der Zeile 1 wird eine Liste von Mustern, bezeichnet *resultsIPM2*, erzeugt, in der die entdeckten Muster gespeichert werden. Die Schleife von der Zeile 2 iteriert über alle Zelle aus *ptListVec*. Die Schleife von der Zeile 3 iteriert über alle Muster *ptListVec*[*id*]. Für jedes solche Muster *p* wird in der Zeile 4 eine Subprozedur *Extend*(*p*) aufgerufen, die alle erweiterte zugelassene Muster von *p* liefert, die ebenfalls maximal in Bezug zu einander sind, d.h. dass kein von ihnen ein Teilmuster von

einem anderen mit dem gleichen Support ist. In der Zeile 5 werden die gefundenen Erweiterungen zu *resultsIPM2* hinzugefügt. Der Schritt 6 entfernt alle nichtmaximale Muster aus den endgültigen Ergebnissen. In der Zeile 7 werden alle in *resultsIPM2* übrig gebliebenen Muster ausgegeben.

Die Subprozedur *Extend*(*p1*) arbeitet folgenderweise. Im Schritt 1 wird die Liste von Mustern *extensionResults* erzeugt, in der die Muster abgespeichert werden, die sich nach den erfolgreichen Erweiterungen des Parameters *p1* ergeben haben. In der Zeile 2 beginnt die Schleife, die über alle Muster *p2* iteriert, die *p1* erweitern können, d.h. alle Muster, deren erstes Ereignis das gleiche wie das letzte von *p1* ist. Schritte 3 und 4 konstruieren das erweiterte Muster *p3* und seine Adressenliste. In der Zeile 5 wird die Bedingung $support(p3) \geq minSupp$ überprüft. Schritte 6 bis 11 werden in dem Fall ausgeführt, wenn die Bedingung erfüllt ist, und anderenfalls Schritte 13 bis 15. Im Falle der erfolgreichen Zusammensetzung erweitert der Schritt 6 rekursiv den neuen Kandidaten *p3* durch den Aufruf *Extend*(*p1*) mit *p3* als Parameter. In der Zeile 7 werden die zugelassenen Muster, die sich durch Erweiterung von *p3* ergeben haben, zu *extensionResults* hinzugefügt. Die Schritte 8 bis 11 erweitern die *extensionResults* um *p1*, wenn es den größeren Support-Wert als seine erfolgreiche Erweiterung *p3* hat, zugelassen ist und nicht in *extensionResults* enthalten ist. Missglückt die Zusammensetzung von *p1* und *p2*, wird das erweiterte Muster ignoriert und in Zeilen 13 bis 15 wird *p1* zu Ergebnisliste *extensionResults* hinzugefügt, wenn es zugelassen wird und nicht bereits in *extensionResults* enthalten ist. Anschließend werden alle zugelassene maximale (in Bezug zu einander) Erweiterungen von *p1* zurückgegeben.

Der Algorithmus in der Abbildung 2.7.4 beschreibt die Prozedur 3 von *IPM2* zur Generierung von Adressenliste für ein neues Kandidatenmuster. Er kombiniert die Adressenlisten von 2 Mustern *p1* und *p2*, wobei $|p2|=2$, um die Adressenliste von *p3* zu liefern, wobei $p3 = p1 + p2[2]$. Der Schritt 2 iteriert über alle Adressen von Episoden, die *p1* enthalten. Der Schritt 3 liefert eine solche Episode *e1*. Im Schritt 4 werden die Adressen der Episoden, die *p2* enthalten und andere Bedingungen erfüllen, gefunden. Wenn so eine Episode gefunden wird, dann:

- müssen *e1* und *e2* in einer Sequenz sein
- *e2* darf nicht die Teilsequenz von *e1* sein und umgekehrt
- *e1* und *e2* müssen sich nur an einer Stelle, nämlich an $e1[|e1|]$, überlappen
- der Abstand zwischen *startLoc* von *e1* und *endLoc* (einschließlich) von *e2* darf nicht größer sein als $|p1|+1+maxError$.

In Zeilen 5 bis 7 wird die Adressenliste von *p3* konstruiert und Duplikate aus ihr entfernt. Schließlich wird die resultierende Adressenliste von *p3* zurückgegeben.

Procedur 3: *Konstruieren der Location List von Kandidatenmuster.*

Input*: Die Location Lists von Mustern p1 und p2 und maxError. Die listen sind nach seqnum und startLoc sortiert.*

Ouput*: Die Location List von p3, wo p3 = p1 + p2[2]*

1. **Create** an empty location list *listLoc3*
2. **For** *i* =1 **to** *loclist(p1).length*
3. *loc1* = location *i* in *loclist(p1)*
4. **Find a set** *Loc1 = (* any *loc2* ∈ *loclist(p2))* **such that**
   loc2.seqnum = loc1.seqnum AND
   loc2.startLoc = loc1.endLoc AND
   loc2.endLoc<=loc1.startLoc + maxError + |p1|
5. **For every** *loc1* ∈ *Loc1*
6. **Add** *(loc1.seqnum, loc1.startLoc, loc2.endLoc)* to *listLoc3*
7. **Remove** any duplicates from *listLoc3*
8. **Return** *listLoc3*

Abb. 2.7.4: *Prozedur 3* von *IPM2* [14]

2.7.2.3 Verstehen von gefundenen Mustern

Nach der Begutachtung von entdeckten Mustern kann Kriterium *c* so modifiziert werden, dass man die Ergebnismenge kleiner oder größer wird, je nachdem ob es zu viele oder zu wenige Muster gefunden wurden. Durch Verändern von verschiedenen Bestandteilen des Kriteriums *c* stellt man unterschiedliche Anforderungen an die Ergebnismuster. So können Muster mit verschiedenen minimalen Längen, mit verschiedenen minimalen Häufigkeiten oder mit unterschiedlichen Dichten, ja nachdem was interessant erscheint, analysiert werden. Außerdem kann eine Gruppe von Muster, deren Wertung und/oder Häufigkeit in einem spezifischen Bereich liegen, ausgesucht und verkleinert werden durch das Entfernen von Mustern, die Teilmuster von anderen sind, auch wenn sie maximal sind.

Kapitel 3

Datenanalyse

3.1 Ansatz

In dieser Diplomarbeit werden verschiedene Data Mining Techniken eingesetzt. Dabei müssen Gütekriterien wie Eignung der Methoden zur Analyse der Klickfolgen, d.h. Unempfindlichkeit gegen eventuelle Störungen in der Eingabemenge und für verschiedene Eingabeparameter wie Häufigkeit und Konfidenz, sowie Interpretierbarkeit der Ergebnisse, d.h. nicht zu große Ergebnismenge, die sich deuten lässt, beachtet werden.

Nach der Extraktion der notwendigen Daten werden sie mit Methoden wie Apriori, WINEPI, Abgeschlossene Itemmengen und IPM2 untersucht und interessante Informationen über Nutzungsverhalten dargelegt. Es wird unter anderem vorgeschlagen, welche Informationen über unerwartete Nutzung entdeckt und wie Nutzungsverhalten visuell präsentiert werden können.

3.2 Daten

Die Analyse des Benutzerverhaltens wird anhand der Daten aus den speziellen Logfiles durchgeführt.

Diese Logfiles werden von der Web-Applikation generiert, durch die ein großes Unternehmen ihre Leistungen online anbietet. Die Leistungen können nur von registrierten Firmen bzw. Personen in Anspruch genommen werden. Benutzer können serielle (aber auch einzelne) Briefe, E-Mails, Postkarten oder Verteiler in Auftrag geben. Der Anbieter stellt mehrere Hilfsmittel, die üblicherweise beim Design von Textdokumenten und Graphiken verwendet werden, und Hilfswerkzeuge wie Adressendatenbanken bereit, so dass durch ihre Nutzung unterschiedliche Muster in Zugriffspfaden zu Stande kommen.

Ähnlich wie bei Serveraccesslogs werden in diesen Logfiles IP-Adresse des Benutzers, Datum und Zeit, URL und Zugriffsart protokolliert. Bei jedem Zugriff auf die Website wird eine neue Session angelegt. Die Session-ID wird als URL-Parameter weitergegeben und in dem Logfile gespeichert. Da die Kunden sich zuerst einloggen müssen, ist die Session-ID mit Kundennamen fest verknüpft und so kann man durch die Session-ID den Benutzer identifizieren. Jeder Zugriff kann eindeutig zu einer Session und somit einem bestimmten User zugeordnet werden.

Ein Eintrag könnte z.B. so aussehen:

123.123.123.123 - - [01/Jan/2000:15:02:10 –0800] “GET /info/index.html HTTP/1.0” 200 6822 “http://www.ihrefirma.de/index.html” “Mozilla/4.01 (WinNT)” sessionId=zufällig_generierte_zeichenkette

In diesen Logfiles werden unter anderem Fehlermeldungen und andere von den Entwicklern eingefügte Debugmeldungen protokolliert. Diese Einträge werden z.B. bei der Useridentifikation genutzt, andere werden beim Extrahieren der relevanten Daten ignoriert.

3.3 Preprocessing

Die Ermittlung der Assoziationsregeln und Statistiken ist nur dann nützlich, wenn die Daten aus den Logfiles ein genaues Bild der Benutzerzugriffe auf der Website widerspiegeln. Deswegen müssen Daten zuerst aus dieser Informationsmenge extrahiert und für weitere Verarbeitung vorbereitet werden.

Erster Schritt, Preprocessing genannt, umfasst Datenfilterung, Benutzeridentifikation, Sessionidentifikation und Präsentation der Transaktionen und der Sitetopologie in verschiedenen Datenstrukturen. Alle diese Aufgaben werden parallel beim Auslesen der Logfiles durchgeführt. Während des Preprocessing werden Zeilen des Logfiles in eine Liste von Besucher-Sessions “übersetzt“.

Die Web-Applikation generiert beim Einloggen sowohl neue Session-Objekte als auch entsprechende Eintragungen mit der Eingabe des eingeloggten Kunden in diesen Logfiles, so dass durch bestimmte Schlüsselwörter diese Zeilen erkannt werden. Falls der Benutzer seine Sitzung nicht durch Abmelden beendet, wird das zugeordnete Session-Objekt spätestens beim Datumwechsel zerstört. So kann der User während des Tages mit dem gleichen Session-Objekt mehrfach auf die Website zugreifen. Allerdings ist die Applikation so gestaltet worden, dass nach der Wahl eines Produktes der Wechsel zu einem anderen nicht möglich ist. In diesem Fall muss der Benutzer eine neue Sitzung

beginnen, somit ist eine Session immer mit einem Produkt verknüpft. Da jeder Zugriff zusammen mit der entsprechenden Session-ID protokolliert wird, ist es leicht, alle Transaktionen einer Sitzung zuzuordnen. So wird jedem User eine Liste der Sessions mit Transaktionen zugeordnet. Um weitere Bearbeitung von Daten zu vereinfachen, wird jeder Seite eine *id* zugeordnet.

O=(*name, id, date, time, duration*), wobei O ein Seite-Objekt mit *name* als Bezeichnung der Seite, *id* die der Referenzseite zugeordnete Zahl, *date*, *time* und *duration* Datum, Zeit des Zugriffs und Verweilzeit sind.

S=(*id*, o_1,o_2, ... o_n), wobei S ein Session-Objekt mit *id* als Session-ID und o_i mit $i=1,\ldots n$ Referenzseiten und n die Anzahl der besuchten Seiten sind.

Während der Preprocessing-Phase wird überprüft, ob es o_i und o_{i+1} mit o_i.*name* = o_{i+1}.*name* gibt. Falls dies zutrifft, wird o_{i+1} aus der Referenzseitenliste des entsprechenden Session-Objekts S entfernt.

U=(*name*, s_1, s_2, …s_m), wobei U ein User-Objekt mit *name* als Username und s_i mit $i=1,...m$ seine Sessions sind.

Die Sessions mit Referenzseiten werden durch eine Liste repräsentiert und für die Sitetopologie wird ein Graph initialisiert. Jeder Knoten des Graphen stellt eine Seite und jede Kante einen direkten Übergang von einer Seite zu einer anderen dar. Dabei werden nur die Seitenzugriffe in Betracht gezogen, die durch Filterung nicht entfernt wurden. Ich habe z.B. solche Zugriffe wie auf die Pop-Up-Fenster mit Hilfsinformationen, Registrierungsseiten und die Objekte, die zur Darstellung der Graphiken dienen, ignoriert. Die Registrierungsaktivitäten waren nicht der Bestand der Analyse und Pop-Up-Fenster erscheinen an verschiedenen Stellen der Applikation mit unterschiedlichem Inhalt, so dass ihre Rolle in einem Zugriffspfad nicht immer eindeutig zu bestimmen ist. Deswegen wurden solche Objekte in die Analyse nicht einbezogen.

3.4 Statistische Analyse

Um einen ersten Eindruck über Userverhalten zu verschaffen, werden Statistiken erstellt. Nachdem Daten gefiltert wurden, werden Transaktionen aus allen Sitzungen in einer Datei gespeichert, d.h. jede Zeile repräsentiert eine Sitzung und enthält eine Zahlenfolge, die Seitenzugriffe darstellt.

Jede Zeile hat das Format

Id:SessionId o_1*.id:1* o_2*.id:1 . . .* o_l*.id:1*

wobei *SessionId* die Session-ID, o_i*.id* die *id* des Seite-Objektes o_i sind und «*:1*» bedeutet, dass die Referenzseite o_i in der Session mit *SessionId* besucht wurde.

In einer anderen Datei wird die Zuordnung zwischen den Zahlen und Seitenbezeichnungen gespeichert. Diese 2 Dateien sind die Eingabedateien für den *SparseFormatExampleSource–Operator* von *YALE*[5], der als Ergebnis die Häufigkeit des Auftretens (in %) für jede Seite liefert.

Für die Statistiken, die die Anzahl der Besucher pro Seite oder Häufigkeit des Abbruchs für jede Seite berichten sollen, wird das Format der ersten Datei entsprechend modifiziert. Bei der Verweilstatistik sieht es folgendermaßen aus:

Id:SessionId o_1*.id:dur1* o_2*.id:dur2 . . .* o_l*.id:duri*

duri entspricht hier der durchschnittlichen Verweildauer auf der Seite o_i in der Session mit *SessionId*, d.h. wenn diese Seite mehrmals während der Sitzung angeschaut wurde, wurde die durchschnittliche Dauer ermittelt. Bei dieser Statistik werden Durchschnittswerte nicht über alle Sitzungen gebildet, sondern nur über die Sitzungen, in denen die Seite o_i angeklickt wurde.

3.4.1 Besuchstatistik/Bestellstatistik

Die ersten 30 Hit-Seiten, sortiert nach der Häufigkeit des Auftretens, sind in Abb. 3.4.1 zu sehen. Es wurde Statistik zu 244 Webseiten der Applikation erstellt. Für jede Webseite bedeuten *min*=0.0, dass die Seite nicht von jedem Besucher angeklickt (bzw. angeschaut) wurde, *max*=100.0, dass die Seite von mindestens einem Besucher angeklickt (bzw. angeschaut) wurde, und *avg* die Prozentzahl der Benutzer, die jeweilige Seite mindestens einmal besucht haben. Zur Prozentzahl *avg* wird zur besseren Übersichtlichkeit in der letzten Spalte auch der tatsächliche Wert angegeben. Die «*index*» und «*distribution/index*» wurden z.B. von 42 bzw. 41 Kunden (von 45 insgesamt) in 225 bzw. 137 Sessions (s. Abb. 3.4.2) besucht, was 93,3% bzw. 91,1% aller Kunden entspricht. Es gibt aber 29 Seiten, auf die nur 1 oder 2 Besucher zugegriffen haben. In dieser Statistik sind nur die Seiten berücksichtigt,

[5] YALE = Yet Another Learning Environment, entwickelt von Artificial Intelligence Unit of the University of Dortmund

die von mindestens einem Besucher wenigstens einmal “gesehen“ worden sind, d.h. alle anderen Referenzseiten wurden entweder von keinem besucht oder aus der Analyse ausgeschlossen. Aus der Bestellstatistik (s. Abb. 3.4.3) kann man entnehmen, dass es 26 Kunden gegeben hat, die einen Email-Auftrag aufgegeben haben, 22 einen Brief-Auftrag, 11 einen Distribution-Auftrag und nur 6 einen Postkarte-Auftrag. Es sind aber 41 Kunden, die auf die Startseite für das Distribution-Produkt, 36 für Email, 34 für Brief, 18 Postkarte geklickt haben. So sieht man, dass die Relation zwischen der Anzahl derjenigen, die sich für ein Produkt interessieren oder nur zufällig zu diesem Produkt gekommen sind und derjenigen, die das Produkt bestellen, stark schwankt. Man kann hier aber nicht erkennen, welchen Pfaden Benutzer durch die Website folgen, ob sie den kurzen geplanten Weg gewählt haben oder ihr Verhalten eher chaotisch ist. Die Statistik berichtet auch, dass 40 Besucher (88,8%) ihre 126 Sessions (31,4%) durch Ausloggen beendet haben. Man kann aus diesen Berichten nicht sehen, an welchen Stellen (außer Ausloggen) und warum Kunden ihre Bestellungen abgebrochen haben. Zu diesem Zweck wäre es interessant, die Statistik zu Abbruchseiten und die Verweilstatistik zu betrachten.

Number of Users: 45
Number of Websites: 244

Index	Website Name	User Visit statistik [in %]			Absolute
		min	max	avg	
3	index	0.0	100.0	93.3	42
4	project/chooseproduct	0.0	100.0	91.1	41
126	distribution/index	0.0	100.0	91.1	41
130	distribution/project/projectwelcome	0.0	100.0	91.1	41
52	logout	0.0	100.0	88.8	40
107	email/index	0.0	100.0	80.0	36
132	distribution/project/projectwork	0.0	100.0	80.0	36
133	distribution/project/templatechoice	0.0	100.0	80.0	36
6	letter/index	0.0	100.0	75.5	34
135	distribution/sender/agencieschoice	0.0	100.0	75.5	34
108	email/project/projectwelcome	0.0	100.0	73.3	33
110	email/project/projectwork	0.0	100.0	73.3	33
111	email/address/address	0.0	100.0	73.3	33
208	email/address/addressuploadfile	0.0	100.0	73.3	33
213	email/address/addressfieldallocation	0.0	100.0	73.3	33
7	project/projectwelcome	0.0	100.0	71.1	32
9	project/projectwork	0.0	100.0	71.1	32
10	address/address	0.0	100.0	71.1	32
128	distribution/login/login	0.0	100.0	71.1	32
11	address/addressupload	0.0	100.0	66.6	30
112	email/mailing/emailsortchoice	0.0	100.0	66.6	30
214	email/address/addressfieldcheck	0.0	100.0	66.6	30
8	project/projectnew	0.0	100.0	64.4	29
113	email/mailing/onlinetext	0.0	100.0	64.4	29
114	email/mailing/emaildata	0.0	100.0	64.4	29
207	email/project/projectnew	0.0	100.0	64.4	29
218	email/order/auftragoption	0.0	100.0	64.4	29
219	email/order/auftragdata	0.0	100.0	64.4	29
220	email/order/ordersupply	0.0	100.0	64.4	29
36	order/ordersupply	0.0	100.0	62.2	28

Abb. 3.4.1: 30 Top-Seiten der Statistik “#User pro Seite“

Number of Sessions: 401
Number of Websites: 244

Index	Website Name	Visit Statistik [in %]			Absolute
		min	max	avg	
3	index	0.0	100.0	56.1	225
4	project/chooseproduct	0.0	100.0	43.6	175
126	distribution/index	0.0	100.0	34.1	137
130	distribution/project/projectwelcome	0.0	100.0	33.6	135
52	logout	0.0	100.0	31.4	126
107	email/index	0.0	100.0	26.1	105
6	letter/index	0.0	100.0	24.6	99
108	email/project/projectwelcome	0.0	100.0	24.6	99
110	email/project/projectwork	0.0	100.0	24.6	99
111	email/address/address	0.0	100.0	24.1	97
208	email/address/addressuploadfile	0.0	100.0	23.6	95
7	project/projectwelcome	0.0	100.0	22.4	90
132	distribution/project/projectwork	0.0	100.0	22.1	89
213	email/address/addressfieldallocation	0.0	100.0	22.1	89
9	project/projectwork	0.0	100.0	21.6	87
10	address/address	0.0	100.0	21.6	87
133	distribution/project/templatechoice	0.0	100.0	20.4	82
112	email/mailing/emailsortchoice	0.0	100.0	19.9	80
214	email/address/addressfieldcheck	0.0	100.0	19.4	78
113	email/mailing/onlinetext	0.0	100.0	18.9	76
114	email/mailing/emaildata	0.0	100.0	18.9	76
218	email/order/auftragoption	0.0	100.0	18.4	74
219	email/order/auftragdata	0.0	100.0	18.4	74
220	email/order/ordersupply	0.0	100.0	18.4	74
36	order/ordersupply	0.0	100.0	17.4	70
37	order/ordershow	0.0	100.0	17.4	70
221	email/order/ordershow	0.0	100.0	16.9	68
222	email/order/ordersend	0.0	100.0	16.9	68
223	email/billingserver/billingserver	0.0	100.0	16.9	68
224	email/billingserver/billingserverbill	0.0	100.0	16.9	68

Abb. 3.4.2: 30 Top-Seiten der Statistik “#Sessions pro Seite“

126	distribution/index	0.0	100.0	91.1	41
163	distribution/order/orderthankyou	0.0	100.0	24.4	11
107	email/index	0.0	100.0	80.0	36
227	email/order/orderthankyou	0.0	100.0	57.7	26
6	letter/index	0.0	100.0	75.5	34
43	order/orderthankyou	0.0	100.0	48.8	22
55	postcard/index	0.0	100.0	40.0	18
94	postcard/order/orderthankyou	0.0	100.0	13.3	6

Abb. 3.4.3: Bestellstatistik

Number of Users: 45
Number of Websites: 244

Index	Website Name	Start-Page-Statistik [in %]			Absolute
		min	max	avg	
3	index	0.0	100.0	93.3	42
126	distribution/index	0.0	100.0	84.4	38
107	email/index	0.0	100.0	53.3	24
55	postcard/index	0.0	100.0	13.3	6
1	portal_content_start	0.0	100.0	6.7	3
6	letter/index	0.0	100.0	4.4	2
128	distribution/login/login	0.0	100.0	4.4	2

Abb. 3.4.4: 7 Startseiten

3.4.2 Startseiten-Statistik

Diese Statistik (s. Abb. 3.4.4) zeigt, dass nur 7 Seiten als Startseiten genutzt wurden. Die meisten User nutzten die «*index*» und «*distribution/index*» als Startseiten. Wenn man diese Statistik mit der Besuchstatistik vergleicht, sieht man, dass es viel mehr User gegeben hat, die die so genannten Startseiten besucht haben, als solche, die mit diesen Seiten gestartet haben. Das bedeutet, dass die meisten Besucher durch «*index*» und «*distribution/index*» auf die anderen Seiten kommen,

die ebenfalls als Startseiten genutzt werden können. So kann man «*index*», «*distribution/index*» und «*email/index*» als "echte" Startseiten qualifizieren. Vermutlich werden diese Seiten von meisten Besuchern zu Bookmarks hinzugefügt und die Anwendung wird öfters durch sie gestartet.

3.4.3 Abbruchstatistik

Es gibt insgesamt 64 Abbruchseiten, d.h. 64 (von 244) Seiten, wo mindestens einmal abgebrochen wurde. Man sieht wieder (s. Abb. 3.4.5 und Abb. 3.4.6), dass 126 Sessions von 40 Kunden durch Ausloggen beendet wurden. Man kann aber daraus nicht ablesen, ob diese Besucher ihre Sitzungen erst nach dem Bestellen beendet oder nachdem sie gewünschte Informationen gefunden oder dass sie ihre Bestellungen bzw. Suche erfolglos abgebrochen haben. Man entnimmt dieser Statistik, dass viele Kunden ihre Sessions nach dem Einloggen (44,4% bei «*distribution/login/checklogin*») abgebrochen haben, dass auch oft User die Sitzungen wegen eines Auftragfehlers (bei 17,7% aller Kunden) oder beim Distribution-Auftrag gleich bei den ersten Schritten der Bearbeitung des Projektes (bei «*distribution/project/projectwelcome*» – 24,4%) oder nach der Auswahl der Vorlage (bei «*distribution/sender/agencieschoice*» – 20,0%) beendet haben. Diese Information deutet darauf, dass auf diesen Seiten möglicherweise Benutzer auf Schwierigkeiten stoßen, die sie nicht überwinden können. Es können unverständliche Hinweise oder Fehlermeldungen sein. 13 User haben in 26 Sessions gleich nach der Bestätigung ihres Brief-Auftrages die Sitzung beendet. Diese Information ist für den Anbieter uninteressant, weil dies dem erwarteten Verhalten entspricht, da viele Kunden ihre Sitzungen trotz mehrfachen Mahnungen nicht explizit beenden. Falls Anbieter das Beenden einer Sitzung nur durch Abmelden fordern, müssen sie dafür sorgen, dass dies nach einer bestimmten Zeit automatisch erfolgt.

Number of Users: 45
Number of Websites: 244

Index	Website Name	Break-Off-Statistik[in %]			Absolute
		min	max	avg	
52	logout	0.0	100.0	88.8	40
129	distribution/login/checklogin	0.0	100.0	44.4	20
43	order/orderthankyou	0.0	100.0	28.8	13
130	distribution/project/projectwelcome	0.0	100.0	24.4	11
128	distribution/login/login	0.0	100.0	22.2	10
135	distribution/sender/agencieschoice	0.0	100.0	20.0	9
229	email/reporting/report	0.0	100.0	20.0	9
51	order/orderbillingerror	0.0	100.0	17.7	8
166	distribution/info/showtemplates	0.0	100.0	13.3	6
2	login/checklogin	0.0	100.0	11.1	5
124	info/brief/info_service_prz	0.0	100.0	11.1	5
137	distribution/address/address	0.0	100.0	11.1	5
241	postcard/login/checklogin	0.0	100.0	11.1	5
47	order/order	0.0	100.0	8.8	4
94	postcard/order/orderthankyou	0.0	100.0	8.8	4
112	email/mailing/emailsortchoice	0.0	100.0	8.8	4
126	distribution/index	0.0	100.0	8.8	4
196	info/info_kontakt	0.0	100.0	8.8	4
227	email/order/orderthankyou	0.0	100.0	8.8	4
88	postcard/order/ordererror	0.0	100.0	6.6	3
125	letter/login/checklogin	0.0	100.0	6.6	3
132	distribution/project/projectwork	0.0	100.0	6.6	3
139	distribution/address/previewie	0.0	100.0	6.6	3
140	distribution/order/sendoptions	0.0	100.0	6.6	3
143	distribution/info/help	0.0	100.0	6.6	3
177	distribution/address/addressfieldcontrol	0.0	100.0	6.6	3
213	email/address/addressfieldallocation	0.0	100.0	6.6	3
4	project/chooseproduct	0.0	100.0	4.4	2
34	order/mailingprintoptions	0.0	100.0	4.4	2
59	postcard/address/address	0.0	100.0	4.4	2

Abb. 3.4.5: Ausschnitt aus der Abbruchstatistik(#User pro Abbruchseite)

Number of Sessions: 401
Number of Websites: 244

Index	Website Name	Break-Off-Statistik [in %]			Absolute
		min	max	avg	
52	logout	0.0	100.0	31.4	126
129	distribution/login/checklogin	0.0	100.0	8.7	35
43	order/orderthankyou	0.0	100.0	6.4	26
130	distribution/project/projectwelcome	0.0	100.0	3.7	15
135	distribution/sender/agencieschoice	0.0	100.0	3.4	14
128	distribution/login/login	0.0	100.0	2.9	12
137	distribution/address/address	0.0	100.0	2.4	10
229	email/reporting/report	0.0	100.0	2.4	10
51	order/orderbillingerror	0.0	100.0	2.2	9
166	distribution/info/showtemplates	0.0	100.0	2.2	9
2	login/checklogin	0.0	100.0	1.4	6
94	postcard/order/orderthankyou	0.0	100.0	1.4	6
124	info/brief/info_service_prz	0.0	100.0	1.4	6
88	postcard/order/ordererror	0.0	100.0	1.2	5
126	distribution/index	0.0	100.0	1.2	5
227	email/order/orderthankyou	0.0	100.0	1.2	5
241	postcard/login/checklogin	0.0	100.0	1.2	5
47	order/order	0.0	100.0	0.9	4
112	email/mailing/emailsortchoice	0.0	100.0	0.9	4
119	letter/beispiel_brief	0.0	100.0	0.9	4
125	letter/login/checklogin	0.0	100.0	0.9	4
139	distribution/address/previewie	0.0	100.0	0.9	4
140	distribution/order/sendoptions	0.0	100.0	0.9	4
143	distribution/info/help	0.0	100.0	0.9	4
163	distribution/order/orderthankyou	0.0	100.0	0.9	4
196	info/info_kontakt	0.0	100.0	0.9	4
132	distribution/project/projectwork	0.0	100.0	0.7	3
159	distribution/order/ordershow	0.0	100.0	0.7	3
177	distribution/address/addressfieldcontrol	0.0	100.0	0.7	3
213	email/address/addressfieldallocation	0.0	100.0	0.7	3

Abb. 3.4.6: Ausschnitt aus der Abbruchstatistik(#Sessions pro Abbruchseite)

3.4.4 Verweilstatistik

Number of Users: 45
Number of Websites: 244

Index	Website Name	Length-Of-Stay[in %]		
		min	max	avg
1	portal_content_start	0.0	3979	735.5
2	login/checklogin	0.0	2229	642.3
163	distribution/order/orderthankyou	0.0	5376	520.6
239	email/login/checklogin	0.0	1653	453.0
179	distribution/order/ordererror	0.0	362	206.0
144	distribution/info/datenschutz	0.0	915	153.5
183	distribution/info/addressfieldcheck	0.0	165	149.3
187	address/addressuploaderror	0.0	159	147.0
106	order/orderdetailshow	0.0	154	141.2
39	billingserver/billingserver	0.0	728	120.7
182	distribution/reporting/reporting	0.0	214	109.5
186	distribution/adrmngt/new	0.0	146	104.1
193	handbuch/postkarte/adressen	0.0	105	83.7
205	info/brief/vier_schritte_brief	0.0	112	81.0
173	distribution/address/csvdownload	0.0	126	82.5
139	distribution/address/previewie	0.0	244	70.1
85	postcard/postcard/postcarderror	0.0	92	69.5
117	handbuch/brief/auftrag/druckoptionen/digital_offset	0.0	131	69.0
111	email/address/address	0.0	604	59.4
128	distribution/login/login	0.0	2009	53.2
159	distribution/order/ordershow	0.0	336	52.7
33	mailing/pdfpreviewie	0.0	59	49.3
216	email/mailing/emailupload	0.0	67	49.2
137	distribution/address/address	0.0	1099	48.3
135	distribution/sender/agencieschoice	0.0	1898	47.8
181	disribution/info/sofunk	0.0	88	46.5
146	distribution/buyaddress/selection	0.0	677	45.2
11	address/addressupload	0.0	174	44.0
231	email/mailing/wizardtext	0.0	55	43.5
13	address/addressfieldallocation	0.0	125	40.2
43	order/orderthankyou	0.0	122	40.1

34	order/mailingprintoptions	0.0	239	35.7
113	email/mailing/onlinetext	0.0	258	32.5
184	distribution/info/info_service_agb	0.0	36	32.3
119	letter/beispiel_brief	0.0	246	30.2
140	distribution/order/sendoptions	0.0	394	30.1
53	order/billingfehlerauthorisierung	0.0	46	30.0
236	mailing/mailinguploaderror	0.0	44	30.0
189	billingserver/billingservercheckssave	0.0	67	29.6
132	distribution/project/projectwork	0.0	2178	28.5
152	distribution/address/addressfieldallocation	0.0	220	27.6
116	email/mailing/inputerror	0.0	143	26.3
99	postcard/buyaddress/pickupstatus	0.0	46	26.0
167	distribution/info/sofunk	0.0	187	25.6
133	distribution/project/templatechoice	0.0	556	25.3
124	info/brief/info_service_prz	0.0	88	24.0
195	handbuch/email/projekte	0.0	74	24.0
130	distribution/project/projectwelcome	0.0	559	23.8
240	postcard/beispiel_postcard	0.0	43	23.1
29	mailing/mailingsalutationoptions	0.0	91	22.4
166	distribution/info/showtemplates	0.0	160	22.0
82	postcard/postcard/postcardeditor	0.0	21	19.5
123	info/brief/preise_brief	0.0	30	19.0
201	buyaddress/selectnone	0.0	33	19.0
169	distribution/info/kontakt	0.0	191	18.6
168	distribution/info/handbuch	0.0	102	18.6
188	billingserver/billingservercheck	0.0	45	18.5
228	email/reporting/reportings	0.0	35	18.5
48	handbuch/handbuchindex	0.0	47	18.2
191	handbuch/support	0.0	34	18.1
203	buyaddress/selectsectorshow	0.0	28	18.0
238	email/beispiel_email	0.0	26	18.0
218	email/order/auftragoption	0.0	63	17.7
115	email/mailing/previewfeedback	0.0	87	17.6
4	project/chooseproduct	0.0	470	17.5
44	project/projectselect	0.0	91	17.3

Abb. 3.4.7: Verweilstatistik (Fort.)

Die Verweilstatistik zeigt, wie lange Besucher einzelne Seiten im Schnitt “anschauen“. Es gibt 139 Seiten mit der durchschnittlichen Verweildauer über 5 Sekunden (s. Abb. 3.4.7). Dabei variieren die maximalen Verweilzeiten bis zu 5376 s bei «*distribution/order/orderthankyou*» (ca. 1,5 St.). In diesem Fall hat der Besucher nach der Bestellung nicht ausgeloggt und erst nach 5376 s eine andere Seite angeschaut, die Session hat auf ihn sozusagen “gewartet“. Diese langen Verweilzeiten deuten darauf hin, dass die Sitzung an diesen Stellen aufgehalten wurde; man kann im Allgemeinen nicht sagen, dass diese Seiten den Kunden richtig interessiert haben, wahrscheinlicher ist, dass der User sich mit anderen Tätigkeiten beschäftigt war. Man kann aus dieser Statistik ablesen, an welchen Seiten verschiedene Objekte “zu lange“ geladen werden und wie die Kunden die Seiten mit wichtigem Inhalt (wie z.B. mit Hilfsinformationen oder AGB) anschauen, d.h. ob sie diese nur flüchtig betrachten oder richtig durchlesen. Auf den Seiten mit der langen Verweilzeit haben Benutzer möglicherweise verschiedene Pop-Up-Fenster aufgerufen, die in dieser Statistik nicht berücksichtigt wurden, so kann das Kundenverhalten unterschiedlich interpretiert werden, was nicht immer den Tatsachen entspricht. Insbesondere muss die Besuchhäufigkeit einbezogen werden, weil selten genutzte Seiten bzw. Seiten ohne wichtigen Inhalt mit der langen Verweildauer keine Problemstellen aufweisen. Dagegen können häufige Seiten mit der langen Verweildauer auf Schwierigkeiten bei der Bestellung deuten.

Die Statistiken geben verschiedene Informationen über Nutzung der Website wieder. Die Besuch-/Bestellstatistik liefert zu jeder Referenzseite die Anzahl der Besucher bzw. die Anzahl der Sitzungen, wo diese Seite angeklickt wurde. Insbesondere kann der Websiteanbieter Informationen entnehmen, von wie vielen Benutzern und in wie vielen Sitzungen bestimmte Produkte bestellt wurden. Außerdem kann der Anbieter diese Zahlen mit der Anzahl der Interessierten vergleichen und Anhaltspunkte gewinnen, wie groß die potentielle und tatsächliche Kundschaften sind. Mit der Startseiten-Statistik stellt man fest, von welchen Seiten die Anwendung häufig gestartet wird, man sieht aber nicht, an welchen Websites Benutzer auf die angebotenen Produkte gestoßen haben. Die Abbruchstatistik gibt Hinweise, wo User auf Schwierigkeiten bei der Bearbeitung einer Bestellung stoßen. Durch die Analyse verschiedener statistischen Daten kann der Anbieter solche Seiten finden, die eventuell inhaltlich überarbeitet werden müssen, z. B. Seiten mit einer ungewöhnlich hohen durchschnittlichen Verweilzeit oder häufige Abbruchseiten.

Im Allgemeinen kann man sagen, dass die Statistiken sehr wenige konkrete Informationen über das Kundenverhalten liefern. Man kann nur sehr wenige Merkmale aus den Statistiken ablesen und somit ist es

sehr schwierig, ein Bild über das Kundenverhalten oder über Benutzermerkmale zu verschaffen.

Zur Analyse des Benutzerverhaltens oder zur Entdeckung von Benutzereigenschaften bzw. -präferenzen sind solche Statistiken nicht geeignet.

Anhand dieser Statistiken kann man feststellen, ob gewählte Marketingstrategien einen Erfolg versprechen oder weitere Verbesserungen notwendig sind bzw. ganz andere Mittel herangezogen werden müssen.

3.5 Gemeinsame Seitenmengen

Da die Statistiken relativ wenige Informationen über das Navigations- bzw. Nutzungsverhalten der Kunden (z.B. welche Seiten häufig zusammen aufgerufen werden) liefern und keine Schlüsse über die Benutzermerkmale erlauben, werden weitere Analyseschritte benötigt. In diesem Kapitel wird mit verschiedenen Methoden nach den Gemeinsamkeiten in den Klickfolgen gesucht, d.h. nach den gemeinsamen Seitenmengen.

3.5.1 Apriori

Als Eingabe erhält Apriori eine Datei mit dem folgenden Format:

Id:SessionId $o_1.id$ $o_2.id$ . . . $o_l.id$

wobei *SessionId* die Session-ID, $o_i.id$ die *id* des Seite-Objektes o_i sind und o_i die Referenzseite, die in der Session mit SessionId besucht wurde, so dass eine Zeile eine Session darstellt. Dabei müssen alle $o_i.id$ paarweise verschieden sein, deswegen ist diese Repräsentation keine Darstellung der Klickfolge, weil jede Seite nur einmal in der Zeile vorkommen darf. Die Ergebnismengen sind geordnete Zahlenreihen, die der Ordnung der Seiten in der Klickfolge widersprechen.

# Sessions	min. Support	# Mengen	# Supermengen
401	39	6351615	18
401	40	3189418	14
401	44	3155449	12
401	50	2113867	13

Tabelle 3.5.1: Ergebnisse von Apriori

Apriori berechnet alle Teilmengen mit dem gegebenen minimalen Support. Die Größe der Ergebnismenge wächst exponentiell, z.B. bei 244 Seiten und 401 Sessions wurden die in der Tabelle 3.5.1 vorgestellten Ergebnisse geliefert.

Man sieht, dass die Anzahl der Mengen sehr groß ist (s. Tabelle 3.5.1), wobei der Anteil der redundanten Teilmengen fast 100% ausmacht, was das Interpretieren zu einem unlösbaren Problem macht. Man kann aber aus dieser Menge die Supermengen (häufige maximale Itemmengen) rausfiltern (4. Spalte). So wird das Interpretieren wesentlich erleichtert, man sieht aber in diesem Fall die größten Seitenmengen, die den minimalen Support nicht unterbieten. Es können z.B. mehrere Supermengen gefunden werden, die die gleiche Bestellseite enthalten. In diesem Fall wird dem Anbieter sichtbar, dass User verschiedene Seiten nutzen, um ihre Bestellung abzugeben. Kommt dagegen in den Supermengen jeweils nur eine Bestellseite, kann der Anbieter zu dem falschen Schluss kommen, dass alle bzw. die meisten Besucher den gleichen Pfad nutzen bzw. auf die gleiche Weise bestellen. Das Kriterium der Häufigkeit spielt eine große Rolle bei solchen Verallgemeinerungen. Es bezieht sich auf die Anzahl der Sessions und wenn es klein gewählt wird, kann man den Fehler vernachlässigen. Bei immer größer gewählten Häufigkeiten wird die Anzahl der unter dieser Grenze liegenden und deswegen nicht gefunden Mengen immer größer, so steigt die Wahrscheinlichkeit, dass der Anbieter nur die meist besuchten Pfade entdeckt und sie als von den meisten Besuchern genutzt identifiziert. Das andere Problem bei den häufigen maximalen Itemmengen ist, dass es nicht immer aus einer Supermenge eine Klickfolge rekonstruiert werden kann, weil die Supermenge nicht unbedingt alle Items einer Klickfolge enthält, wie es im Fall der 4. Supermenge aus Abbildung 3.5.1 ist. (In Klammern steht die absolute Häufigkeit der entsprechenden Menge.) Hier enthält die Supermenge Items, die den Beginn und das Ende der Klickfolgen darstellen, die dazwischen liegenden Items fehlen. Solche Informationen sind kaum nützlich bei der Analyse der Klickpfade. Andererseits decken lange Supermengen Teilpfade, die eventuell viel häufiger auftreten als die Supermenge selbst. Diese Information geht ganz verloren bei der Analyse der Supermengen.

52 107 108 110 111 112 113 114 208 213 214 218 219 220 221 222 223 224 225 226 ***227*** (42)
107 108 110 111 112 113 114 207 208 213 214 218 219 220 221 222 223 224 225 226 ***227*** (44)
107 108 110 111 112 113 114 208 213 214 218 219 220 221 222 223 224 225 226 ***227*** 228 229 (39)
3 4 6 7 9 10 36 37 38 39 40 42 ***43*** (52)

Abb. 3.5.1: Beispiele der Supermengen

Unter 18 Supermengen mit dem minimalen Support 39 sind 4 Seitenmengen mit Bestellseiten, 3 Mengen enthalten Bestellseite *227* und eine enthält Bestellseite *43*. Und unter 14 bzw. 12 Supermengen mit minimalen Supports 40 bzw. 44 sind 3 (2 für Bestellseite *227* und 1 für *43*) von 4 Bestellseiten (für 4 verschiedene Produkte) vorhanden. Die ersten 2 Supermengen sind fast identisch: die erste Menge enthält die Seite 52, die zweite – Seite 207, alle anderen Seiten haben sie gemeinsam. Man kann nur sagen, dass es in 42 Sessions alle Items aus der 1. Supermenge vorhanden sind (alle aufgezählte Seiten angeklickt wurden) und entsprechend in 44 Sessions – alle Items aus der 2. Supermenge. Die 3. Supermenge enthält alle Items aus der 2. außer des Items 207 und 2 weitere Items 228 und 229. Man kann aus dieser Information nicht entnehmen, ob es mindestens 125(=44+42+39) Sessions mit Bestellseite 227 gibt oder die entsprechenden Sessions sich überschneiden und es weniger als 125 Sessions mit Bestellseite *227* gibt.

Man könnte denken, dass man bei klein gewähltem Support alle nützlichen Informationen findet, aber das Hauptproblem dabei ist enormer Speicherplatzbedarf, so dass die Methode zur Ermittlung der gemeinsamen Mengen eher ungeeignet erscheint.

3.5.2 WINEPI

Eine weitere Methode zur Entdeckung der gemeinsamen Items ist *WINEPI*-Algorithmus. Als Eingabe erhält *WINEPI* eine Datei des gleichen Formats wie *Apriori*, hier wird aber nicht gefordert, dass alle Seiten innerhalb einer Session verschieden sein müssen. Ferner kann man hier eingeben, ob die gemeinsamen Mengen injektiv sein dürfen (d.h. jede Seite nur einmal vorkommen darf) oder nicht. Beim gleichen minimalen Support mit Forderung, dass nur injektive Mengen betrachtet werden, liefert *WINEPI* ähnliche Ergebnismenge wie *Apriori*. Die häufigen Seitenmengen spiegeln die Klickfolge ohne wiederholte Seiten wider. Da das Verfahren alle häufigen Reihenfolgen betrachtet, kommen beispielsweise sowohl (*a*, *b*) als auch (*b*, *a*) in der Ergebnismenge vor, deswegen ist die Ergebnismenge noch größer als bei *Apriori*. Wählt man als Eingabe alle Seitenmengen (d.h. auch nicht injektive) kommt man zu einer wesentlich größeren Ergebnismenge als bei der injektiven Version.

Diese Methode erfordert noch mehr Speicherplatz als *Apriori*, deswegen sollte man die Verfahren nutzen, die Seitenmengen finden und das beschriebene Speicherplatzproblem umgehen. So bieten sich als Alternativen die Verfahren *Close-FCI* und *IPM2* an.

3.5.3 Frequent Closed Sequence Itemsets

Mit dieser Methode findet man die häufigen abgeschlossenen Itemmengen, d.h. für häufige Items und für einige Itemmengen werden Schnittmengen der Transaktionen berechnet, in denen dieses Item bzw. Itemmengen vorkommen. Dabei werden Itemmengen nicht lexikographisch (wie bei *Apriori*), sondern nach der zeitlichen Ordnung der Items in der Menge, d.h. nach der Zeit des Eventauftretens, geordnet. Die Kandidaten von Itemmengen werden mit dem Algorithmus *WINEPI* berechnet. Wie bei *WINEPI* sind verschiedene Datenformate als Eingabe möglich. Bei der Suche nach den häufigen Itemmengen wurden Klickfolgen ohne Wiederholungen analysiert.

# Sessions	min. Support	# Mengen	# Mengen mit Bestellseiten
401	50	52	3
401	44	69	4
401	40	78	4
401	39	84	5
401	35	100	9
401	25	155	18
401	20	211	27
401	15	272	42
401	10	327	52
401	5	418	68

Tabelle 3.5.2: häufige abgeschlossene Itemmengen

Die Menge der häufigen Seitenmengen enthält zwar auch redundante Mengen, ist aber wesentlich kleiner (auch beim kleinen minimalen Support) als bei *Apriori* (vgl. Tabellen 3.5.1 und 3.5.2) oder *WINEPI*, was das Auffinden der gemeinsamen Charakteristiken bedeutend erleichtert. In der 3. Spalte (s. Tabelle 3.5.2) steht die Anzahl der gefundenen Itemmengen und in der 4. Spalte die Anzahl der Mengen mit Bestellseiten. Z.B. von 52 gefundenen Itemmengen (1. Zeile, Support = 50) sind 3 Mengen mit Bestellseiten.

3.5.3.1 Charakteristika der Besteller und Nichtbesteller

Da die Itemmengen als Episoden bei *WINEPI* generiert wurden, treten die einzelnen Items in Itemmengen in der Reihenfolge auf, in der entsprechende Seiten angeklickt wurden. So sind unter den Itemmengen sowohl die Mengen, die die kompletten Seitenfolgen darstellen, die alle während einer Session angeklickt wurden, als auch ihre Teilmengen, d.h.

lückenhafte Klickfolgen. Diese Itemmengen werden in 2 Mengen aufgeteilt: eine Menge mit Itemmengen mit den Bestellseiten und zweite ohne sie. Zusätzlich werden aus der zweiten Menge die Itemmengen ausgesondert, die Teilmengen der Itemmengen aus der ersten Menge sind. Die erste Menge enthält dann die Auflistung der Charakteristiken der Besteller und zweite die der Nichtbesteller. Mit anderen Worten, enthält die erste Liste Seitenmengen, die Besteller anklicken und zweite Seitenmengen, die Nichtbesteller anklicken. Die entdeckten Itemmengen stellen Entscheidungskriterien einer Qualifikationsfunktion dar, mit deren Hilfe die Kundengemeinschaft (*Population*) in *Subgruppen* eingeteilt werden kann (s.[29]). Die Beispiele dieser Mengen sind in Abbildungen 3.5.2 und 3.5.3 zu sehen. Der Schnitt der Itemmengen aus diesen 2 Mengen ist nicht leer, sie haben viele Items gemeinsam. Entscheidend für die Charakteristiken von Nichtbesteller ist das Vorhandensein einiger bestimmten Items, die in keiner Itemmenge vorkommen, die Charakteristiken der Besteller darstellen. Unter diesen Mengen sind sehr viele Itemmengen aufzufinden, wo einzelne relativ selten vorkommende Items, fehlen. 21 von 68 Mengen mit Bestellseiten entsprechen den genutzten "Klickfolgen"[6] wie Folgen 2,6,7,8,10 und 13 aus Abbildung 3.5.2, andere sind "Teilfolgen" oder "Folgen" mit "Lücken".

1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 29 30 31 34 35 36 37 38 39 40 42 ***43*** (32)
2	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42 ***43*** (11)
3	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 29 30 31 32 33 34 35 36 37 38 39 40 42 ***43*** (19)
4	3 4 6 7 9 10 36 37 38 39 40 41 42 ***43*** (33)
5	3 4 6 7 9 10 36 37 38 39 40 42 ***43*** (52)
6	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42 ***43*** 44 45 (5)
7	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 42 ***43*** (21)
8	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 236 29 30 31 32 33 34 35 36 37 47 38 39 40 41 42 ***43*** 44 (7)
9	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (10)
10	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (9)
11	3 4 120 190 122 191 192 193 194 195 196 6 7 9 10 36 37 38 39 40 41 42 ***43*** (8
12	3 4 6 7 9 10 36 37 38 39 40 41 42 ***43*** 52 (15)
13	3 4 6 7 44 9 10 47 36 37 38 39 40 41 42 ***43*** (10)

[6] Klickfolgen, Pfade ohne wiederholte Seiten

14	55 55 56 58 59 87 89 90 91 92 93 94 (21)56 58 59 87 89 90 91 92 93 ***94*** (21)
15	3 4 55 56 86 58 59 87 89 90 91 92 93 94 60 69 57 61 95 96 (7)
16	3 4 55 56 86 58 59 87 89 90 91 92 93 ***94*** 60 69 57 61 95 96 63 64 65 66 67 68 97 98 99 100 101 78 79 80 81 82 83 84 85 (5)
17	55 56 57 58 59 71 72 73 74 75 76 77 78 79 80 81 82 102 83 242 243 87 89 90 91 92 244 93 ***94*** (14)
18	55 56 57 58 59 71 72 73 74 75 76 77 78 79 80 81 82 102 83 242 243 87 89 90 91 92 244 93 ***94*** 52 (8)

19	126 130 132 133 140 157 158 159 160 161 162 ***163*** (27)
20	126 130 131 140 157 158 159 160 161 162 ***163*** (14)
21	126 130 131 132 133 135 136 137 149 150 151 152 153 154 177 155 156 140 157 158 159 160 161 162 ***163*** 141 128 167 166 142 143 169 168 129 (5)
22	126 130 132 133 140 157 158 159 160 161 162 ***163*** 134 (10)
23	126 130 132 133 135 136 137 140 157 158 159 160 161 162 ***163*** 168 (9)
24	126 130 145 132 133 135 136 137 146 147 148 140 157 158 159 160 161 162 ***163*** 134 (9)
25	126 130 132 133 135 136 137 140 157 158 159 160 161 162 ***163*** (26)
26	126 130 145 132 133 135 136 137 146 147 148 138 139 140 157 158 159 160 161 162 ***163*** 134 (6)
27	126 130 145 132 133 135 136 137 138 139 140 157 158 159 160 161 162 ***163*** (

28	107 108 110 111 208 213 214 112 216 217 222 223 224 225 226 ***227*** (38)
29	107 108 110 111 208 213 214 112 113 114 218 219 220 221 222 223 224 225 226 ***227*** (68)
30	107 108 110 111 208 213 214 112 113 114 218 219 220 221 222 223 224 225 226 ***227*** 229 (39)
31	3 4 107 108 207 110 111 208 213 214 112 113 114 218 219 220 221 222 223 224 225 226 ***227*** 228 237 229 (12)
32	3 107 108 110 111 208 213 214 112 215 216 217 222 223 224 225 226 ***227*** (15)
33	3 4 107 108 110 111 208 213 214 112 215 216 217 222 223 224 225 226 ***227*** (13)
34	3 4 107 108 109 110 111 208 213 214 112 215 216 217 113 114 115 218 219 220 221 222 223 224 225 226 ***227*** (10)
35	107 108 109 110 111 208 213 214 112 113 114 218 219 220 221 222 223 224 225 226 ***227*** 229 (14)
36	107 108 207 110 111 208 213 214 112 113 114 116 218 219 220 221 215 230 231 232 233 216 217 222 223 224 225 226 ***227*** 228 229 (13)
37	3 4 107 108 109 110 111 208 213 214 112 113 114 218 219 220 221 222 223 224 225 226 ***227*** 228 229 (8)

Abb. 3.5.2: Beispiele der gemeinsamen Seitenmengen mit minimalem Support 5

So sieht der Anbieter, nachdem die Itemmengen in eine geordnete Liste der Referenzseiten umgewandelt werden, die so genannten Haupt(teil)pfade der Besteller und Nichtbesteller, d.h. “Pfade“, in denen seltene Seiten nicht vorkommen. Der Anbieter sieht außerdem, welche (Haupt)Pfade wie oft genutzt wurden. Man sieht z.B., dass die 5. Menge (s. Abb. 3.5.2), die kürzeste Menge mit Bestellseite *43*, 52 Mal vorkommt. (Zufälligerweise ist diese Menge die abgeschlossene Menge des Items *43*, d.h. die kürzeste abgeschlossene Menge ist nicht notwendigerweise die abgeschlossene Menge eines uns interessierten Items, es kann eine Itemmenge geben, deren abgeschlossene Menge noch kürzer ist.) Genau so viele Male wurde das Produkt A bestellt. Das bedeutet, dass die Seiten aus dieser Menge immer bei der Bestellung angeklickt wurden. Diese Seiten kann man vermutlich bei der Bestellung auf keinen Fall umgehen.

Mit dieser Methode erhält man für jede Bestellseite jeweils das gleiche (abgeschlossene) Set, unabhängig davon, wie klein/groß der minimale Support gewählt wurde (vorausgesetzt er ist nicht größer als die tatsächliche Häufigkeit). Es ist unrealistisch zu erwarten, dass die kürzeste abgeschlossene Itemmenge einer kompletten Klickfolge entspricht, das wird auch an den praktischen Beispielen sichtbar. Der Webdesigner sollte nach der Entdeckung dieser Itemmengen die möglichen Folgen so optimieren, dass sich die Menge der notwendigen Seiten an das kürzeste abgeschlossene Itemset nähert, d.h. den Hauptpfad einer Bestellung so gestalten, dass er möglichst der abgeschlossenen Itemmenge der Bestellseite entspricht und die Anzahl der Zwischenseiten, die in dieser Menge nicht enthalten sind, reduziert wird. Bei mehreren möglichen verzweigten Bestellungspfaden erscheint dieser Weg zur Optimierung sehr schwierig zu sein, eine andere Alternative zur Optimierung der Website bietet sich nach der Entdeckung der Nutzungsmodelle an.

Besonders interessant sind Mengen, die keine Teilmengen der Itemmengen mit Bestell-URL sind, Es wurden insgesamt 84 solche Seitenmengen (mit Kardinalität größer 3) gefunden. Diese sind Charakteristika der Nichtbesteller (s. Abb. 3.5.3). Das bedeutet, dass in weniger als in 5 Fällen (bzw. kein einziges Mal) nach dem Besuch dieser Seiten bestellt wurde. Durch die Analyse dieser Charakteristika geht der Anbieter dem Grund nach, warum Benutzer ihre Bestellungen abbrechen. Es ist zu beachten, dass diese Seitenmengen nicht unbedingt möglichen Klickfolgen entsprechen, so dass das Interpretieren mancher Mengen kaum möglich wird. Mit steigender Anzahl der Items in der Menge steigt auch die Wahrscheinlichkeit, dass diese Itemmenge sich zu einer (fast) vollständigen Klickfolge rekonstruieren lässt, aber das Auffinden einzelner problematischen Seiten wird schwieriger. Durch den direkten Vergleich mit den Itemmengen, die Charakteristika der

Besteller sind, findet man einzelne Items bzw. Itemmengen, die speziell für Nichtbesteller repräsentativ sind, d.h. in keiner Bestellfolge vorkommen. Aus 84 Seitenmengen konnte man 23 Seiten bzw. Seitenmengen aussondern, die in keiner (bzw. weniger als in 5) Bestellfolge(n) erschienen sind (s. Abb.3.5.4). Manche von ihnen sind direkte Hinweise auf die Seiten, wieso die Bestellung nicht stattgefunden hat. Andererseits kann die "Ursache" nicht nur an solchen "auffälligen" Seiten liegen, die Bestellung könnte beispielsweise wegen einiger bestimmten nacheinander folgenden Seiten scheitern. Mit dieser Methode kann man dies nicht in jedem Fall feststellen. Dazu sollte man Ergebnisse dieser Methode mit Ergebnissen der Methode kombinieren, die sequentielle Abhängigkeiten findet, z. B. der *IPM2*-Methode.

1	3 4 48 (5)
2	3 4 6 7 9 10 11 12 13 14 15 50 16 17 18 (11)
3	3 4 6 7 9 10 36 37 38 39 40 51 (10)
4	3 4 55 56 57 58 59 60 61 62 63 70 (8)
5	3 4 55 56 86 58 59 87 88 (5)
6	3 4 6 7 9 10 197 198 199 200 201 1 119 52 (12)
7	3 4 6 7 9 10 197 198 202 203 1 119 52 (12)
8	3 4 6 7 8 9 10 11 197 198 199 200 201 202 203 204 1 119 205 206 52 (6)
9	3 4 55 56 57 58 59 60 61 62 63 209 70 (7)
10	3 4 107 108 207 110 111 208 213 214 112 215 230 231 232 233 113 114 115 218 219 220 234 235 52 (6)
11	3 1 2 107 238 239 55 240 241 (5)
12	1 6 119 123 124 (6)
13	1 2 6 119 123 124 (5)

Abb. 3.5.3: Beispiele der abgeschlossenen Itemmengen (keine Teilmengen der Bestellmengen)

48 (5)
50 (11)
51 (10)
62 70 (8)
88 (5)
119 123 124 (10)
197 198 199 200 201 1 119 (12)
197 198 202 203 1 119 (12)
234 235 (6)
1 2 238 239 240 241 (5)

Abb. 3.5.4: Beispiele von Items, die in "keiner" Bestellmenge vorhanden sind

Mit dieser Methode lassen sich die Charakteristika der Besteller bzw. Nichtbesteller finden. Selten vorkommende Merkmale werden dabei ignoriert. Durch das Extrahieren der Items bzw. Itemmengen, die in Charakteristiken der Besteller nicht vorkommen, findet man direkt Referenzseiten, die sehr wahrscheinlich die Ursache des Nichtbestellens waren. Diese Seiten sollten auf jeden Fall überprüft und inhaltlich überarbeitet werden, insbesondere wenn es um Fehlermeldungen handelt, die eventuell keine Hilfe für Benutzer darstellen, sondern eher irreführend erscheinen.

3.5.3.2 Nutzungsmodelle

Aus den oben präsentierten häufigen Itemmengen lassen sich Nutzungsmodelle ableiten. Da einige kurze Itemmengen die Teilmengen von vielen anderen längeren sind, sollte man solche Itemmengen bei der Bildung von Nutzungsmodellen nicht berücksichtigen. Es sind insgesamt 68 abgeschlossene Itemsets mit Bestellseiten gefunden worden: 21 für Seite *43* (Produkt A), 5 für Seite *94* (Produkt B), 18 für Seite *163* (Produkt C), 24 für Seite *227* (Produkt D). Nach dem Ähnlichkeitsprinzip kann man die gefundenen Itemmengen in die Cluster aufteilen. Dabei werden einzelne Itemmengen einer Gruppe entsprechend der Entfernung zu einem Muster-Itemset (d.h. Zentrum) jeder Klasse zugeordnet. Falls ein Objekt zwei Clustern zugeordnet werden könnte (d.h. die zulässigen Abstände zu beiden Zentren nicht überschritten sind), wird es in das Cluster eingeteilt, dessen Zentrum am nächsten liegt. Das Muster-Itemset wird iterativ bestimmt, so dass Objektmenge in einem Cluster maximiert wird, d.h. wenn bei altem Zentrum ein Objekt in das Cluster nicht aufgenommen werden kann und durch Zuweisung des Zentrum-Status einem anderen Objekt des Clusters in die Gruppe eingeteilt werden kann, wird dies (die beschriebene Zuweisung) auch durchgeführt und die Abstände werden entsprechend aktualisiert. Dazu muss man definieren, welche Kriterien einzelne Objekte (Itemmengen) erfüllen müssen, um als ähnlich bewertet zu sein, bzw. wie die Qualitätsfunktion (d.h. der Abstand) berechnet wird. Man kann z.B. als einen zulässigen Abstand zweier ähnlichen Itemmengen die maximale Anzahl unterschiedlicher Items in 2 Itemmengen festlegen, sinnvoller kann aber erscheinen, den ungleichen Items eine Gewichtung beizumessen, indem man z.B. nicht die gesamte Anzahl unterschiedlicher Items, sondern ihre Anzahl am Anfang, in der Mitte und am Ende einer Itemmenge begrenzt (Items sind in der Menge geordnet). In Abbildung 3.5.5 ist ein Beispiel vorgeführt, wie man eine Itemmenge in 3 Teile unterteilen kann: die Anzahl der Items am Anfang und in der Mitte ist gleich Itemsanzahl *div* 3 (im Beispiel sind es 5) und alle übrig gebliebenen (hier 6) werden dem

Ende einer Itemmenge zugewiesen. Die Zeitabstände sind fiktiv und daher ohne Bedeutung.

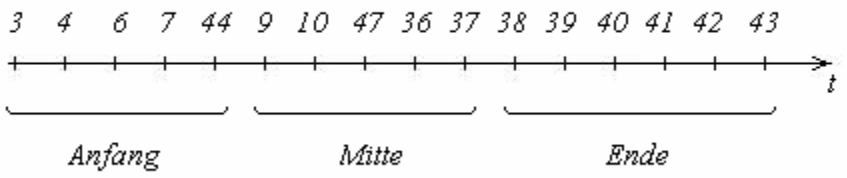


Abb. 3.5.5: Beispiel einer Itemmenge in zeitlicher Abfolge

Nachdem als solches Ähnlichkeitskriterium der Tripel (2,1,3) (2 unterschiedliche Items am Anfang, 1 Item in der Mitte und 3 am Ende der Itemmenge) gewählt war, konnte man 9 Nutzungsmuster (A bis I) beim Bestellen von Produkt A, jeweils 5 Nutzungsmuster – von Produkten C und D und 2 Muster – von Produkt B finden. Die 9 Modelle mit dem Produkt A würden sich schon mit dem Kriterium (0,0,2) ergeben (s. Abb. 3.5.6). Wenn man als Kriterium den Tripel (0,2,6) wählt, entdeckt man 6 Nutzungsmodelle beim Bestellen von Produkt A: Modell (A) mit Zentrum A.2, Modell (B+C) mit Zentrum C.1, Modell (D) , Modell (E) mit Zentrum E.1, Modell (F+G) mit Zentrum G.1 und Modell (H+I) mit Zentrum H1 .

Dabei ist es zu beachten, dass nicht alle Nutzungsmodelle "echt" sind (z.B. Modelle B.1 bis E.2 sind nicht echt), da sie keiner vollständigen Klickfolge entsprechen. Nur 21 von 68 Itemsets entsprechen den vollständigen Klickfolgen: 9 mit Seite *43*, 4 mit Seite *94*, 1 mit Seite *163*, 7 mit Seite *227*. Die Modelle B.1 bis E.2 sind Schnittmengen der anderen Muster. Auch die möglichen kompletten Klickfolgen können auch Schnittmengen von anderen Pfaden sein.

Bei der Analyse von Nutzungsmodellen ist außerdem nützlich zu wissen, wie häufig während einer Session einzelne Seiten angeklickt wurden und wie lange Benutzer auf ihnen verweilen. Deswegen ist es zweckmäßig, für jede Seite in einem Nutzungsmodell diese Daten zu präsentieren. In Abbildung 3.5.7 sind Modelle A.1 und A.2 aus Abbildung 3.5.6 mit diesen Informationen zu sehen. Der Ausdruck x ($a = b * c$) bedeutet folgendes: die Benutzer haben durchschnittlich a Sekunden auf der Seite x während einer Session verweilt, indem sie durchschnittlich b Mal auf die Seite x angeklickt haben (bzw. weitergeleitet wurden) und jedes Mal durchschnittlich c Sekunden jedes Mal auf ihr verweilt haben. Auffällig ist es, dass jede Seite in diesen 2 Nutzungsmodellen durchschnittlich 1 Mal während der Sitzung besucht wurde. Dies ist ein gutes Zeichen

A.1	3 4 6 7 44 9 10 47 36 37 38 39 40 41 42 ***43*** (10)
A.2	3 4 6 7 44 9 10 47 36 37 38 39 40 42 ***43*** (15)
B.1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 29 30 31 32 33 34 35 36 37 38 39 40 42 ***43*** (19)
B.2	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** 52 (8)
C.1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 29 30 31 34 35 36 37 38 39 40 42 ***43*** (32)
C.2	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 29 30 31 34 35 36 37 38 39 40 42 ***43*** 52 (14)
D.1	3 4 120 190 122 191 192 193 194 195 196 6 7 9 10 36 37 38 39 40 41 42 ***43*** (8)
E.1	3 4 120 190 122 191 192 193 194 195 196 6 7 8 9 10 11 12 13 14 15 16 17 18 26 29 30 31 34 35 36 37 38 39 40 41 42 ***43*** (6)
E.2	3 4 120 190 122 191 192 193 194 195 196 6 7 8 9 10 11 12 13 14 15 16 17 18 26 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (5)
F.1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (10)
G.1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42 ***43*** (11)
G.2	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42 ***43*** 44 45 (5)
G.3	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (9)
H.1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 42 ***43*** (21)
H.2	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 42 ***43*** 52 (13)
I.1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 236 29 30 31 32 33 34 35 36 37 47 38 39 40 41 42 ***43*** 44 (7)
I.2	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 236 29 30 31 32 33 34 35 36 37 47 38 39 40 41 42 ***43*** 44 52 (6)

Abb. 3.5.6: Nutzungsmodelle beim Bestellen des Produktes A

dafür, dass Benutzer erwartungsgemäß durch die Website navigieren. Das gilt allerdings nicht für alle Nutzungsmuster. Im Weiteren wird gezeigt, dass auch Modelle entdeckt wurden, wo einzelne Seiten mehrfach während einer Session angeschaut wurden. Ausgehend von dieser Darstellung kann der Analyst zum Schluss kommen, dass bei Zuordnung der Seitenmengen in Nutzungsmodelle auch weitere Ähnlichkeitskriterien wie durchschnittliche

Anzahl der Besuche der entsprechenden Seite in einer Session und durchschnittliche Verweildauer auf dieser Seite berücksichtigt werden sollten. Diese sollten jedoch nicht zu fein gewählt werden, um die Einteilung der *n* Seitenmengen in *n* Modelle zu vermeiden.

3(1=1*1) 4(18=1*18) 6(0=1*0) 7(8=1*8) 44(5=1*5) 9(3=1*3) 10(2=1*2) 47(2=1*2) 36(5=1*5) 37(20=1*20) 38(0=1*0) 39(83=1*83) 40(2=1*2) 41(20=1*20) 42(0=1*0) 43(105=1*105) [10]
3(1=1*1) 4(15=1*15) 6(0=1*0) 7(8=1*8) 44(4=1*4) 9(3=1*3) 10(2=1*2) 47(2=1*2) 36(5=1*5) 37(18=1*18) 38(0=1*0) 39(67=1*67) 40(2=1*2) 42(0=1*0) 43(83=1*83) [15]

Abb. 3.5.7: Nutzungsmodelle mit Besuchzahl und Verweildauer

Für die Analyse des Nutzungsverhaltens der Nichtbesteller (d.h. der Seitenmengen, die in den Bestellmengen nicht vorkommen), ist es besonders wichtig zu wissen, wie häufig und wie lange einzelne Seiten angeschaut wurden. Die Abbildung 3.5.8 gibt eine Übersicht einiger häufigen Seitenmengen (bzw. Nutzungsmodellen) der Nichtbesteller mit Verweilzeiten. Die Seiten mit der großen durchschnittlichen Verweildauer oder mit der überdurchschnittlich großen Zahl der Besuche (z.B. 3 oder mehr) sind wahrscheinlich diejenigen, die Probleme beim Bestellen verursachen. Es fällt auf, dass die Seiten 1, 2 und 239 in der Zeile 5 sehr lange Verweilzeiten aufweisen und Seiten 198 (Zeile 3) und 82 (Zeile 6) unerwartet oft angeschaut wurden. Solche Seiten müssen zu

3(1=1*1) 4(6=1*6) 6(0=1*0) 7(18=1*18) 9(1=1*1) 10(4=1*4) 11(10=1*10) 12(6=1*6) 13(6=1*6) 14(0=1*0) 15(3=1*3) 50(4=1*4) 16(3=1*3) 17(0=1*0) 18(3=1*3) [9]
3(0=1*0) 4(43=1*43) 6(0=2*0) 7(18=2*9) 9(2=1*2) 10(2=1*2) 36(4=1*4) 37(7=1*7) 38(0=1*0) 39(46=1*46) 40(1=1*1) 51(6=1*6) [6]
3(3=3*1) 4(13=1*13) 6(2=2*1) 7(45=3*15) 9(1=1*1) 10(24=3*8) 197(30=3*10) 198(75=5*15) 202(0=1*0) 203(54=3*18) 1(26=2*13) 119(78=3*26) 52(0=1*0) [12]
3(1=1*1) 4(2=1*2) 55(1=1*1) 56(8=2*4) 86(10=2*5) 58(4=2*2) 59(4=2*2) 87(4=2*2) 88(0=1*0) [5]
3(0=1*0) 1(313=1*313) 2(448=1*448) 107(0=1*0) 238(18=1*18) 239(453=1*453) 55(1=1*1) 240(23=1*23) 241(0=1*0) [5]
3(1=1*1) 4(2=1*2) 55(1=1*1) 56(9=3*3) 57(8=1*8) 78(15=3*5) 79(9=3*3) 80(3=3*1) 81(6=3*2) 82(105=5*21) 102(0=1*0) [8]
3(0=1*0) 107(1=1*1) 108(7=1*7) 110(1=1*1) 111(2=1*2) 112(2=1*2) 113(168=3*56) 114(0=2*0) 116(110=2*55) [7]

Abb. 3.5.8: Nutzungsmodelle der Nichtbesteller mit Besuchzahl und Verweildauer

allererst unter die Lupe genommen werden. Dabei sind jedoch die Schlussfolgerungen von den Mittelwerten abhängig, die von der einzelnen stark abweichenden Verweildauer beträchtlich beeinflusst werden können, d.h. wenn ein Benutzer ungewöhnlich lange auf einer Seite verweilt hat, kann dies den Durchschnittswert erheblich erhöhen währenddessen die Mehrheit der Benutzer viel weniger Zeit auf der gleichen Seite verbracht hat.

Durch die Analyse der entdeckten Nutzungsmuster sieht der Anbieter, welche Pfade (auch wenn sie nicht vollständig sind) wie oft genutzt werden. An dem Beispiel der Nutzungsmodelle aus Abbildung 3.5.6 kann man 3 Hauptpfade identifizieren: A.2, G.1 und H.1. Ihre gemeinsame Häufigkeit ist 47; das Produkt A wurde 52 Mal bestellt (dies bestätigt noch mal, dass fast alle Bestellungen von Produkt A diese "Pfaden" durchgelaufen haben). Alle Nutzungsmuster von B.1 bis I.1 sind die Abweichungen von dem Pfad C.1. Bei der Optimierung der Website sollte der Anbieter solche Pfade zu Hauptfaden seiner Applikation gestalten (falls sie es noch nicht sind) und die Anzahl der Abweichungen (von diesen Pfaden) möglichst klein zu halten bzw. selten vorkommende ganz zu eliminieren oder mit anderen Seiten zusammenzuführen.

Die häufigen Seitenmengen liefern dem Analysten viele wichtige Informationen, nämlich welche Pfade werden von Bestellern und Nichtbestellern genutzt, d.h. die Nutzungsmodelle ihrer Website. Aus diesen Daten kann er ableiten, ob bestimmte Pfade bzw. Seiten einer Verbesserung oder Überprüfung unterzogen werden müssen. Dies gilt insbesondere für die Pfade der Nichtbesteller und die seltenen Seiten (die werden im Kapitel 3.6.1 näher behandelt). Der wichtige Vorteil dieser Methode ist Unempfindlichkeit gegen die Ereignisse, die selten vorkommen, sie werden durch diese Methode nicht in Betracht gezogen.

3.5.4 IPM2 (Interaction-Pattern Mining)

Mit dieser Methode findet man häufige sequentielle Muster, d.h. sich mehrfach wiederholende Sequenzen, unter den Klickfolgen.

Als Eingabe können auch hier 2 Invarianten der Klickfolgen genutzt werden:

- wo alle Seiten-Klicks registriert wurden oder
- wo keine wiederholten Seiten vorkommen.

Als Kriterium (s. Kapitel 2.7.1) c = (*minLen*, *minSupp*, *maxError*, *minScore*) wurde c = (5,5,0,0) gewählt. Indem man *maxError* als 0 eingibt, werden ausschließlich Sequenzen entdeckt, die lückenlosen (Teil)Klickfolgen

entsprechen. Wenn man die Transaktionen ohne wiederholte Seiten analysiert, werden Reihen von Seiten gefunden, die nacheinander angeklickt wurden, die aber nicht unbedingt in einem “Quelle-Ziel“-Verhältnis stehen. Diese Reihen ähneln den gefundenen abgeschlossen Itemmengen aus dem vorigen Kapitel, hier werden aber die selten vorkommenden Seiten nicht ausgeschlossen, eine (Teil)Sequenz wird in diesem Fall nicht mehr ausgeweitet und als abgeschlossen betrachtet. Um auffällige Muster im Benutzerverhalten zu finden, müssen außerdem die tatsächlichen Klickfolgen analysiert werden.

Invariante	# Sessions	min. Support	min. Länge	#Mengen =(Teil)sequenzen	# Mengen mit Bestellseiten
1	401	5	5	382	82
2	401	5	5	235	66

Tabelle 3.5.3: Ergebnisse nach dem IPM2-Algorithmus

Diese Methode liefert relativ kleine Ergebnismenge (s. Tabelle 3.5.3), die einfach zu interpretieren ist und auch zur Entdeckung von Nutzungsmodellen und zur Bildung der Benutzergruppen genutzt werden kann. In der Ergebnismenge sowohl vollständige Klickfolgen als auch Teilsequenzen vorhanden. Diese Methode bietet dem Anbieter die Möglichkeit, alle häufig genutzte Klickfolgen in einer Übersicht zu betrachten. Es hängt selbstverständlich von der Anzahl der entdeckten Muster, ob es alle den Analysten interessierten Folgen auf einen Blick zu betrachten sind. Verzichtet der Web-Designer auf die Darstellung der Teilsequenzen bzw. der seltenen Sequenzen, bleiben in der Regel nicht viele unterschiedliche Muster übrig.

3.5.4.1 Charakteristika der Besteller und Nichtbesteller

Die Sequenzen, die spezielle Bestellseiten beinhalten, können als Charakteristika der Besteller eines entsprechenden Produktes (wie auch bei der “Abgeschlossene Itemsets“-Methode) betrachtet werden (s. Abb. 3.5.9).

Um für Nichtbesteller charakteristische Folgen in der Sequenzmenge zu finden, muss man solche Reihen suchen, die in keiner Bestellsequenz enthalten sind. Es gibt insgesamt 103 solche (Teil)Pfade mit der minimalen Länge 3, die mindestens 5 Mal vorgekommen sind. Beispiele sind in der Abbildung 3.5.10 zu sehen. Es ist allerdings schwierig zu beurteilen, welche Seite die Ursache des Phänomens ist, dass es nach dem Besuch dieser Seitenfolgen nicht bestellt wurde. Als Erstes fällt es auf, dass in keiner Bestellsequenz die Seite “50“ enthalten ist. Es kann

auch sein, dass Benutzern entweder einige wichtige Informationen auf Seiten “18“, “34“, “35“, “37“, “40“ fehlen und sie deshalb in Schwierigkeiten bei Auftragsbearbeitung geraten. Der Anbieter sollte solche Sequenzen näher untersuchen um festzustellen, ob auf diesen

3 4 6 7 8 9 10 11 12 13 11 12 13 11 12 13 14 15 16 17 18 26 210 211 212 236 212 29 30 31 32 33 34 35 36 37 47 34 35 36 37 38 39 40 41 42 ***43*** 44 3 4 107 108 207 110 111 208 213 111 208 213 111 208 213 214 112 113 114 218 219 220 221 222 223 224 225 226 ***227*** 228 237 229 52
3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 236 29 30 31 32 33 34 35 36 37 47 38 39 40 41 42 ***43*** 44 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** 3 4 6 7 44 9 10 47 36 37 38 39 40 41 42 ***43*** 3 4 107 108 207 110 111 208 213 214 112 113 114 218 219 220 221 222 223 224 225 226 ***227*** 228 237 229 52

Abb. 3.5.9: Beispiele von sequentiellen Mustern mit (in oberer Zeile) und ohne Wiederholungen (in unterer Zeile)

Seiten wichtige Informationen fehlen oder es z.B. an der Teilsequenz “18 19“ oder an der Seite “50“ liegt, dass es zur Bestellung nicht kommt. Die Tatsache ist, dass es über Seiten “18 26“ mehrfach bestellt wurde und über “18 19“ entweder kein oder weniger als 5 Mal.

3 4 6 7 8 9 10 11 12 13 14 15 50 16 17 18 19 20 21 22 23 24 25 27 29 30 31 34 35 36 37 38 39 40 (5)
3 4 6 7 8 9 10 11 12 13 14 15 50 16 17 18 19 20 21 22 23 24 25 27 29 30 31 34 35 36 37 (6)
3 4 6 7 8 9 10 11 12 13 14 15 50 16 17 18 19 20 21 22 23 24 25 27 29 30 31 34 35 (7)
3 4 6 7 8 9 10 11 12 13 14 15 50 16 17 18 19 20 21 22 23 24 25 27 29 30 31 34 (8)
3 4 6 7 44 9 10 197 198 202 203 199 200 201 1 119 123 52 (6)
3 4 6 7 44 9 10 47 36 37 38 39 188 (6)
11 12 13 14 15 50 16 17 18 19 20 21 22 23 24 25 27 29 30 31 34 (9)
11 12 13 14 15 50 16 17 18 (11)

Abb. 3.5.10: Bespiele der (Teil)Sequenzen (keine “Bestellsequenzen“)

3.5.4.2 Nutzungsmodelle

Jede häufige Sequenz, die eine (mögliche) komplette Klickfolge widerspiegelt, kann auch als Nutzungsmodell betrachtet werden, und zu jedem Modell können eventuell interessante Merkmale gefunden werden.

	Muster
1	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (6)
2	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 41 42 ***43*** 44 52 (5)
3	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 42 ***43*** (8)
4	3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 236 29 30 31 32 33 34 35 36 37 47 38 39 40 41 42 ***43*** 44 (5)
5	3 4 6 7 44 9 10 47 36 37 38 39 40 41 42 ***43***(8)
6	3 4 6 7 44 9 10 47 36 37 38 39 40 42 ***43*** (5)
7	6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (9)
8	6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 41 42 ***43*** 44 52 (6)
9	6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 236 29 30 31 32 33 34 35 36 37 47 38 39 40 41 42 ***43*** 44 (7)
10	6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 236 29 30 31 32 33 34 35 36 37 47 38 39 40 41 42 ***43*** 44 52 (6)
11	6 7 44 9 10 47 36 37 38 39 40 41 42 ***43*** (10)
12	29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (10)
13	29 30 31 34 35 36 37 38 39 40 42 ***43*** (12)
14	34 35 36 37 38 39 40 41 42 ***43*** 44 (10)
15	34 35 36 37 38 39 40 41 42 ***43*** (16)
16	34 35 36 37 38 39 40 42 ***43*** (14)
17	36 37 38 39 40 41 42 ***43*** (26)
18	36 37 38 39 40 42 ***43*** (19)
19	38 39 40 41 42 ***43*** 44 (17)
20	38 39 40 41 42 ***43*** 44 52 (12)
21	38 39 40 41 42 ***43*** (33)

Abb. 3.5.11: (Teil)Sequenzen mit Bestellseite "*43*" aus der Musteranalyse

Beim Bestellen jedes Produkts gibt es verschiedene Pfade, und jeder vollständige Pfad kann als ein Nutzungsmodell interpretiert werden. Nach der Methode der häufigen abgeschlossenen Itemmengen wurden zum Beispiel 21 verschiedene Mengen mit dem Item *43* entdeckt, die man in 6 Nutzungsmodelle einteilen konnte (vgl. Abb. 3.5.6 und Abb. 3.5.11). Einige von derjenigen Itemmengen sind Supermengen von anderen, es kann durch den Vergleich mit tatsächlichen Klickpfaden festgestellt werden, welche Itemmengen den gesamten Bestellpfad darstellen und welche nicht. Somit kann das Interpretieren von Modellen nach der Methode der häufigen abgeschlossenen Itemmengen nicht immer klare Ergebnisse liefern. Mit der Analyse der sequentiellen Muster erhält man gleich die unterschiedlichen Nutzungsmodelle, wenn man nur aus bestimmten Startseiten ausgehende Folgen zulässt. Da Benutzer selten gleiche Pfade der Website durchlaufen (sie müssen eventuell ihre Angaben an verschiedenen Seiten erneut eingeben oder korrigieren), ist es sehr unwahrscheinlich, dass mehrmals vorkommende identische Pfade gefunden werden. Deswegen werden zur Entdeckung von Nutzungsmodellen die Klickfolgen ohne wiederholte Seiten herangezogen. Sucht man mit dieser Methode Klickmuster, die mindestens 5 Mal vorgekommen sind, findet man 21 unterschiedliche (Teil)Sequenzen mit der Seite *43*, die sich eventuell weiter in die Gruppen zusammenschließen lassen. Die in der Abbildung 3.5.11 aufgeführten Sequenzen 1 bis 6 kann man in die 3 Cluster einteilen: 1. Folge in das 1. Cluster, 2., 3. und 4. Folge in das 2. Cluster und 5. und 6. Folge in das 3. Cluster. Die Folgen 7 bis 21 sind Teilsequenzen, die zu mehreren Clustern gehören können.

Wenn man mit dieser Methode gewonnene Nutzungsmodelle mit Modellen aus der Analyse mit häufigen abgeschlossenen Itemmengen

3 4 6 7 44 9 10 47 36 37 38 39 40 42 ***43*** (15)
3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42 ***43*** (11)
3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 42 ***43*** (21)

Abb. 3.5.12: Hauptpfade aus der “abgeschlossenen Itemmengen“– Analyse

3 4 6 7 44 9 10 47 36 37 38 39 40 41 42 ***43*** (8)
3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ***43*** (6)
3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 26 210 211 212 29 30 31 34 35 36 37 38 39 40 41 42 ***43*** 44 52 (5)

Abb. 3.5.13: Hauptpfade aus der sequentiellen Analyse

vergleicht, sieht man, dass die gelieferten Ergebnisse zweier Methoden nicht gleich sind. Wenn man die Ergebnisse dieser Methode als Grundlage zur Optimierung heranzieht, so werden 3 ähnliche (wie bei der ersten Methode) Hauptpfade erkennbar (vgl. Abb. 3.5.12 und Abb. 3.5.13), an die die Navigation angepasst werden soll.

Durch den Vergleich der Klickfolgen der Besteller und Nichtbesteller erhält der Anbieter Informationen über "problematische" Seiten auf der Website, die er bei der Überarbeitung der Website nutzten soll. Auch ermittelte Nutzungsmodelle müssen dabei berücksichtigt werden

3.5.5 Einteilung der Benutzer in Gruppen

Eine Möglichkeit, Benutzer nach ihrem Verhalten in Gruppen einzuteilen, ist Anwendung von Assoziationsregeln. Sie können sowohl aus den häufigen abgeschlossenen Itemmengen als auch aus den Bestellsequenzen abgeleitet werden. In diesem Kapitel wird die Einteilung der Kunden in die Produktgruppen mit Assoziationsregeln, die aus den Bestellsequenzen gewonnen wurden, mit der Einteilung durch Assoziationsregeln aus den häufigen abgeschlossenen Itemmengen verglichen.

3.5.5.1 Häufige Abgeschlossene Itemmengen

Bei der Einteilung der Kunden in die Produktgruppen mit Assoziationsregeln als Kriterien sollte man solche Regeln nutzen, die einerseits möglichst früh (bezüglich des Ereignisauftretens in einer Klickfolge) das Bestellen vorhersagen und andererseits nicht zu hohe Fehlerquote aufweisen.

Die Tabelle 3.5.4 zeigt die tatsächliche Benutzereinteilung in Gruppen gemäß ihren abgegebenen Bestellungen.

#Besteller von Produkt A (#Sessions)	#Besteller von Produkt B (#Sessions)	#Besteller von Produkt C (#Sessions)	#Besteller von Produkt D (#Sessions)	#Nicht-Besteller
22 (52)	6(21)	11(27)	26(68)	7

Tabelle 3.5.4: Benutzereinteilung nach bestellten Produkten

Wie die Anzahl der Regeln in Abhängigkeit von verschiedenen Support- bzw. Konfidenz-Werten schwankt, ist in der Tabelle 3.5.5 zu sehen. Dabei wurden nur nicht redundanten Regeln gezählt.

Min.support	Min. confidence	#Frequent Closures	#Rules
5	0.2	416	142
5	0.4	416	139
5	0.6	416	186
5	0.75	416	488
5	0.85	416	631
10	0.2	327	130
10	0.4	327	123
10	0.6	327	134
10	0.75	327	265
10	0.85	327	340
15	0.2	272	97
15	0.4	272	89
15	0.6	272	87
15	0.75	272	130
15	0.85	272	156
20	0.2	211	80
20	0.4	211	73
20	0.6	211	69
20	0.75	211	102
20	0.85	211	127
25	0.2	155	64
25	0.4	155	57
25	0.6	155	54
25	0.75	155	90
25	0.85	155	117

Tabelle 3.5.5: aus den häufigen abgeschlossenen Itemmengen abgeleitete Regeln

Die Tabelle 3.5.6 zeigt die Klassifikation der Sessions nach den Regeln mit verschiedenen minimalen Supports und Konfidenz. Bei dieser Klassifikation wurden Sessions nach der ersten geeigneten Regel einem Produkt zugewiesen. D.h. bei der anderen Reihenfolge der Regeln könnte die Klassifikation zu anderen Ergebnissen führen. Man sieht, dass bei klein gewählter Konfidenz die Fehlerquote ziemlich groß ist. Der angegebene Durchschnittsfehler ist der Trainingfehler (d.h. der Fehler der Klassifikation von bekannten Objekten), im Folgenden wird gezeigt, wie der Durchschnittsfehler der Klassifikation einer Testmenge (d.h. die Menge von unbekannten Objekten) ermittelt werden kann.

Min. Support	Min Confidence	#Sessions mit Produkt A	#Sessions mit Produkt B	#Sessions mit Produkt C	#Sessions mit Produkt D	Durchschnittsfehler (in %)
		52	21	27	68	
5	0.2	235	13	114	39	58,6
5	0.4	104	36	72	96	45,8
5	0.6	89	34	62	105	42,8
5	0.75	76	34	60	97	44,5
5	0.85	76	23	59	97	31,0
10	0.2	230	14	117	40	57,1
10	0.4	94	34	72	105	45,1
10	0.6	89	34	59	105	42,2
10	0.75	76	33	53	97	36,6
10	0.85	76	23	52	97	29,4
15	0.2	230	14	117	40	57,1
15	0.4	94	34	72	105	45,1
15	0.6	89	34	53	105	40,9
15	0.75	76	22	45	97	26,4
15	0.85	76	22	45	88	24,6

Tabelle 3.5.6: Klassifikation der Sessions nach jeweils einer Regel

Um die Klassifikation der Sessions und somit auch der Benutzer zu verfeinern, sollte man bei der Einteilung der Sessions die Anzahl der gedeckten Regeln erhöhen bzw. maximieren.

Mit der Anzahlmaximierung der gedeckten Regeln lassen sich bessere Ergebnisse erzielen, allerdings nur bei kleiner minimalen Konfidenz wie 20% bzw, 40%. Bei der minimalen Konfidenz wie 60%, 75% oder 85% sind die Ergebnisse sehr ähnlich wie bei der Klassifikation nach einer Regel (vgl. Tabellen 3.5.6 und 3.5.7). Das bedeutet, dass bei der Einteilung der Besucher in die Gruppen die Deckung durch eine einzige Regel genügt, wobei die Konfidenz über 50% gewählt werden soll.

Min. Support	Min Confidence	#Sessions mit Produkt A	#Sessions mit Produkt B	#Sessions mit Produkt C	#Sessions mit Produkt D	Durchschnittsfehler (in %)
		52	21	27	68	
5	0.2	157	32	114	98	52,0
5	0.4	100	37	72	99	46,2
5	0.6	89	34	62	105	42,8
5	0.75	76	34	60	97	30,5
5	0.85	76	23	59	97	31,0
10	0.2	151	34	117	99	52,9
10	0.4	94	39	72	100	46,3
10	0.6	89	39	59	100	43,4
10	0.75	76	33	53	97	36,6
10	0.85	76	23	52	97	29,4
15	0.2	151	34	117	99	52,9
15	0.4	94	39	72	100	46,3
15	0.6	89	34	53	105	40,9
15	0.75	76	22	45	97	26,4
15	0.85	76	22	45	88	24,6

Tabelle 3.5.7: Klassifikation der Sessions nach mehreren Regeln

Die in der Tabelle 3.5.8 aufgeführte Klassifikation wurde nach den Regeln mit dem minimalen Support 15 und der minimalen Konfidenz 0.85 durchgeführt. Mit Ausnahme des Produktes B liegt die Anzahl falsch klassifizierter Sessions im Schnitt bei 20, wobei die Anzahl der Bestell-Sessions zwischen 27 und 68 liegt. Somit fällt die Fehlerquote bei der Benutzerklassifikation von 16.7% bis zu 81.8% aus. Zu den Nichtbestellern wurden keine Angaben über zugehörige Sessions gemacht, weil sie mit Sessions von Bestellern, die in machen Fällen nicht bestellt haben, zusammengezählt wurden. Diese Klassifikation verbirgt jedoch einen beträchtlichen Fehler, nämlich die Lernmenge und die Testmenge sind identisch. Dadurch wird der Trainingsfehler minimiert, aber es kann die so genannte Überanpassung auftreten. Deswegen um einen wahren Fehler zu ermitteln und zu reduzieren, muss die Datenmenge in Lernmenge und Trainingmenge aufgeteilt werden. Bei der 10-fachen Kreuzvalidierung wird diese Prozedur 10 Mal wiederholt: die Testmengen sind disjunkt und enthalten jedes Mal 40 verschiedene Transaktionen, die übrig gebliebenen 361 Transaktionen bilden die Trainingsmenge. Jedes Mal wird auch der Fehler berechnet. Der wahre

Fehler der Klassifikation ist dann der aus 10 Versuchen gebildete durchschnittliche Fehler.

	#Besteller (#Sessions)	#Besteller nach Klassifikation (#Sessions)	# falsch klassifizierter Benutzer (Sessions)	Benutzer/ Sessions Klassifikations-fehler (in %)
Produkt A	22(52)	31(76)	9(24)	29.0 / 31.6
Produkt B	6(21)	7(22)	1(1)	14.2 / 4.5
Produkt C	11(27)	20(45)	9(18)	45.0 / 40.0
Produkt D	26(68)	32(88)	6(20)	18.8 / 22.7
Nicht-Besteller	7	2	5	71.4

Tabelle 3.5.8: Klassifikation der Besucher nach mehreren Regeln

min. support	min. confidence	Error Produkt A(%)	Error Produkt B(%)	Error Produkt C(%)	Error Produkt D(%)	Error Nicht-Besteller(%)
5	0.85	35,4	20	51	31,4	49,6
15	0.85	31,2	2	38,5	28,1	36,0

Tabelle 3.5.9: Durchschnittsfehler nach 10-facher Kreuzvalidierung

Aus der Tabelle 3.5.9 wird wieder deutlich, dass der aus der Klassifikation mit dem minimalen Support 15 resultierende Fehler wesentlich kleiner ist als aus der Klassifikation mit dem minimalen Support 5. Die wahren Fehler der Sessionsklassifikation (2. Zeile, Spalten 3 bis 7), die aus 10-fachen Kreuzvalidierung berechnet wurden, sind sehr ähnlich den Werten aus der Tabelle 3.5.8; der wahre Fehler der Nichtbesteller-Klassifikation ist jedoch bedeutend kleiner, was auf die Überanpassung der ersten Klassifikation zurückzuführen ist. Der Klassifikationsfehler ist (mit Ausnahme der zu dem Produkt B zugeordneten Sessions) relativ groß, so dass im ähnlichen Fall das Verfahren zu Klassifikation der Benutzer nicht zu empfehlen ist.

Weitere Bewertungskriterien der Klassifikation sind Precision- und Recall-Werte. Diese Kennzahlen geben die Genauigkeit und Vollständigkeit eines Verfahrens wieder, in diesem Fall der Klassifikation. Durch diese Zahlen wird die Genauigkeit der Sessionsklassifikation vorgestellt, unabhängig davon welches Produkt bestellt wurde, was bei der Berechnung des Durchschnittsfehlers zu Verfälschung geführt hätte (weil die Anzahl der Bestellsessions von einem Produkt zu einem anderen sehr stark schwankt).

$$R = \frac{P_+}{P_+ + P_-}$$

$$P = \frac{P_+}{P_+ + N_+}$$

P_+ : die Anzahl der korrekt klassifizierten positiven Einträge (in den Sessions wurde gekauft und die Sessions wurden als solche klassifiziert)

P_- : die Anzahl der falsch klassifizierten positiven Einträge (in den Sessions wurde gekauft und die Sessions wurden als "nicht gekauft" klassifiziert)

N_+ : die Anzahl der falsch klassifizierten negativen Einträge (in den Sessions wurde nicht gekauft und die Sessions wurden als solche klassifiziert)

Precision gibt also den Anteil der Bestellsessions unter den als "gekauft" klassifizierten Sessions an, Recall gibt den Anteil der Bestellsessions an, die klassifiziert wurden. Die Idealwerte für Precision und Recall sind jeweils 100%.

Min. support	*Min. confidence*	*Precision(%)*	*Recall(%)*
5	0.85	65.1	100
10	0.85	67.7	100
15	0.85	72.7	100

Tabelle 3.5.10: Klassifikationsbewertung

Die in der Tabelle 3.5.10 vorgeführten Werte wurden für die Klassifikation nach mehreren Regeln mit verschiedenen Support und Konfidenz ermittelt. Man sieht, dass die Klassifikation alle Bestellsessions abdeckt, d.h. Sitzungen, in denen gekauft wurde, sind immer als solche klassifiziert wurden. Mit dem steigenden Support wird jedoch der Anteil der falsch klassifizierten Nicht-Bestellsessions kleiner. Diese Werte wurden allerdings für die Klassifikation ermittelt, die die gesamte Beispielmenge sowohl zum Lernen als auch zum Testen genutzt hat. In der Tabelle 3.5.11 sind Precision- und Recall-Werte für die Klassifikation anhand der 10-fachen Klassifizierung zu sehen; die sind den Werten aus Tabelle 3.5.10 sehr ähnlich.

Min. support	*Min. confidence*	*Precision(%)*	*Recall(%)*
5	0.85	64.8	100
15	0.85	71.6	100

Tabelle 3.5.11: Klassifikationsbewertung nach 10-facher Kreuzvalidierung

3.5.5.2 Häufige Sequenzen

Da häufige (Teil)Sequenzen den häufigen Itemmengen entsprechen, kann man aus ihnen auch Regel zu Sessions- und Benutzerklassifizierung ableiten. Dabei muss man beachten, dass die Werte für Support und Konfidenz nicht berechnet, sondern nur abgeschätzt werden können. Diese Werte sind für eine Regel r: $X \Rightarrow Y$ folgendermaßen definiert: $sup(r) = sup(X \cup Y)$ und $conf(r) = sup(X \cup Y) / sup(X)$. Ersetzt man $sup(X \cup Y)$ durch die Häufigkeit einer Sequenz, aus der die Regel r abgeleitet wird, so schrumpfen die Werte für Support und Konfidenz nach unten. Dementsprechend muss man einen kleineren Support und eine kleinere Konfidenz zur Generierung von Regeln wählen.

In der Tabelle 3.5.12 ist die Klassifikation der Sessions anhand von Regel vorgestellt, die aus den häufigen (Teil)Sequenzen gewonnen wurden. Dabei wurden Sessions nach der ersten geeigneten Regel jeweils einem Produkt zugeordnet. Diese Klassifikation ist viel besser gegenüber der Klassifikation ausgefallen, die anhand von Regel aus der “Abgeschlossene Itemsets“-Analyse durchgeführt wurde (Der Durchschnittsfehler ist ein Trainingsfehler). Dies kann dadurch erklärt werden, dass die tatsächlichen Werte für den Support und die Konfidenz einer Regel viel höher liegen, als die Häufigkeit der entsprechenden Sequenz und aus dieser Häufigkeit abgeleitete Konfidenz. Was dabei auch interessant erscheint, ist, dass bei der Werteauswahl für Support und Konfidenz die höheren Werte nicht unbedingt bessere Ergebnisse liefern. So sieht man beispielsweise, dass bei dem minimalen Support 15 keine einzige Session dem Produkt B zugeordnet werden konnte und bei der Konfidenz 0.6 die Sessions mit der Bestellung des Produktes A exakt klassifiziert wurden, bei der höheren Konfidenz dagegen, die ein ziemlich sicheres Ergebnis liefern sollte, wurden sehr wenige (und nicht unbedingt oft zutreffende) Regeln gefunden, so dass mit steigender Konfidenz sogar die Fehlerquote gestiegen ist. Allerdings ist der Durchschnittsfehler am kleinsten bei dem Support 5 und Konfidenz 0.85. Aus dieser Tabelle wird ersichtlich, dass die Auswahl der Häufigkeit einer Sequenz und somit der aus dieser Sequenz abgeleiteten Regel entscheidend für die Klassifikation der Sessions ist. Dies liegt in der Natur der entdeckten Sequenzen: je höher man die minimale Häufigkeit der Sequenz wählt, desto größer ist die Wahrscheinlichkeit, dass nur wenige oder überhaupt keine Sequenzen und folglich keine Regel gefunden werden.

Min. Support	Min. Confidence	#Sessions mit Produkt A	#Sessions mit Produkt B	#Sessions mit Produkt C	#Sessions mit Produkt D	Durchschnittsfehler (in %)
		52	21	27	68	
5	0.2	78	38	76	100	43,6
5	0.4	66	30	49	76	26,0
5	0.6	52	25	39	74	13,7
5	0.75	35	25	32	68	25,0
5	0.85	33	23	29	68	12,9
10	0.2	77	38	45	100	37,2
10	0.4	66	25	45	76	21,9
10	0.6	52	25	39	74	13,7
10	0.75	33	21	32	68	13,0
10	0.85	33	15	29	68	17,9
15	0.2	77	0	39	105	49,5
15	0.4	66	0	39	76	40,6
15	0.6	52	0	39	74	34,7
15	0.75	33	0	32	68	38,0
15	0.85	33	0	29	68	35,8

Tabelle 3.5.12: Klassifikation der Sessions nach einer Regel

	#Besteller (#Sessions)	#Besteller nach Klassifikation (#Sessions)	# falsch klassifizierter Benutzer(Sessions)	Benutzer/ Sessions Klassifikations-fehler (in %)
Produkt A	22(52)	19(33)	3(19)	13.6 / 36.5
Produkt B	6(21)	8(23)	2(2)	25.0 / 8.7
Produkt C	11(27)	13(29)	2(2)	15.4 / 6.9
Produkt D	26(68)	26(68)	0(0)	0
Nicht-Besteller	7	7	0	0

Tabelle 3.5.13: Klassifikation der Besucher nach einer Regel

Die in der Tabelle 3.5.13 aufgeführte Klassifikation wurde nach der ersten passenden Regel mit dem minimalen Support 5 und der

minimalen Konfidenz 0.85 durchgeführt; mit diesen Regeln hatte man den kleinsten Trainingsfehler bei der Sessionsklassifikation. Die Anzahl der falsch klassifizierten Sessions mit Produkt A gegenüber anderen Sessions ist viel höher und beträgt 19 Sessions gegenüber 0 bzw. 2 anderen falsch klassifizierten Sessions. Dennoch werden nur 3 Benutzer nicht als Besteller eingestuft und sie gehen in die Statistik mit der zweitkleinsten Fehlerquote ein.

Auch für dieses Klassifikationsverfahren wurden wahre Fehler nach 10-facher Kreuzvalidierung ermittelt, die in Tabelle 3.5.14 zu sehen sind. Hier erkennt man deutlich, dass höhere Support-Werte bei dieser Methode einen größeren Fehler verursachen können. Dies geschieht insbesondere in den Fällen, wenn sehr wenige (oder gar keine) Regeln gefunden werden aufgrund des kleinen Unterschieds zwischen dem Support und Anzahl der Lernbeispiele. Hier wurde z.B. Support gleich15 gewählt, es gab aber höchstens 21 Lernbeispiele für das Produkt B. An diesem Beispiel wird auch der Effekt der Überanpassung deutlich: bei diesen Support- und Konfidenz-Werten wurde bei der Klassifikation der Sessions für das Produkt B zuerst der Fehler 100% ermittelt; nach 10-facher Kreuzvalidierung kommt man auf 80% Fehler. Die Klassifikation mit Support 5 und Konfidenz 0.85 zeigt (mit Ausnahme des Produkts A) sehr gute Ergebnisse. Hier könnte man z.B. zur Klassifikation der Sessions, die sich auf das Produkt A beziehen, Regel hinzuziehen, die mit Support 5 und Konfidenz 0.6 gewonnen wurden.

min. support	min. confidence	Error Produkt A(%)	Error Produkt B(%)	Error Produkt C(%)	Error Produkt D(%)	Error Nicht-Besteller(%)
5	0.85	40.3	18.4	5.8	0.0	4.8
15	0.85	43.7	80.0	5.8	0.0	16.2

Tabelle 3.5.14: Durchschnittsfehler nach 10-facher Kreuzvalidierung

Dass höhere Support-Werte bei dieser Methode einen größeren Fehler verursachen können, wird noch mal durch die berechneten Precision- und Recall-Kennzahlen bestätigt. Die Tabelle 3.5.15 zeigt diese Werte für Klassifikation, bei der die Lernmenge auch als Testmenge agiert hat, und Tabelle 3.5.16 zeigt sie nach 10-facher Kreuzvalidierung. Die Genauigkeitskriterien sind in beiden Fällen ziemlich nah den optimalen Werten. Die Vollständigkeit (Recall) ist wesentlich höher beim kleinen Support.

Min. support	*Min. confidence*	*Precision(%)*	*Recall(%)*
5	0.6	88.4	100
5	0.85	97.4	88.7
15	0.85	98.5	76.2

Tabelle 3.5.15: Klassifikationsbewertung

Min. support	*Min. confidence*	*Precision(%)*	*Recall(%)*
5	0.85	97.0	89.4
15	0.85	98.8	75.4

Tabelle 3.5.16: Klassifikationsbewertung nach 10-facher Kreuzvalidierung

Nach der Gegenüberstellung der Klassifikationsfehler und Bewertungskriterien von 2 Methoden zur Klassifikation von Sessions und Benutzer zeigt sich die Methode der häufigen (Teil)Sequenzen genauer. Man sollte bei diesem Verfahren Regeln mit dem relativ kleinen minimalen Support und der minimalen Konfidenz von mindestens 50% heranziehen. Die tatsächliche Häufigkeit und Konfidenz der Regeln liegen viel höher als minimale zulässige Werte und deswegen weist die Klassifikation ziemlich gute Ergebnisse auf. Je nachdem, ob man eine breitere Gruppe erreichen möchte oder nicht, soll man das Verfahren wählen, das höhere Fehlerquote erlaubt oder bessere Genauigkeit bietet. Hat man z.B. als Zielgruppe auch die potentiellen Auftraggeber gesetzt, sollte man das Verfahren der “Abgeschlossene Itemsets“-Analyse anwenden. Überschreitet der Klassifikationsfehler für bestimmte Benutzer eine gewisse Toleranzgrenze (z.B. 20%), sollte man für diese Benutzer eine allgemeingültige zielgerichtete Strategie in Betracht ziehen.

3.5.5.3 Interpretation von Assoziationsregeln

Die Regeln selbst liefern interessante Erkenntnisse über die Website-Besucher. Z.B. Regel „3 => 43“ bei Support 10 und Konfidenz 0.2 besagt, dass über 20% User, die Seite «*index*» besucht haben, das Produkt A bestellt haben. Unter den Regeln mit Konfidenz 0.4 ist diese Regel nicht mehr vorhanden. Das bedeutet, dass der Anteil solcher Besucher zwischen 20% und 40% liegt. Wenn man die Regeln “26 => 43“ bzw. “ 8 & 9 => 43“ mit der Konfidenz 0.75 heranzieht, kann man sagen, dass über 75% User, die die Seite “26“ bzw. die Seiten “8“ und “9“ besucht haben, ihre Bestellung des Produktes A abgegeben haben. In der 1. Klickfolge aus der Abbildung 3.5.9 steht “26“ an der 16. Stelle und “8 & 9“ an der 4. und 5. Stelle. Da Seiten “8“ und “9“ früher in der Bestellungssequenz

vorkommen und die Regeln “8=>43“, “9=>43“ die Bedingung der Konfidenz 0.75 nicht erfüllen, kann man erst nach Besuch beider Seiten “8“ und “9“ die Bestellung mit Sicherheit von mindestens 75% vorhersagen. Diese Erkenntnisse kann der Anbieter ebenfalls bei der Reorganisation der Website nutzen. Nämlich sollten die Seiten, die als Regelprämissen in den Regeln mit niedrigen Konfidenz vorhanden und in den Regeln mit hohen Konfidenz nicht vorhanden sein, auf ihren Inhalt und ihre Funktionalität überprüft werden.

3.6 Vergleich mit der erwarteten Nutzung

3.6.1 Unerwartete Nutzung

Um das Nutzungsverhalten von Kunden mit erwarteter Nutzung zu vergleichen, muss man zuerst definieren, was der Web-Designer unter erwarteter Nutzung versteht. Bei der Entwicklung einer Website steht der Designer vor der Aufgabe, den Kunden mit jedem Schritt immer näher zum Ziel zu bringen. Somit bemüht er sich, die Anzahl der Seiten (bzw. der auszufüllenden Formulare oder notwendigen Angaben) so klein wie möglich zu halten, und die Web-Seiten nur mit wichtigem Inhalt zu füllen. Er erwartet folglich von dem User, dass dieser, wenn er alle Hinweise beachtet, sicher das Ziel erreicht. Idealerweise sollte der Kunde immer weiter klicken und so zum Ziel (Bestellung) kommen. D.h. dass im Idealfall bewegt sich der Benutzer entlang eines “roten Fades“, ohne zu vorher besuchten Seiten zurückzugehen. Der Designer versucht, alle notwendigen Informationen zu vermitteln und Werkzeuge zu Verfügung zu stellen, damit der Kunde dem so genannten “roten Faden“ folgen könnte. Man kann solche “rote Faden“ als erwartete Nutzung bezeichnen. In der Praxis ist dies selten der Fall, dass der User alles vom Anfang an richtig macht und dabei alle Dienste einwandfrei funktionieren, so dass man in der Regel mit einigen wenigen oder mehreren Wiederholungen in der Klickfolge rechnen muss. Man kann sagen, dass einige wenige Wiederholungen den Normalfall darstellen, mehrere an einer Stelle werden dagegen nicht erwartet. Dass bei der Eingabe von Daten bzw. Abschicken oder Bearbeiten von Dateien oder wegen technischer Probleme zu Fehlermeldungen kommt, ist zwar unerwünscht aber bei der Entwicklung der Website berücksichtigt. Für solche Fälle überlegt sich der Entwickler eine Strategie, wie er dem Kunden aus der “Sackgasse“ helfen kann und versucht ihn wieder auf den richtigen Weg (“roten Faden“) zu bringen. Wenn der Benutzer trotzdem die Sitzung abbricht, kann man dieses Verhalten als “nicht erwartet“ bezeichnen. Mit anderen Worten kann man sagen, dass jede Session, die nicht mit der Bestellung endet, stellt nicht erwartete Nutzung dar. Mein Ziel ist es, solche

Sessions zu untersuchen und Auffälligkeiten zu berichten. Außerdem können als nicht erwartete Nutzung auch mehrere sich wiederholende Besuche von Seiten betrachtet werden. Auch nicht bzw. selten besuchte Seiten bzw. Pfade stellen eine nicht erwartete Nutzung dar. Auf diese wird der Anbieter ebenfalls aufmerksam gemacht.

Die nicht erwartete Nutzung kann durch Aufzeigen verschiedener Informationen wie

- nicht bzw. selten besuchte Seiten
- nicht genutzte Links
- nicht besuchte verlinkte Seiten
- nicht besuchte Pfade
- Seiten, die selten oder gar nicht in den Bestellungssessions vorkommen
- häufige zyklische Wiederholungen in den Klickfolgen
- Seiten, die häufiger als Bestellung vorkommen

präsentiert werden. Diese Informationen lassen sich am besten entweder in einer Text-Datei oder durch graphische Darstellung vorführen.

Die selten besuchten Seiten kann der Anbieter der Besuchstatistik entnehmen. An dieser Stelle wird sich der Anbieter interessieren, warum diese Seiten unerwartet selten angeklickt wurden. Um diese Frage zu beantworten, muss man Pfade von den selten besuchten Seiten zurückverfolgen, bis man eine oft genutzte Seite entdeckt. Kommt eine selten besuchte Seite immer nach einer (oder einigen) anderen selten besuchten Seite(n), so muss man den/die Pfad(e) weiter zurückverfolgen. Man sucht dabei 2 verlinkte Seiten, von denen die erste oft und die zweite selten besucht wurde. Dieser Link ist der Grund, warum die Seite, von der die Suche begonnen wurde, selten angeklickt wird. So kann man für jede selten genutzte Seite Pfade anzeigen, die mit einer oft besuchten Seite starten und mit dieser selten genutzten Seite enden. Man kann aber auf die vollständige Darstellung solcher Pfade verzichten und nur interessante Links, d.h. die ersten 2 Seiten dieser Pfade, dem Anbieter präsentieren. Zu 47 selten genutzten Seiten wurden 62 Pfade und 45 interessante Links gefunden. In der Abbildung 3.6.1 sind die Beispiele solcher Pfade zu sehen. Alle durch einen Pfeil getrennten Items entsprechen einem Link, nur der erste ist der Link von einer oft besuchten zu einer selten besuchten Seite. Alle weiteren Referenzen sind Links zwischen 2 selten angeklickten Seiten. Das letzte Item entspricht einer selten genutzten Seite, von der die Suche begonnen wurde. Als

selten genutzte Seiten wurden solche initialisiert, die von maximal 3 Benutzern angeklickt worden waren.

48 -> **49**
48 -> **105**
3 -> **53**
3 -> **54**
63 -> **64** -> **65** -> **66** -> **67** -> **68**
83 -> **84** -> **85**
82 -> **85**
87 -> **88**
61 -> **95**
69 -> **97** -> **98** -> **99** -> **100** -> **101**

Abb. 3.6.1: Pfade zu selten genutzten Seiten

Es ist interessant, die entdeckten Teilpfade, die zu seltenen Seiten führen, mit den abgeschlossenen Itemsets der seltenen Seiten zu vergleichen. Die abgeschlossenen Itemsets sind mit der in Kapiteln 2.4 und 2.5 beschriebenen Methode zu ermitteln. Als Kriterium für Interessantsein wird in diesem Fall nicht die minimale, sondern maximale Häufigkeit genutzt. Die abgeschlossenen Itemsets der seltenen Items zeigen, mit welchen anderen (häufigen oder seltenen) Items sie zusammen auftreten (im Gegensatz zu abgeschlossenen Itemsets von häufigen Items beinhalten abgeschlossene Itemsets von seltenen Items auch häufige Items). Andererseits wird aus dieser Darstellung nicht ersichtlich, aus welchen häufigen Seiten die seltenen Seiten zu erreichen sind, weil diese Darstellung die wiederholten Seiten nicht beinhaltet, und folglich entsprechen 2 nacheinander folgende Items nicht unbedingt einem möglichen bzw. genutzten Link. Das sieht man am Beispiel des

3 4 48 **49**
3 4 **53 54**
3 4 55 56 58 59 60 61 63 **64 65 66 67 68**
3 4 55 56 57 61 63 64 65 66 67 68 78 79 80 81 82 83 **84 85**
3 4 55 56 86 58 59 87 **88**
3 4 55 56 86 58 59 87 89 90 91 92 93 94 60 69 57 61 **95** 96
3 4 48 **105**
3 4 55 56 86 58 59 87 89 90 91 92 93 94 60 69 57 61 95 96 63 64 65 66 67 68 **97 98 99 100 101** 78 79 80 81 82 83 84 85

Abb. 3.6.2: Beispiele der seltenen abgeschlossenen Itemsets

letzten Itemsets aus der Abbildung 3.6.2: der Link “69 -> 97“ (s. Abb. 3.6.1) kann man aus dem Itemset nicht erkennen, weil zwischen den genannten Items sehr viele andere stehen. Somit führt die Darstellung wie in der Abbildung 3.6.1 direkt zu problematischen Links, was durch die Darstellung wie in der Abbildung 3.6.2 nicht unbedingt ersichtlich ist.

Um nicht besuchte Seiten zu finden, muss man die Sitemap der Website mit der Struktur, die durch die Analyse der Klickfolgen konstruiert wurde, vergleichen und diejenigen, die in der zweiten nicht vorhanden sind, sind exakt die gesuchten nicht besuchten Seiten. Nach diesem Vergleich wurden 97 Seiten gefunden, die in dem durch die Logfiles abgedeckten Zeitraum von keinem User besucht wurden. Auch die nicht genutzten Links werden bei der Gegenüberstellung dieser 2 Strukturen entdeckt. Sollten im zweiten Graphen Kanten aus dem ersten Graphen fehlen, sind diese als nicht genutzte Kanten zu vermerken. Wenn solche Kanten sich außerdem zu Pfaden verbinden lassen, werden sie als nicht genutzte Pfade präsentiert. Außerdem wurden 93 ungenutzte Links zwischen den besuchten Seiten entdeckt. Die ungenutzten Pfade zwischen den nicht besuchten Seiten kann man auf verschiedene Weise zeigen:

1. als Pfade zwischen jeweils 2 nicht besuchten Seiten,
2. als nicht genutzte Pfade, die in einer nicht besuchten Seite starten,
3. als die längsten nicht genutzten Pfade, die in einer nicht besuchten Seite starten,
4. als die längsten nicht genutzten Pfade, die in einer besuchten Seite starten.

Es wurden 51 Pfade der Art 1., 122 Pfade der Art 2., die auch nicht unter 1. sind, 52 längste aus einer nicht besuchten Seite ausgehende Pfade und 57 Pfade der 4. Art gefunden. Die Pfade, die durch 3 erste Darstellungen aufgezeigt werden sind wenig aussagekräftig, sie liefern fast genau so viele Informationen wie die Auflistung einzelner nicht besuchten Seiten. Interessant sind Pfade, die in besuchten Seiten starten und zu nicht besuchten Seiten führen, wie unter 4. Darstellung. Das größte Interesse sollten die Pfade bzw. Links erwecken, die in oft besuchten Seiten starten und zu selten (bzw. nicht) genutzten Seiten führen.

Die nicht besuchten Seiten und Pfade werden in separaten Text-Dateien aufgelistet und in die Struktur integriert, die in dem Visualisierungstool präsentiert wird. In dem Visualisierungstool wird die Sitemap als Graph dargestellt. Jede besuchte Seite wird durch einen rosafarbigen Knoten und jeder besuchte Link durch eine blaue Kante mit der Angabe der Anzahl der User, die diese Seite angeklickt bzw. diese 2 verlinkte Seiten hintereinander besucht haben, präsentiert. Die nicht besuchten Seiten

und Links werden durch andere Farbe (z. B. rot) markiert. Die nicht genutzten Links zwischen 2 besuchten Seiten werden schwarz angezeigt.

Ein Ausschnitt dieser Darstellung ist in Abbildung 3.6.3 zu sehen.

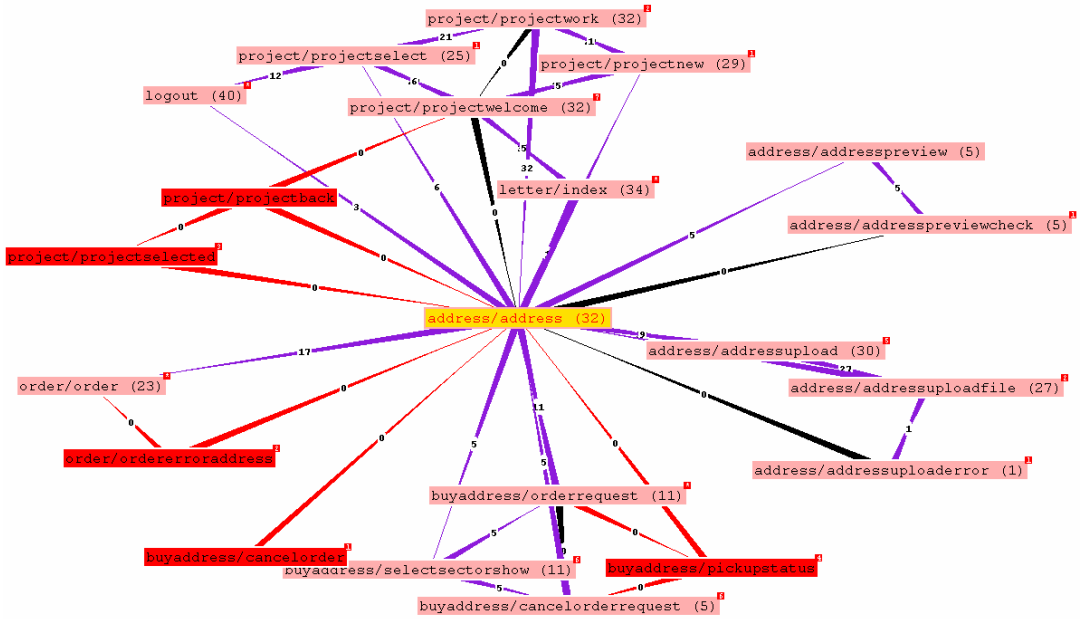


Abb. 3.6.3: Sitemap (Ausschnitt)

In dem Kapitel über gemeinsame Seitenmengen wurden bereits Beispiele aufgeführt, nach welchen Seiten keine (bzw. weniger als in 5 Fällen) Bestellung erfolgt ist (s. Abb. 3.5.4). Diese Informationen sollte der Anbieter besonders beachten, da diese Seiten höchstwahrscheinlich einen der wichtigsten Schlüssel beim Verstehen des Benutzerverhaltens sind, genauer gesagt beim Verstehen der Ursachen der unerwarteten Nutzung. Diese Seiten müssen entweder inhaltlich überarbeitet oder am besten ganz entfernt werden.

Um Abbruchmuster aus der Menge der Sequenzen auszufiltern, habe ich jede Klickfolge gedanklich und somit auch jede Zeile der Eingabedatei um eine Markierung erweitert, d.h. ich habe am Ende jeder Zeile eine spezifische Zahl hinzugefügt. Da ich vorher mit 244 Seitenobjekten gearbeitet habe, wurde als spezifische Zahl 245 ausgewählt. Als Abbruchmuster habe ich diejenigen Folgen ausgesondert, die keine Bestellungsseite und die virtuelle Seite 245 enthalten (s. Abb. 3.6.4). In dem Kapitel über die sequentielle Analyse wurden auch die Klickfolgen aufgezeigt, die ebenfalls zu keiner (bzw. weniger als in 5 Fällen) Bestellung “geführt“ haben (s. Abb.3.5.9). Alle diese Klickfolgen spiegeln ebenfalls unerwartete Nutzung wider und müssen vom Anbieter weiter

analysiert werden. Außerdem sollte unter Umständen auch die Funktionalität einzelner Seiten getestet werden.

3 1 2 107 238 239 55 240 241 245 (6)
3 128 129 126 164 130 245 (5)
36 37 38 39 40 51 245 (6)
56 57 58 59 60 61 62 63 209 61 70 59 52 245 (5)
107 108 207 110 111 208 213 111 52 245 (5)
126 130 141 126 164 128 129 245 (8)
126 130 131 132 133 135 245 (9)
130 3 128 129 245 (5)
141 126 164 128 129 245 (9)

Abb. 3.6.4: Beispiele der Abbruchmuster

Durch Entdeckung von Seiten bzw. Seitenmengen, die in den Mengen von Bestellseiten enthalten sind, aber häufiger als die Bestellung selbst vorkommen, kann Anbieter auch wichtige Anhaltspunkte erhalten. Dies ist der erste Hinweis, dass es viel mehr Sessions gegeben hat, wo der Benutzer auf dem Weg zur Bestellung war, es aus irgendwelchen Gründen nicht getan hat. Unter diesen Seitenmengen müssen "schwierige" Stellen gefunden und möglichst überarbeitet werden. So zeigt sich z.B., dass mindestens in 82 Sessions (die 1. Menge aus Abb. 3.6.5) Kunden mit der Bestellung des Produktes A begonnen haben, dass in 65 Sessions die Seiten aus der 3. Menge besucht wurden, bestellt wurde aber nur in 52 Fällen.

1	3 4 6 7 9 10 (82)
2	3 4 6 7 9 10 11 (55)
3	3 4 6 7 9 10 36 37 (65)
4	107 108 110 111 208 (95)

Abb. 3.6.5: Seitenmengen, die häufiger vorkommen als die Bestellung

In den untersuchten Sessions wurden 28 (Teil)Sequenzen entdeckt, die mehr als 3 sich wiederholenden Seitenbesuche enthalten. Die fett gezeichneten Zahlen in der Abbildung 3.6.6 entsprechen den mehr als 3 Mal in einer Session besuchten Seiten. Da jede wiederholte Dateneingabe bzw. Überprüfung der eingegebenen Daten oder Zurückweisung den Benutzer ärgert, sollte der Anbieter diese Seiten näher in Betracht nehmen und dem User eventuell zusätzliche Hilfswerkzeuge zur Verfügung stellen, um die Servicefreundlichkeit und somit auch die Akzeptanz zu erhöhen.

3 128 3 128 3 128 3 128 3 128 (5)
13 14 **13** 14 **13** 14 **13** (15)
113 114 116 **113 114** 116 **113 114** 116 **113 114** (9)
137 146 137 146 137 146 137 146 137 146 137 146 (5)

Abb. 3.6.6: Beispiele der (Teil)Pfade mit sich wiederholenden Seitenbesuchen

3.6.2 Visualisierung der Website-Nutzung

Durch die Visualisierung der Sitemap erhält der Anbieter die Einsicht in das Benutzerverhalten (s. Abb. 3.6.3). Jeder Knoten sowie jede Kante informieren auch darüber, wie viele User die entsprechende Seite besucht bzw. den dazugehörigen Link genutzt haben.

Die Möglichkeit vorzustellen, wie die Website genutzt wird, kann dem Anbieter durch Präsentieren der Sitemap in Form eines Graphen geboten werden, wobei einzelne Seiten der Website als Knoten und Links als gerichtete Kanten mit den Übergangswahrscheinlichkeiten dargestellt werden. Diese Visualisierung zeigt, mit welcher Wahrscheinlichkeit Benutzer von einer Seite zu einer anderen gehen (s. Abb. 3.6.7), d.h. wie geplante Links genutzt werden oder wie wahrscheinlich das Fehlerauftreten auf bestimmten Seiten ist. Seiten, die durch Links nicht verbunden sind, werden dementsprechend in solchem Teilgraphen durch keine Kante verbunden. Mögliche Links, die von Usern nicht genutzt wurden, werden als schwarze Kanten mit Wahrscheinlichkeit 0.0 dargestellt. Links mit Wahrscheinlichkeit kleiner als 15% werden blau und mit größeren Wahrscheinlichkeit grün gefärbt. Die nicht genutzte Seiten und Links zu ihnen werden rot markiert (wie in Abbildung 3.6.3).

Da bei der Darstellung aller möglichen Pfade sehr umständlich ist, die Übersicht zu behalten, kann es sehr nützlich sein, nur bestimmte Pfade abzubilden, z. B. nur sehr wahrscheinliche Pfade. Hier stellt sich die Frage, welche Pfade für sehr wahrscheinlich gehalten werden sollten. Bei der minimalen Wahrscheinlichkeit von 40%, die als Kriterium festgelegt wurde, wurden viele Kanten ausgesondert und somit Knoten nicht mehr erreichbar. Bei der minimalen Wahrscheinlichkeit von 15% sind nur sehr selten genutzte Links und dadurch auch selten besuchte Seiten ausgefiltert worden. Die Anzahl der eingehenden und ausgehenden Kanten wird trotz der nicht sehr hohen Wahrscheinlichkeit wesentlich reduziert und somit die Übersichtlichkeit bedeutend verbessert (vgl. Abb. 3.6.7 und Abb. 3.6.8).

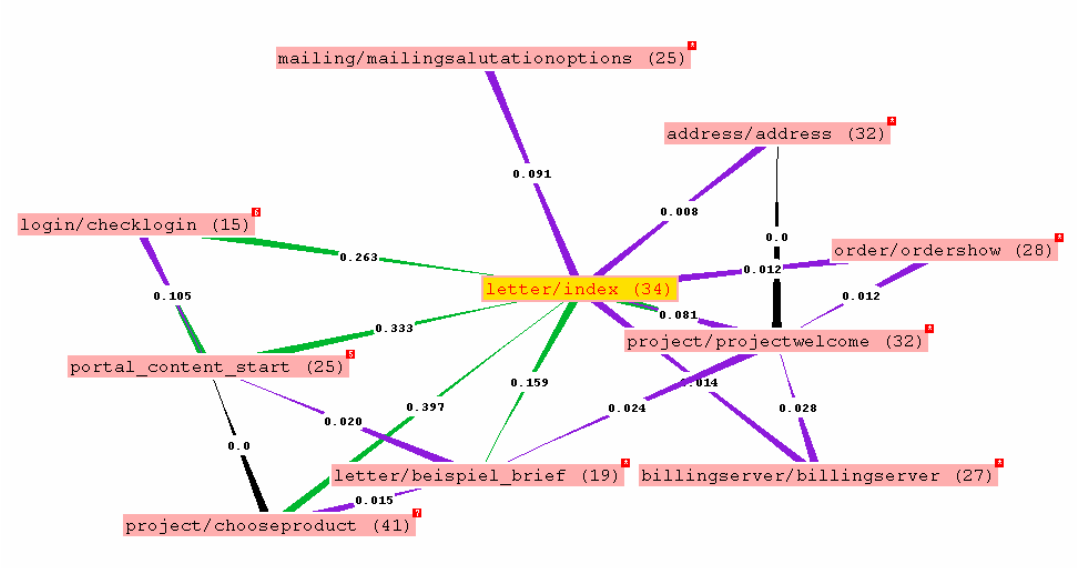


Abb. 3.6.7: Sitemap mit Übergangswahrscheinlichkeiten (Ausschnitt)

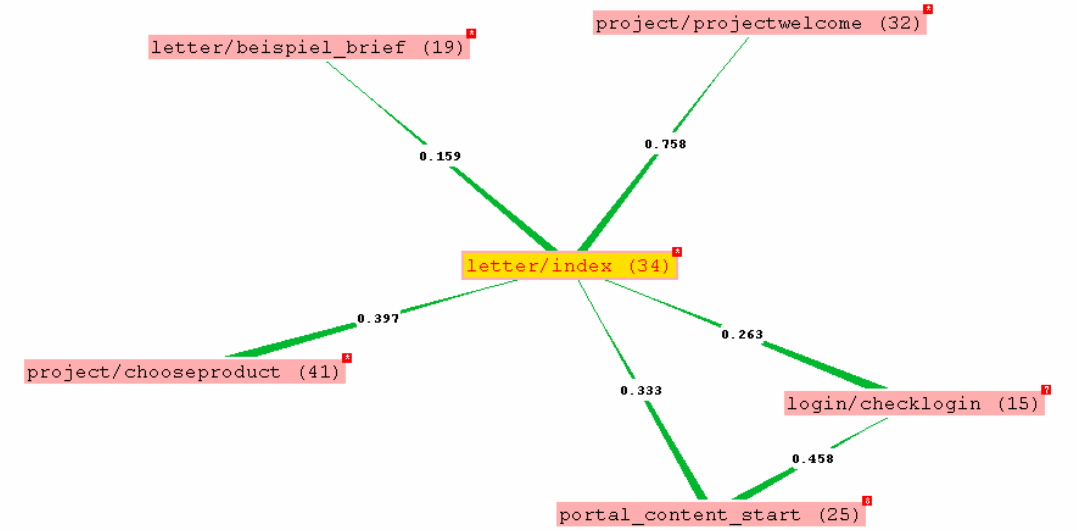


Abb. 3.6.8: Sitemap mit “wahrscheinlichen“ Pfaden (Ausschnitt)

Eine Möglichkeit, Klickfolgen darzustellen, ist in Form einer Text- oder HTML-Datei (s. Abbildung 3.6.9). Bei großer Anzahl der entdeckten Mustern (wie hier 235 (Teil)Sequenzen und 66 Bestell(teil)sequenzen bzw. 382 (Teil)Sequenzen und 82 Bestell(teil)sequenzen mit wiederholten Seiten) ist es sehr kompliziert, problematische Stellen, d.h. sich mehrfach wiederholende Seiten oder Sequenzen, die zu Bestellung nicht “führen“, zu finden. Durch die Eingabe verschiedener Parameter wie minimale Häufigkeit oder minimale Widerholungshäufigkeit von Seiten in einem Pfad oder Start- bzw. Zielseite der Pfade wird die Anzahl der den Analysten interessierenden Pfade wesentlich reduziert. Wie es in der Abbildung 3.6.9 zu sehen ist, kommt der Website-Analyst zu 7

(Teil)Pfaden, die die Seite 13 haben aber die Seite 32 nicht und mindestens 4 Mal sich wiederholende Seiten enthalten. Durch die Angabe einer Startseite und einer Zielseite werden Klickfolgen präsentiert, die zwischen 2 entsprechenden Seiten verlaufen. Man kann die Pfadmenge auf diejenigen Folgen einschränken, die bestimmte Seiten beinhalten bzw. ausschließen. Z.B. wenn man nur die Abbruchsequenzen betrachten möchte, wird die virtuelle Abbruchseite 245 als Bedingung (und alle Bestellseiten als Ausschlussbedingung) eingegeben. Solche Darstellung bietet ein nützliches Werkzeug bei der Analyse des Nutzungsverhaltens von Kunden.

minimale Häufigkeit

Startseite Zielseite

und / oder

Pfade mit Seite 13 und ohne Seiten 32

mit mindestens 4 mal sich wiederholenden Seiten GO

	Episode	support	users
0	3,4,6,7,8,9,10,11,12,13,14,15,**16**,17,18,26,18,26,18,**16**,15,**16**,17,**16**,13,14,15,**16**	8	8
1	3,4,6,7,8,9,10,11,12,**13**,**14**,15,**16**,17,18,26,18,26,18,**16**,15,**16**,17,**16**,**13**,**14**,15,**16**,**13**,**14**,**13**,**14**,**13**,**14**,**13**	6	6
2	3,4,6,7,8,9,10,11,12,**13**,**14**,15,**16**,17,18,26,18,26,18,**16**,15,**16**,17,**16**,**13**,**14**,15,**16**,**13**,**14**,**13**,**14**,**13**	7	7
3	6,7,8,9,10,11,12,13,14,15,**16**,17,18,26,18,26,18,**16**,15,**16**,17,**16**,13,14,15,**16**	11	9
4	6,7,8,9,10,11,12,**13**,**14**,15,**16**,17,18,26,18,26,18,**16**,15,**16**,17,**16**,**13**,**14**,15,**16**,**13**,**14**,**13**,**14**,**13**,**14**,**13**	9	8
5	6,7,8,9,10,11,12,**13**,**14**,15,**16**,17,18,26,18,26,18,**16**,15,**16**,17,**16**,**13**,**14**,15,**16**,**13**,**14**,**13**,**14**,**13**	10	8
6	**13**,14,**13**,14,**13**,14,**13**	15	8

Abb. 3.6.9: Pfaddarstellung in einer html-Datei

Andere Möglichkeit, die Bestell/Nichtbestellpfade zu verfolgen und sie zu analysieren, ist, alle bzw. häufige Klickfolgen durch einen oder mehrere Präfixbäume darzustellen. Jede Startseite wird zur Wurzel eines Präfixbaumes. In vorliegendem Fall gibt es 7 Startseiten, also 7 Präfixbäume. In Abbildung 3.6.10 ist ein Präfixbaum mit der Startseite 3(«*index*») in Zentrum gezeigt. Jeder Knoten trägt in sich auch die Information (in Klammern), wie viele User / in wie vielen Sessions die entsprechende Seite durch den von der Startseite ausgehend zu ihr führenden Pfad erreicht haben. Diese Abbildung ist folgendermaßen zu verstehen: 42 Benutzer in 217 Sitzungen haben die Website im Knoten (Seite) 3 “betreten“, 18 Benutzer in 24 Sessions sind zur Seite 128, 13 in 18 Fällen zur Seite 1 und 41 in 175 Sessions zur Seite 4 gegangen. So erkennt der Anbieter, dass die User nicht immer denselben (d.h. wie in der früheren Sitzung) Navigationspfad nutzen. Diese Darbietung ergänzt die Vorstellung über das Nutzungsverhalten und liefert Daten darüber,

wie oft welche Pfade genutzt wurden. Um lange Ketten der aufeinander folgenden Knoten mit dem einzigen Zielknoten zu vermeiden, wurden solche Ketten in einem Knoten zusammengefasst. So bedeuten Knoten mit mehreren Seiten (die durch einen Strich getrennt sind), dass alle User, die auf diesem bestimmten Weg zu ersten Seite in der Reihe gelangen sind, haben auch alle darauf folgenden Seiten in dieser Reihenfolge besucht. Jede Kante wird dabei mit einem Seitenpaar "beschriftet": Ausgangs- und Zielseite. Man sieht, wie im oberen Kästchen, sich wiederholende Seitenzugriffe (s. auch Abb. 3.6.6). Von 41 Besuchern, die von Seite 3 zur Seite 4 gekommen sind, haben 16 in 19 Sitzungen wieder Seiten 3 und 4 besucht. Diese auffälligen Pfade ziehen die Aufmerksamkeit des Analysten an. Durch diese Darstellung hat man den Vorteil, schnell kurze Abbruchfolgen zu entdecken, lange Abbruch- bzw. Bestellfolgen lassen sich bei verzweigten Bäumen erst nach einer sorgfältigen und mühsamen Suche entdecken.

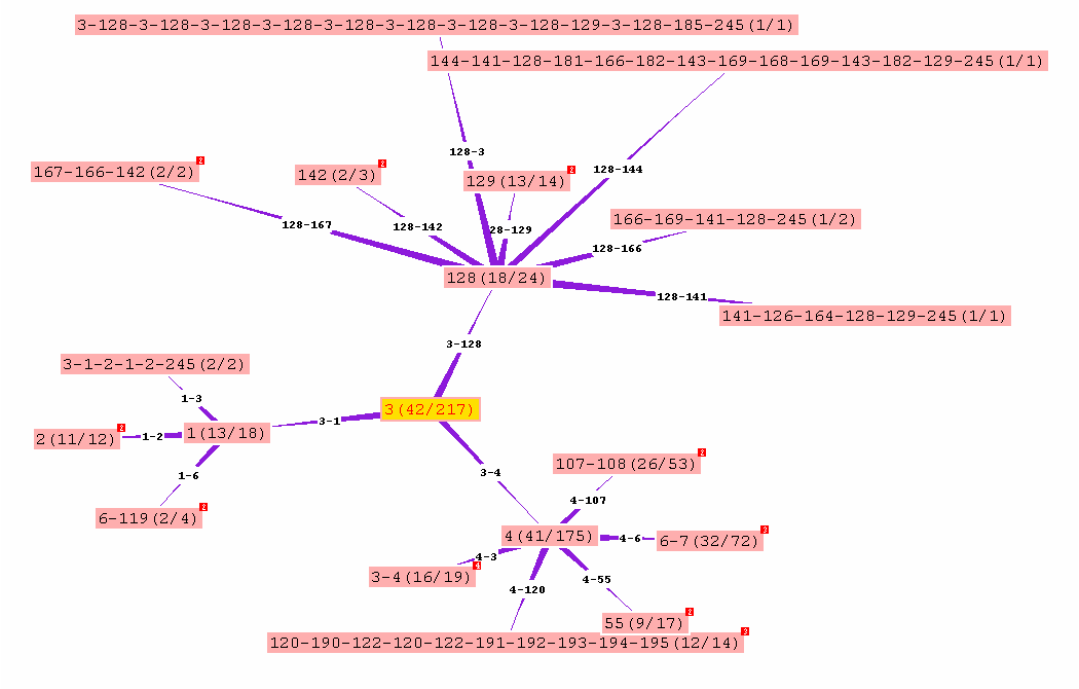


Abb. 3.6.10: Ausschnitt aus einem Präfixbaum

Wichtig kann dem Website-Analysten erscheinen, die Übergangswahrscheinlichkeiten in solchen Präfixbaum einzubeziehen, weil diese Daten ebenfalls die Nutzung widerspiegeln und somit das Userverhalten "sichtbar" machen. Hier kann er die unbedingten und bedingten Übergangswahrscheinlichkeiten vergleichen. Die Kanten des Graphen in der Abbildung 3.6.11 werden durch diese Wahrscheinlichkeiten beschriftet: die erste ist die bedingte und die zweite die unbedingte Übergangswahrscheinlichkeit von der Start- zu Zielseite.

Um die Kanten, deren bedingte und unbedingte Wahrscheinlichkeiten sich um einen gewissen Wert (z.B. um mehr als 15%) unterscheiden, prompt erkennbar zu machen, können sie wie in Abbildung 3.6.11 rot markiert werden. Auch mit Hilfe dieser Daten kann Anbieter bestimmte Teilpfade entdecken, die vom geplanten Verhalten “abweichen“ und deshalb bei der Umstrukturierung bzw. Optimierung der Website beachtet werden müssen.

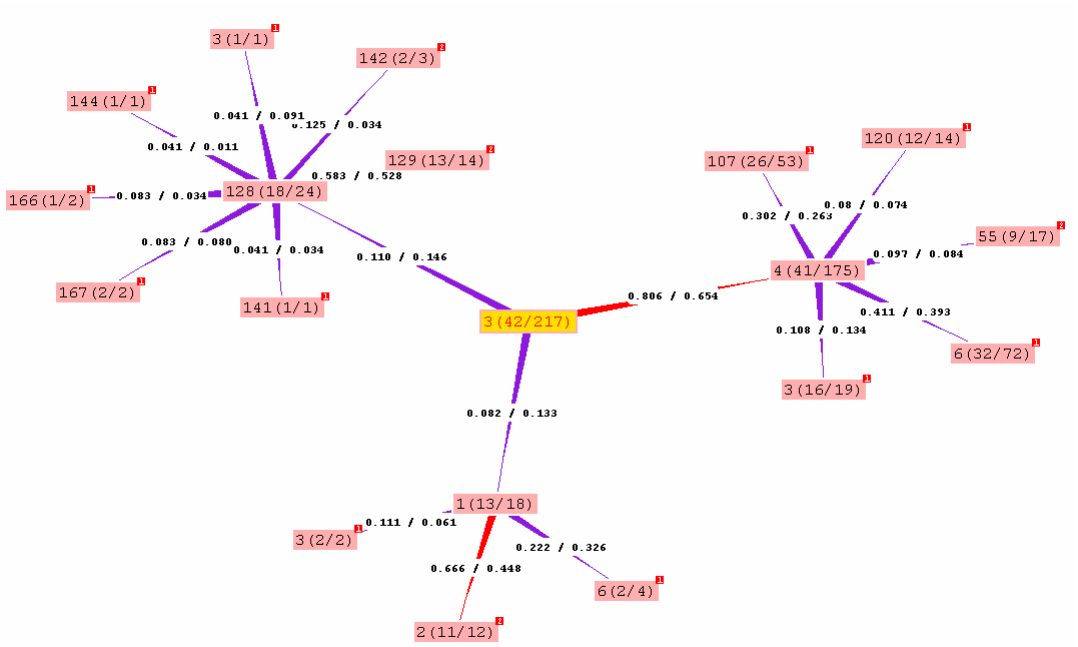


Abb. 3.6.11: Präfixbaum mit Übergangswahrscheinlichkeiten (Ausschnitt)

Zusammenfassend kann man sagen, dass die Pfadverfolgung mit der graphischen Darstellung unter Umständen sehr ineffizient sein kann, andererseits werden die selten/häufig genutzte Links bzw. selten/häufig auftretende Fehler durch farbige Unterschiede hervorgehoben. Auf einer Website mit sehr vielen einzelnen Seiten kann das Auffinden von interessierenden Informationen lange dauern.

3.6.3 Nicht geplante Links

Die Entdeckung von “nicht geplanten Links“ sollte der Erleichterung der Navigation dienen.

Wenn man von der Entdeckung von nicht geplanten Links spricht, wird gemeint, dass die Seitenpaare der Website gesucht werden, die keine direkten Links sind, jedoch miteinander als Links verbunden werden sollten. Dabei müssen die Klickpfade analysiert werden und solche Seitenpaare gefunden werden, die nur durch “Zurück“-klicken nacheinander besucht werden können. Man kann sich diese Seiten als

Blätterknoten eines Baumes vorstellen, auf die man hintereinander nicht zugreifen kann, nur über den gemeinsamen Wurzelknoten (s. Abbildung 3.6.12). Zu diesem Zweck werde ich die Klickfolgen aus der Datei analysieren, die keine wiederholten Seitenobjekte enthält, da jedes hintereinander folgende Seitenpaar aus diesen Folgen einen potentiellen "nicht geplanten Link" darstellt. Die Folgen aus dieser Datei werden mit Folgen aus der Datei mit wiederholten Klicks verglichen und solche Seitenpaare ausgesondert, die nur durch "Zurück"-klicken besucht werden können. Danach sollte man Sitemap auf das Vorhandensein der ausgesonderten Links überprüfen. Die Seitenpaare, denen kein Link in dem Sitemap entspricht, sind die gesuchten "nicht geplanten Links".

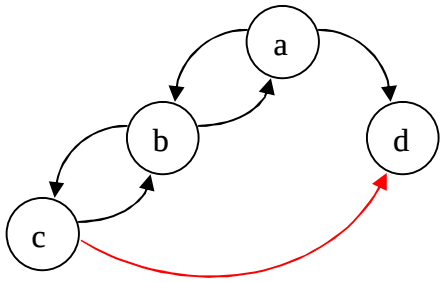


Abb. 3.6.12: nicht geplanter Link c → d

In der Abbildung 3.6.13 sind Beispiele der gefundenen nicht geplanten Links aufgeführt. Es wurden 85 solche Links entdeckt, von ihnen sind jedoch nur 28 mindestens 5 Mal vorgekommen. Man sollte diese Links hinzufügen, weil die Benutzer in manchen Fällen den ganzen Pfad zurückverfolgen müssen, um zur gewünschten Stelle hinzugelangen. So ist es beispielsweise sinnvoll, dem Kunden nach der Vorschau des Auftrages die Möglichkeit zu geben, mit einem neuen Projekt zu beginnen (1. Zeile) oder nach dem Auftragerteilen die Anforderungen wieder anzuschauen (3. Zeile) oder ein bestehendes Projekt wieder in Auftrag zu

1	order/ordershow → project/projectnew
2	mailing/mailingdata → order/orderattachment
3	postcard/order/orderthankyou → postcard/buyaddress/orderrequest
4	project/chooseproduct → portal_content_start
5	distribution/project/projectselect → distribution/order/ordersupply
6	postcard/postcard/postcardeditorsave → postcard/order/pdfpreview
7	distribution/sender/agencieschoice → distribution/buyaddress/selection
8	distribution/buyaddress/selection → distribution/address/addressupload
9	distribution/project/templatechoice → distribution/project/projectnew

Abb. 3.6.13: Beispiele der nicht geplanten Links

geben (5.Zeile). Diese Vorschläge müssen sorgfältig analysiert werden und diejenigen, die wirklich sinnvoll erscheinen, in die Sitemap eingearbeitet werden. Die Webdesigner sollten einerseits auf keinen Fall mit vielen Informationen oder weiteren Links die Seiten “füttern“, andererseits müssen sie die Navigation so leicht wie möglich gestalten.

Durch die Analyse der statistischen Daten, der Nutzungsmodelle, der Sitemap, der Abbruchmuster, der nicht geplanten Links und anderen in Kapitel 3.6.1 bis 3.6.3 aufgelisteten Informationen hat der Anbieter die Möglichkeit, die Website zu optimieren bzw. zu reorganisieren. Die entdeckten unerwarteten Ereignisse sollen als Empfehlungen bei der Anpassung der Website an das Benutzerverhalten berücksichtigt werden.

Kapitel 4

Zusammenfassung

4.1 Fazit

Das Ziel dieser Diplomarbeit war, die Nutzung einer Website bzw. Nutzungsverhalten der Besucher anhand der Logfiles zu analysieren. Im Abschnitt 3.4 wurde gezeigt, dass statistische Analyse einen ersten Überblick über die Website-Nutzung bietet. Über die Besucheranzahl in verschiedenen Zeitintervallen kann der Anbieter feststellen, ob die Anzahl der Kunden mit der Zeit wächst, und welche Seiten von meisten User angeschaut werden. Insbesondere kann der Anbieter vergleichen, wie viele Kunden zu einer Website gelangen und wie viele von ihnen ein Produkt bestellen. Aus der Verweilstatistik gewinnt man Information, welche Seiten die Aufmerksamkeit der Benutzer heranziehen, d.h. am längsten angeschaut werden. Durch Einbeziehen von den graphischen Objekten in die Statistik findet man heraus, welche Grafiken besonders lange geladen werden. Oder man kann feststellen, welche Dateien besonders oft herunterladen werden. Mit diesen Informationen kann der Webdesigner erste Verbesserungen der Website vornehmen: ungenutzte Seiten eliminieren oder mit ihnen verlinkte Seiten sowie häufige Abbruchseiten umgestalten.

Um das Verhalten der User zu verstehen, reicht es nicht aus, die Statistiken zu analysieren. Zu diesem Zweck muss der Anbieter die Zugriffspfade der Kunden nachvollziehen können. Um dies zu erreichen, kann er die in Kapitel 2 beschriebenen Verfahren anwenden. Es wurde jedoch gezeigt, dass *Apriori-* und *WINEPI*-Algorithmus zur Analyse von den langen Navigationspfaden wegen der Vielzahl der redundanten Ergebnisse, die praktisch nicht interpretierbar sind, nicht geeignet sind. Deshalb soll der Analyst einen oder besser beide Verfahren, nämlich *Closed Itemsets* und *Interaction Pattern Mining*, heranziehen. Mit diesen Methoden entdeckt man Charakteristika der Besteller und Nichtbesteller

entweder als Mengen gemeinsam besuchten Seiten oder als Sequenzen aufeinander folgenden Seiten. Die gefundenen Charakteristika lassen sich als Nutzungsmodelle von Kunden interpretieren. Für beide Darstellungsarten wurden sehr ähnliche Zugriffsmodelle entdeckt. Da das häufige Vorkommen von vollständigen Pfaden eher unwahrscheinlich ist, wurden Klickfolgen ohne wiederholte Ereignisse (Seiten) analysiert. Mit dem Wissen über häufige Nutzungspfade kann die Website so umgestaltet bzw. umstrukturiert werden, dass nur häufige Pfade (eventuell mit einigen Verbesserungen) in der Website präsent bleiben und Seiten oder geplante Zugriffspfade, die in häufigen Nutzungsmodellen nicht vorkommen, aus der Website entfernt werden. Die Benutzer können entsprechend ihren Nutzungsmodellen (s. Abschnitte 3.5.3.2 und 3.5.4.2) in Gruppen eingeteilt und durch Nutzung dieser Information durch die Website geleitet werden.

Aus den Kapiteln über Charakteristika der Besteller und Nichtbesteller (Abschnitte 3.5.3.1 und 3.5.4.1) und unerwartete Nutzung (Abschnitt 3.6.1) erfährt der Webdesigner, wie man effizient Seiten finden kann, die wahrscheinlich zum Abbruch der Bestellung geführt haben, oder diejenigen, die mehrfach während einer Sitzung besucht wurden. Diese Informationen können aus den Nutzungsmodellen der Nichtbesteller oder auch der von Besteller mit zusätzlichen Daten wie die Häufigkeit und die Verweildauer auf den einzelnen Seiten gewonnen werden. Diese Hinweise sind sehr wichtig für die Website-Betreiber, weil sie sehr bemüht sind, die bereits interessierte Kunden nicht zu verlieren, um immer wieder neue Aufträge zu erhalten. Solche Seiten sollten sorgfältig überprüft und eventuell überarbeitet oder mit zusätzlichen Werkzeugen ausgestattet werden.

Auch die Integration der neuen, früher nicht geplanten Links, wie es im Abschnitt 3.6.3 vorgeschlagen wurde, soll die Navigation durch die Website erleichtern.

Wenn der Dienstanbieter bspw. produktionsabhängige Werbung betreiben möchte, kann er den im Abschnitt 3.5.5 vorgestellten Klassifikationsverfahren nutzen. Die Klassifikation, die mit den aus häufigen Sequenzen gewonnenen Regeln durchgeführt wurde, hat allerdings viel bessere Ergebnisse gezeigt. Man muss jedoch beachten, dass die Fehlerquote unter Umständen relativ hoch sein kann. In solchen Fällen sollte man für die Personen mit hoher Fehlerquote auf die produktionsabhängige Werbung verzichten.

Es wurde auch gezeigt, wie die Nutzung einer Website graphisch dargestellt werden kann (Abschnitt 3.6.2). Die Darbietung der Nutzung einer Website durch einen Graphen ist vor allem aus dem Grund gut

geeignet, weil dadurch die Nutzungsdaten im Kontext des physischen Layouts der Website wiedergegeben werden können. Durch farbige Unterschiede werden z.B. selten/häufig genutzte Seiten und Links sowie wahrscheinliche/weniger wahrscheinliche Links hervorgehoben. Sich wiederholende Seitenzugriffe und kurze Abbruchpfade werden ebenso gut erkannt. Die Verfolgung der langen Navigationspfade ist eher mühsam und deshalb nicht zu empfehlen. Die Analyse solcher Pfade ist z.B. viel einfacher durch die Darstellung der Pfade in einem HTML-Dokument mit der Eingabe bestimmter Filterkriterien wie die Häufigkeit oder die Start- und Endseite, wie es im gleichen Abschnitt vorgestellt wurde.

Die Ergebnisse der Arbeit zeigen, dass geeignete Lernverfahren interessante Daten über die Nutzung einer Website und über die Website selbst liefern können. Informationen über Zugriffspfade der Besteller und Nichtbesteller und über unerwartete Ereignisse wie Abbruchsequenzen, sich wiederholende Seitenzugriffe, selten genutzte Seiten sowie Visualisierung dieser Daten haben die besondere Aufmerksamkeit der Web-Manager geweckt. In diesen Hinweisen sehen Anbieter ein hilfreiches Werkzeug zur Analyse Ihrer Website und Optimierung der anzubietenden Dienste.

Die Schwäche der Diplomarbeit ist vor allem in dem Punkt zu sehen, dass unter den zur Verfügung stehenden Daten nur 401 zur Analyse geeigneten Sitzungen von 45 Benutzern identifiziert werden konnten. Dadurch war die Entscheidung, welche Ereignisse als häufig oder selten zu bewerten sind, sehr stark beeinflusst und hätte für eine größere Datenmenge eventuell zu einer anderen Interpretation geführt.

Des Weiteren wäre es sehr interessant, die Auswirkungen der Empfehlungen nach den Verbesserungsmaßnahmen zu testen. Diese Maßnahmen erfordern jedoch einen gewissen Zeitaufwand, so dass der Vergleich des Nutzungsverhaltens vor und nach der Optimierung ausgeblieben ist.

4.2 Ausblick

Wie bei jedem Wissensentdeckungsprozess ist die Analyse des Nutzungsverhaltens nur ein Teilschritt im Lernprozess. Nach dem Einsatz der optimierten Anwendung müssen die Daten weiter gesammelt und analysiert werden. Nur durch derartigen iterativen Prozess kann die Anwendung den immer steigenden Bedürfnissen der User gerecht werden.

Abgesehen von Umstrukturierungsmaßnahmen können für die bestehende Anwendung Funktionen eingebaut werden, die z.B. auf der Basis der entdeckten Nutzungsmodelle einen User durch die Website weiterleiten oder die Seiten mit dem kundenindividuellen Inhalt dynamisch generieren.

Literaturverzeichnis

1. Agrawal, R., Imelinski, T. und Swami, A. Mining Association Rules Between Sets of Items in Large Databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Washington, D.C., 1993.

2. Agrawal. R. und Srikant, R. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994.

3. Baumgarten, M., Büchner A.G., Anand, S. S., Mulvenna, M.D. und Huges, J. G. User-driven navigation pattern discovery from internet data. In *International ACM Workshop on Web Usage Analysis and User Profiling*, 1999.

4. Bayardo, R., Agrawal, R., und Gunopupulos, D. Constaraint-based rule mining in large, dense databases. In *ICDE-99*, 1998.

5. Berendt, B. Web Usage Mining, site semantics, and the support of navigation. In *Workshop Web Mining for E-Commerce, Challenges and Opprtunities*, Boston, MA, August 2000.

6. Büchner, A.G., Mulvenna, M. Discovering internet marketing intelligence through online analytical web usage mining. *SIGMOD Record*, 1998.

7. Catledge, L., Pitkow, J. Characterizing browsing behaviors on the World Wide Web. Computer Networks and ISDN Systems, 1995.

8. Chen, M.-S., Park, J. S. und Yu, P.S. Data Mining for path traversal patterns in Web environments. In *16th International Conference on Distributed Computing Systems*, May 1996.

9. Cheung, D., Kao, B. und Lee, J. Discovering user access patterns on the world wide Web. In *1st Pacific-Asia Conference on Knowledge Discovery and Data Mining*, February 1997.

10. Chi, E. H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler und Card, S.K. Visualizing the evolution of web ecologies. In *CHI'98*, Los Angeles, 1998.

11. Cohen, E., Krishnamurthy, B. und Rexford, J. Improving end-to-end performance of the web using server volumes and proxy filters. In *Proc. ACM SIGCOMM*, 1998.

12. Cooley, R., Mobasher, B., Srivastava,J. Web Mining: Information and pattern discovery on the World Wide Web . In *International Conference on Tools with Artificial Intelligence*, Newport Beach, CA, 1997.

13. El-Ramly, M., Stroulia, E. und Sorenson, P. Recovering Software Requirements from System-user Interaction Traces, In *Proc. of 14th Int. Conf. on Software Eng. and Knowledge Eng. (SEKE '02)*, ACM Press, Italy, 2002.

14. El-Ramly, M., Stroulia, E. und Sorenson, P. From Run-time Behaivor to Usage Scenarios: An Interaction-Pattern Mining Approach. In *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Edmonton, Canada, July 2002.

15. Godin, R. und Missaoui, R. An Incremental Concept Formation for Learning from Databases. Theoretical Computer Science, 1994.

16. Hansen, M. und Shriver, E. Using navigation data to improve IR functions in the context of web search. *CIKM*, 2001.

17. Harms, S., Deogun, J., Saquer, J. und Tadesse, T. Discovering representative episodal association rules from event sequences using frequent closed episode sets and event constraints. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, San Jose, California, USA, November - Dezember 2001.

18. Joachims, T., Freitag, D., Mitchell, T. Webwatcher: a tour guide for the world wide web. In *15th International Conference on Artificial Intelligence*, Nagoya, Japan 1997.

19. Kaufmann. L., Rousseeuw, P.J. Finding Groups in Data: an Itroduction to Cluster Analysis. John Wiley & Sons, 1990.

20. Kryszliewicz, M. Fast Discovery of Representative Association Rules, in *Proc. RSCTC '98, Lecture Notes in Artificial Intelligence*, vol. 1424 (Springer Verlag 1998).

21. Kryszliewicz, M. Representative Association Rules, in *Proc. PAKDD '98, Lecture Notes in Artificial Intelligence*, vol. 1394 (Springer Verlag 1998).

22. Liebermann, H. Letizia: an agent that assists web browsing. In *Proc. of the 1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.

23. Liu, B., Hsu, W., Ma, Y. Pruning and Summarizing the Discovered Associations. In *ACM SIGKDD'99*, 1999.

24. Masseglia, F., Poncelet, P. und Teisseire, M. Using data Mining techniques on web access logs to dynamically improve hypertext structure. *ACM SigWeb Letters*, Oktober 1999.

25. Mobasher, B., Cooley, R. und Stravistava, J. Creating adaptive web sites through usage-based clustering of urls. In *Knowledge and Data Engineering Workshop*, 1999.

26. Mobasher, B., Jain, N., Han, E. und Stravistava, J. Web Mining: Pattern Discovery from World Wide Transactions, Technical Report TR 96-050, Department of Computer Science, University of Minnesota, 1996.

27. Mobasher, B., Jain, N., Han, E. und Stravistava, J. Web Mining: Pattern Discovery on the World Wide Web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, November 1997.

28. Mannila, H., Toivonen, H. und Verkamo, A.I. Discovery of frequent episodes in event sequences, Department of Computer Science, Series of Publications C, Report C-1997-15.

29. Morik, K., Wrobel, S., und Joachims, T. Maschinelles Lernen. Handbuch der Künstlichen Intelligenz. Addison Wesley, 2000.

30. Nakajima, T., Mizuhara, T., Ohta, M. und Ishikawa, H.: Recommendation System Using User Models based on Web Logs. In *Proc. IPSJ National Convention*, 4W-4, 2001.

31. Nan, N., User Profiling for Web Site Recommendation, Department of Computing Science, University of Alberta, 2002.

32. Nan, N., Stroulia, E. und El-Ramly, M., Understanding Web Usage for Dynamic Web-Site Adaptation: A Case Study. In *Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02)*, Oktober 2002.

33. Ng, R., Lakshmanan, L. V. S., Han, J. , und Pang, A. Exploratory mining and pruning optimizations of constrained associations rules. In

Proc. 1998 ACM SIGMOD Int. Conf. Management of Data. Seattle, Washington, June 1998.

34. Ng, R., Han, J. Efficient and effective clustering method for spatial data mining. In *Proc. Of the 20th VLDB Conference*, Santiago, Chile, 1994.

35. Ngu, D.S.W., Wu, X. SiteHelper: a localized agent that helps incremental exploration of the world wide web. In *6th World Wide Web Conference*, Santa Clara, CA, 1997.

36. Padmanabhan, B. und Tuzhilin, A. A belief-driven method for discovering unexpected patterns. In *4th International Conference on Knowledge Discovery and Data Mining*, New York, 1998.

37. Pasquer, N., Bastide, Y., Taouil, R. und Lakhal, L. Efficient Mining of Association Rules Using Closed Itemsets Lattices. *Information Systems 24*, 1999.

38. Perkowitz, M., Etzioni, O. Adaptive web sites: Automatically synthesizing web pages. In *15th National Conference on Artificial Intelligence*, Madison, WI, 1998.

39. Perkowitz, M., Etzioni, O. Adaptive web sites: Conceptual cluster mining. In *16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.

40. Piroli, P., Pitkow, J., and Rao, R. Silk from a sow's ear: Extracting usable structures from the web. In *Proc. Of 1996 Conference on Human Factors in Conputing Systems (CHI-96)*, Vancouver, British Columbia, Canada, 1996.

41. Savasere, A., Omiecinski, E. und Navathe, S. An efficient algorithm for mining association rules in large databases. In *Proc. Of the 21th VLDD Conference*, Zürich, Switzerland, 1995.

42. Saquer, J., Deogun,S. Using Closed Itemsets for Discovering Representative Association Rules. *12th International Symposium, ISMIS 2000*, Charlotte, NC, USA, October 2000.

43. Srikant, R., Vu, Q., und Agrawal, R. Mining Association Rules with Item Constraints. *KDD-97*, 1997.

44. Spiliopoulou, M: Faulstisch, L.C. Wum: A tool for web utilization analysis. In *EDBT Workshop WebDB'98*, Valencia, Spain, March 1998.

45. Srikant, R., Vu, Q., und Agrawal, R. Mining Sequential Patterns: Generalization and performance improvements. In *Proc. Of the Fifth Int'l Conference on Extending Database Technology*. Avignon, France, 1996.

46. Yan, T. W., Jacobsen, M., Garcia-Molina, H. und Dayal, U. From User Access Patterns to Dynamic Hypertext Linking. In *5th World Wide Web Conference (WWW'96)* Paris, France, May 1996.

47. Wang, X., PagePrompter: An Intelligent Agent for Web Navigation Created Using Data Mining Techniques, Technical Report CS-2000-08, Department of Computing Science, University of Regina, 2000.

48. Wille, R. Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts. In: Ivan Rivali, ed., Ordered sets, 1982.

49. Zaki, M.J. und Hsiao, C-J. CHARM: an Efficient Algorithm for Closed Association Rule Mining. In *Proceedings of the Second SIAM International Conference on Data Mining*, Arlington, VA, 2002. SIAM.