Universität des Saarlandes Naturwissenschaftlich-Technische Fakultät I Fachrichtung Informatik

Regelbasierte Empfehlungsdienste für Kunden im Offline-Handel Basierend auf Automatisch Geplanten Teilsortimenten

Bachelorarbeit

vorgelegt von

Eva Maria Gressung

am 30.09.2015

Betreut durch Christian Felix Bürckert, M.Sc.

Begutachtet von Prof. Dr. Antonio Krüger Dr. Tim Schwartz

Eidesstattliche Erklärung

| Ich erk | däre | hierm | it an Eic | les Sta | tt, dass | ich di | e vorliege | ende | Arbeit se | elbs | tständig |
|----------|------|-------|-----------|---------|----------|--------|------------|------|-----------|------|-----------|
| verfasst | und | keine | anderen | als die | e angege | ebenen | Quellen | und | Hilfsmitt | el v | verwendet |
| | | | | | ha | be. | | | | | |

Statement in Lieu of an Oath

| I hereb | y confirm | that I l | have wri | tten this | thesis or | n my o | own and | that I | have not | used |
|---------|-----------|----------|----------|------------|-----------|---------|----------|---------|----------|------|
| | any othe | er media | or mate | erials tha | n the one | es refe | erred to | in this | thesis. | |

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

|] | agree to | make l | both v | ersions | of my | thesis | (with | a passi | ng gra | ade) a | ccessil | ole to | the |
|---|-----------|--------|--------|---------|--------|---------|-------|---------|--------|---------|---------|----------------|------|
| | public by | having | them | added : | to the | library | of th | e Comp | outer | Science | e Dep | $artm\epsilon$ | ent. |

| Saarbrücken, | | |
|--------------|----------------|---------------------------|
| | (Datum / Date) | (Unterschrift/ Signature) |

Zusammenfassung

Empfehlungsdienste sind im Online-Handel bekannt und weitverbreitet. Sie leiten Kunden durch das große Sortiment des Online-Handels. Vergleichbare Dienste für den Offline-Handel gibt es jedoch keine. Die vorliegende Arbeit löst dieses Problem und entwickelt einen Empfehlungsdienst für den Offline-Handel.

Die meisten Dienste im Online-Handel nutzen kollaboratives oder inhaltsbasiertes Filtern zur Generierung von Empfehlungen. Der naheliegende Weg, diese Filtermethoden für den Offline-Handel zu nutzen, ist nicht umsetzbar. Sowohl das kollaborative als auch das inhaltsbasierte Filtern benötigen Kundeninformationen, die im Offline-Handel nicht zugänglich sind. Daher wird ein regelbasierter Empfehlungsdienst entwickelt, der keine Informationen über das Kaufverhalten der Kunden benötigt. Trotzdem können personalisierte Empfehlungen gegeben werden. Um an der Kasse gegebene Empfehlungen direkt in den Einkaufswagen zu befördern, wird ein Harting Auswurfautomat genutzt. Solch ein Automat hat ein Fassungsvermögen für mehrere Instanzen von wenigen, verschiedenen Produkten, die auf Knopfdruck ausgeworfen werden können. Die Planung des Teilsortiments, das im Automaten gelagert wird, übernimmt ein Planungsalgorithmus für NP-harte Probleme. Um Empfehlungsdienst und Teilsortiment aufeinander abzustimmen werden die gleichen Assoziationen zwischen Produkten für beide Aufgaben verwendet. Diese Assoziationen werden durch den Apriori Algorithmus ermittelt. Dies ist ein Big Data Algorithmus, der aus einer Menge von Transaktionen Assoziationen zwischen deren Objekten findet. Diese Anwendung des Apriori Algorithmus ist neuartig und wurde so noch nicht umgesetzt.

Um das System praktisch zu realisieren, wird eine Schnittstelle entwickelt und implementiert. Die fertige Implementierung wird im Innovative Retail Laboratory (IRL) des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI) integriert. Außerdem sind sowohl die Implementierung des Planungsalgorithmus, als auch die teilweise parallele Implementierung des Apriori Algorithmus, unabhängig nutzbar. Diese können für andere Probleme angepasst werden.

Das System wird mit Originaldaten der GLOBUS SB-Warenhaus Holding Güdingen getestet. Die Tests zeigen, dass der Empfehlungsdienst zu etwa 20% der getesteten Einkäufe mindestens eine Empfehlung gibt, wobei im Durchschnitt zu jedem Einkauf ein Produkt empfohlen wird. Eine Nutzerstudie kann nicht geführt werden, da das System bisher lediglich im IRL umgesetzt ist.

Inhaltsverzeichnis

| 1 | Einf | ührung | 1 |
|---|----------------|---|----------|
| 2 | Ver 2.1 | wandte Arbeiten Empfehlungsdienste im Online-Handel | 5 |
| | 2.2 | Sortimentsplanung im Offline-Handel | |
| 3 | Ver | worfene Ansätze | 9 |
| | 3.1 | Kollaboratives Filtern im Offline-Handel | 9 |
| | 3.2 | Inhaltsbasiertes Filtern im Offline-Handel | 9 |
| | 3.3 | Personalisierte, zeitabhängige Empfehlungen | 10 |
| 4 | Kon | ızept | 11 |
| | 4.1 | Apriori Algorithmus | 11 |
| | 4.2 | Teilsortimentsplanung | |
| | 4.3 | Regelbasierter Empfehlungsdienst | |
| | 4.4 | Filtern | |
| | 4.5 | Schnittstelle RAAsP | 18 |
| 5 | lmp | lementierung | 21 |
| | 5.1 | Apriori Algorithmus | 21 |
| | 5.2 | Planungsalgorithmus | 23 |
| | 5.3 | Implementierung der Schnittstelle RAAsP | 25 |
| | 5.4 | Grafische Benutzeroberfläche | 27 |
| 6 | Eva | luation | 29 |
| | 6.1 | Parameter prüfen | 29 |
| | 6.2 | Empfehlungsdienst testen | 33 |
| 7 | Kon | ıklusion | 35 |
| 8 | Wei | terführende Arbeiten | 39 |
| | 8.1 | Sortimentsebenen | 39 |
| | 8.2 | Allgemeine Empfehlungen | 40 |
| | 8.3 | Anpassen des Teilsortiments | 41 |
| | 8.4 | Produktergänzende Empfehlungen | 42 |
| | 8.5 | Weitere Evaluationen | 43 |
| | 8.6 | Werbemöglichkeiten | 44 |
| 9 | Арр | pendix | 47 |
| | 91 | Evaluationsergebnisse | 47 |

| 9.2 | European Article Number | | • | | | | | | | | 51 |
|----------|-------------------------|------|---|------|--|--|--|--|--|--|----|
| Literatu | ırverzeichnis | | | | | | | | | | 53 |

1 Einführung

Der Online-Handel nimmt stetig zu und es werden immer mehr Produkte und Dienstleistungen über das Internet vertrieben [1]. Daher werden Kunden immer häufiger mit Hilfe von personalisierten Empfehlungen durch das meist sehr große Sortiment eines Online-Handels gelenkt [2]. Mit Empfehlungen wie "Kunden die dieses Produkt gekauft haben, kauften auch jenes Produkt" oder "Das könnte ihnen auch gefallen" werden die Kunden auf Produkte aufmerksam, die sie sonst eventuell nicht betrachtet hätten [3]. Diese Vorteile sind bisher auf den Online-Handel beschränkt. Auch im Offline-Handel gibt es große Sortimente, die schwer auf einen Blick zu erfassen sind. Es stellt sich die Frage, ob und wie sich ein Empfehlungsdienst auch im Offline-Handel umsetzten lässt. Dabei treten zwei wesentliche Probleme im Vergleich mit dem Online-Handel auf.

Anders als im Online-Handel, gibt es im Offline-Handel keine verpflichtenden Kundenprofile. Im Offline-Handel entscheidet sich ein Kunde freiwillig ein Kundenprofil anzulegen. Dies ist meist an Rabattvorteile gekoppelt, wie zum Beispiel das des Payback ¹ Systems. Somit können nur für Kunden mit Kundenprofil vergangene Einkäufe gespeichert werden. Im Online-Handel sind diese jedoch die Grundlage für die meisten Empfehlungsdienste [4]. Ohne Informationen über das Kaufverhalten der Kunden können sowohl das kollaborative, als auch das inhaltsbasierte Filtern, welche die bekanntesten Filtermethoden für Empfehlungsdienste aus dem Online-Handel sind, nicht agieren. Empfehlungsdienste, die auf kollaborativem bzw. inhaltsbasiertem Filtern beruhen, benötigen Produktbewertungen bzw. Kundenpräferenzen, um Empfehlungen geben zu können [3].

Zweitens kann ein empfohlenes Produkt im Offline-Handel nicht einfach durch einen Klick in den Warenkorb befördert werden. Wenn ein Kunde an der Kasse eine Empfehlung für ein Produkt bekommt und sich für dieses entscheidet, müsste er zum entsprechenden Regal laufen, um das Produkt zu holen. Dieser Aufwand kann zu viel sein, wenn der Kunde das Produkt nicht unbedingt braucht, was dazu führt, dass er sich doch gegen das Produkt entscheidet. Eine andere Möglichkeit wäre die Empfehlungen am entsprechenden Regal zu geben, dafür muss der Kunde jedoch an diesem Regal vorbeigehen, wodurch er das Produkt auch ohne Empfehlung sieht.

Die vorliegende Arbeit löst diese zwei Probleme und stellt eine Möglichkeit vor, wie ein Empfehlungsdienst im Offline-Handel umgesetzt werden kann. Die Idee ist es, einen Auswurfautomaten im Kassenbereich des Offline-Handels zu integrieren, sodass empfohlene Produkte direkt ausgeworfen werden können. Solch ein Automat der Firma Harting befindet sich im Innovative Retail Laboratory (IRL) "Das Innovative Retail Laboratory (IRL) ist ein anwendungsnahes Forschungslabor des Deutschen Forschungszentrums für Künstliche Intelligenz² (DFKI), das in der Zentrale der GLOBUS SB-Warenhaus

¹https://www.payback.de/

²http://www.dfki.de/web

Holding³ in St. Wendel eingerichtet ist."⁴ In diesem Labor sind alle Produkte mit einem RFID-Funketikett versehen. Der Kunde geht an der Kasse mit seinem Wagen durch eine RFID-Schleuse und die Produkte werden registriert, ohne dass sie aus dem Wagen genommen werden müssen. Anhand dieser registrierten Produkte soll ein Empfehlungsdienst passende Produkte empfehlen, die der Kunde vor dem bargeldlosen Bezahlen zu seinem Einkauf hinzufügen kann. Die empfohlenen Produkte sollen per Knopfdruck aus dem Harting Automaten ausgeworfen und der zusätzliche Preis auf die Kaufsumme addiert werden.

Ein Anspruch an die Arbeit ist, dass die Kunden sich nicht für den Empfehlungsdienst anmelden müssen und kein Kundenprofil besitzen. Außerdem soll sowohl auf Kundenals auch auf Händlerseite der zusätzliche Aufwand möglichst gering gehalten werden. Die genauen Ziele werden im Folgenden beschrieben.

Zielsetzung

Das Ziel der vorliegenden Arbeit ist die Entwicklung eines Empfehlungsdienstes für den Offline-Handel. Dieser soll ohne Informationen über den einzelnen Kunden Empfehlungen aus einem Teilsortiment geben. Das Teilsortiment besteht aus wenigen Produkten, da es in einen Harting Auswurfautomaten passen soll. Die Anzahl der Produkte ergibt sich in der vorliegenden Arbeit aus dem Fassungsvermögen des Harting Automaten, der im IRL im Kassenbereich integriert ist. Da die Anzahl und Größe der Fächer individuell angepasst werden kann, variiert die Größe des Teilsortiments. Für die vorliegende Arbeit wird der Harting Automat in 20 Fächer unterteilt. Somit enthält der Automat mehrere Instanzen dieser 20 verschiedenen Produkte. Diese können auf Knopfdruck in den Einkaufswagen der Kunden befördert werden. So wird das in Abschnitt 1 genannte Problem, die Produkte wie im Online-Handel per Klick in den Warenkorb zu befördern, gelöst.

Die Arbeit lässt sich in zwei wesentliche Teile untergliedern. Die erste Aufgabe ist, aus einem großen Sortiment ein Teilsortiment auszuwählen, auf dem der Empfehlungsdienst agiert. Darüber hinaus muss das Sortiment an den Harting Automaten angepasst werden. Um dieses NP-harte Problem der Auswahl der Produkte zu lösen, wird ein Planungsalgorithmus genutzt. Dieser wählt auf Grund der Größe der Produkte und der Fächer des Automaten ein Sortiment aus. Außerdem soll die Häufigkeit der Produkte mit in die Planung einfließen. Die Häufigkeit der Produkte wird aus vergangen Einkäufen ermittelt. Je häufiger ein Produkt in der Vergangenheit gekauft wurde, desto wahrscheinlicher ist es, dass es auch in Zukunft gekauft wird. Darüber hinaus werden Assoziationen zwischen Produkten betrachtet, um das Sortiment auszuwählen. Diese sind besonders wichtig, um dem Empfehlungsdienst eine gute Grundlage zu bieten.

Das führt zur zweiten Aufgabe, einen Empfehlungsdienst zu entwickeln. Dieser soll Empfehlungen aus dem Teilsortiment geben und keinerlei Kundeninformationen benötigen. Einige naheliegende Ansätze aus dem Online-Handel werden in Kapitel 3 betrachtet und verworfen. Die bekanntesten Filtermethoden für Empfehlungsdienste im Online-Handel benötigen Kundeninformationen, die im Offline-Handel nicht zugänglich

³http://www.globus.de/de/home.html

⁴http://www.innovative-retail.de/

sind. Stattdessen wird ein regelbasierter Empfehlungsdienst entworfen, der keinerlei Kundeninformation benötigt, um Empfehlungen geben zu können. Dieser umgeht das Problem der fehlenden Informationen über das Kaufverhalten der Kunden.

Ein regelbasierter Empfehlungsdienst benötigt eine Menge von Regeln, die ihm Wissen über Assoziationen zwischen Produkten geben. Mit deren Hilfe werden für einen Einkaufswageninhalt passende Empfehlungen generiert. Damit der regelbasierte Empfehlungsdienst viele Empfehlungen aus dem geplanten Teilsortiment geben kann, werden die gleichen Assoziationen zwischen Produkten für den Empfehlungsdienst verwendet, die auch mit in die Planung des Sortiments einbezogen werden. Da das Finden und Warten von Assoziationen ein großer Nachteil eines regelbasierten Empfehlungsdienstes ist [5], sollen diese automatisch ermittelt werden. Hierfür wird der Apriori Algorithmus verwendet. Dies ist ein Big Data Algorithmus, der 1994 zur Datenanalyse von Rakesh Agrawal und seinem Team [6] entwickelt wurde. Seine Relevanz ist erkennbar durch den hohen Forschungsaufwand, der betrieben wird, um die Laufzeit des Algorithmus zu verbessern [7] [8] [9]. Dieser Algorithmus ermittelt aus großen Transaktionsmengen häufig auftretende Objektmengen und Assoziationsregeln, die Verbindungen zwischen Objekten repräsentieren [7]. Diese Assoziationsregeln dienen dem Empfehlungsdienst als Wissensbasis, mit der er Empfehlungen geben kann.

Um die Teilsortimentsplanung sowie den regelbasierten Empfehlungsdienst umzusetzen, wird eine Schnittstelle entwickelt. Des Weiteren wird die Schnittstelle implementiert und im IRL integriert, um den dort stehenden Harting Automaten in Betrieb zu nehmen. Die Implementierung soll außerdem eine grafische Benutzeroberfläche haben, über die den Kunden Empfehlungen gegeben werden, die sie annehmen oder ablehnen können.

Die Evaluation des Systems geschieht nicht durch eine Nutzerstudie, da das System im Rahmen der vorliegenden Arbeit nur im IRL praktisch umsetzbar ist. Hier stellen sich jedoch zwei Probleme. Zum Einen gibt es im IRL nur eine kleine Auswahl von 1737 Produkten, zum Anderen sind die Einkäufe im Labor gestellt und entsprechen somit keinen realen Einkäufen. Außerdem ist der Aufwand genug "Einkäufe" für einen Big Data Algorithmus aufzuzeichnen zu groß. Auch zufällig generierte Einkäufe geben keine Information über das reale Kaufverhalten der Kunden. Es stehen jedoch Originaldaten über vergangene Einkäufe in einer Zeitspanne von vier Monaten aus der GLOBUS SB-Warenhaus Holding Güdingen zur Verfügung. Diese werden genutzt, um die optimalen Parameter zu ermitteln, mit denen der Apriori Algorithmus initialisiert werden sollte. Um den Empfehlungsdienst zu testen, werden einige der Einkaufszettel genutzt, um ein Teilsortiment zu planen. Mit einer Menge anderer Einkaufszettel wird anschließend ermittelt, in wie viel Prozent der Käufe der Empfehlungsdienst Empfehlungen geben kann.

Es wird Wert darauf gelegt, dass keinerlei Kundendaten benötigt oder gespeichert werden. Dies ist wichtig, damit der Empfehlungsdienst auch für Neukunden nützlich ist und keine Anmeldung erfordert. Ebenso wichtig ist, dass die Kunden keinen Aufwand betreiben müssen, wie zum Beispiel Produkte bewerten, um Empfehlungen zu bekommen.

2 Verwandte Arbeiten

Im Folgenden werden einige verwandte Arbeiten vorgestellt. Sie sind unterteilt in Empfehlungsdienste im Online-Handel und Sortimentsplanung im Offline-Handel. Die Arbeiten bieten einige Ansätze die Aufgaben der vorliegenden Arbeit umzusetzen. In Kapitel 3 werden mögliche Ideen zur Umsetzung eines Empfehlungsdienstes im Offline-Handel beschrieben. Grundlage dafür sind die in Abschnitt 2.1 vorgestellten Empfehlungsdienste aus dem Online-Handel.

2.1 Empfehlungsdienste im Online-Handel

Hier werden zwei der gängigsten Filtermethoden für Empfehlungsdienste im Online-Handel [10] und ihre Anwendungen vorgestellt. Zum besseren Verständnis wird jeweils ein Beispiel gegeben. In dem Artikel "Unifying Collaborative and Content-Based Filtering" von Justin Basilico und Thomas Hofmann [11] wird ein Ansatz beschrieben, der beide Filtermethoden kombiniert, um bessere Prognosen für Empfehlungen geben zu können.

Kollaboratives Filtern

Das kollaborative Filtern (engl. collaborative filtering) basiert auf Produktbewertungen durch Nutzer. Diese können auf verschiedenen Arten angegeben werden, z.B. nutzt das soziale Netzwerk Facebook ¹ ein einfaches "Gefällt mir", wohingegen der Online-Versandhändler Amazon ² ein fünf Sterne System anbietet [1]. Die Bewertungen der Nutzer werden in Kundenprofilen gespeichert. Anhand dieser gespeicherten Bewertungen werden nun Korrelationen zwischen Nutzern ermittelt. Dabei gilt, Nutzer, die gleiche Produkte gleich bewerten, haben eine höhere Korrelation als Nutzer, die gleiche Produkte unterschiedlich bewerten [10]. In dem Artikel "A Comparative Study of Collaborative Filtering Algorithms" von Joonseok Lee [12] werden einige verschiedene Ansätze zur Berechnung und Bewertung der Korrelation beschrieben und verglichen.

Werden zwei Nutzer A und B als stark korrelierend identifiziert, so können aufbauend auf diesem Wissen Empfehlungen gegeben werden. Bei stark korrelierenden Nutzern ist es wahrscheinlicher, dass Objekte, die nur von einem der beiden Nutzern bewertet wurden, von dem anderen Nutzer ähnlich bewertet werden [13]. Somit werden diese Objekte, wenn sie z.B. von Nutzer A gut bewertet werden, Nutzer B empfohlen. Es besteht eine hohe Chance, dass Nutzer B sie auch gut bewertet.

¹https://www.facebook.com/

²http://www.amazon.de/

Beispiel

In diesem Beispiel gibt es vier Nutzer und vier Objekte, für die sie Bewertungen abgeben wie sie in Tabelle 2.1 dargestellt sind. Bewertet wird mit einem binären System, wobei "+" für gut und "-" für schlecht steht. Alice, Bob und Carl haben alle vier Objekte bewertet, wohingegen Dora Objekt 4 noch nicht bewertet hat. Es stellt sich nun die Frage, ob es sinnvoll ist Dora Objekt 4 zu empfehlen oder nicht. Betrachtet man alle Bewertungen, so wird deutlich, dass Alice und Dora Objekt 1-3 gleich bewertet haben. Das bedeutet Alice und Dora korrelieren stark miteinander. Da Alice das fehlende Objekt 4 als gut bewertet hat, wird es Dora empfohlen.

| Nutzer | Objekt 1 | Objekt 2 | Objekt 3 | Objekt 4 | | | |
|--------|----------|----------|----------|----------|--|--|--|
| Alice | + | + | + + | | | | |
| Bob | = | + | + | = | | | |
| Carl | = | - | + | + | | | |
| Dora | + | + | + | ? | | | |

Tabelle 2.1: Exemplarische Produktbewertungen durch Kunden, als Grundlage für kollaboratives Filtern

Inhaltsbasiertes Filtern

Beim Inhaltsbasierten Filtern (enl. content-based filtering) werden Nutzern auf Grund ihrer Präferenzen weitere Objekte empfohlen. Diese Präferenzen werden durch explizite oder implizite Bewertung von Objekten ermittelt. Explizit bedeutet, dass der Nutzer aktiv ein Objekt bewertet, wohingegen implizit heißt, dass die Präferenzen eines Nutzers an Hand seines Verhaltens ermittelt werden. Dies kann z.B. daraus abgeleitet werden, wie lange sich ein Nutzer auf einer bestimmten Website aufhält oder welche Produkte er sich anschaut. Abhängig von den Nutzerpräferenzen werden Objekte ermittelt und empfohlen, die diesen Präferenzen entsprechen [14]. Um dies zu bewerkstelligen ist es notwendig Ähnlichkeiten zwischen Objekten ermitteln zu können bzw. sie mit den Präferenzen identifizieren zu können. Dazu werden Metadaten genutzt, die den Inhalt der Objekte repräsentieren. Das inhaltsbasierte Filtern wird häufig bei Dokumenten oder Internetseiten angewandt, deren Inhalt durch häufig auftretende Worte gut repräsentiert wird [10]. Um die Metadaten zu vergleichen, wird häufig ein Vektorraummodell genutzt, bei dem Objekte durch Vektoren repräsentiert werden [15].

Beispiel

In diesem Beispiel gibt es vier Objekte, die jeweils mit Schlagworten versehen sind, wie in Tabelle 2.2 dargestellt. Dabei bedeutet "+", dass das Objekt für Nutzer mit der entsprechenden Präferenz interessant ist und "-" bedeutet, dass es nicht interessant ist. Aus diesen Objekten soll für einen Nutzer mit den Präferenzen "Garten, Auto" ein passendes Objekt empfohlen werden. Auf Grund der Präferenzen des Nutzers wird Objekt 2 als interessant ermittelt, da es mit den passenden Metadaten annotiert ist. Dieses Objekt wird dem Nutzer empfohlen.

| Objekt | Garten | Basteln | Auto | Kunst |
|----------|---------|---------|------|-------|
| Objekt 1 | - | - | + | - |
| Objekt 2 | + | - | + | - |
| Objekt 3 | 3 - + - | | | + |
| Objekt 4 | + | + | - | - |

Tabelle 2.2: Exemplarische Metadaten für Objekte, als Grundlage für inhaltsbasiertes Filtern

2.2 Sortimentsplanung im Offline-Handel

Ein grundlegendes Problem im Offline-Handel ist die Sortimentsplanung. Diese wird unterteilt in strategische und operative Sortimentsplanung. Erstere beschäftigt sich mit der Auswahl der Produkte für das Sortiment, wobei Letztere dafür zuständig ist diese Produkte optimal zu platzieren [16]. Optimal bedeutet, dass durch die Platzierung der größtmögliche Gewinn erzielt wird.

Für die Planung des Sortiments ist es sinnvoll es zunächst über eine hierarchische Ordnung zu definieren. Hierzu hat sich der Begriff der Sortimentspyramide etabliert [17]. Abbildung 2.1 zeigt eine Mögliche Sortimentspyramide.

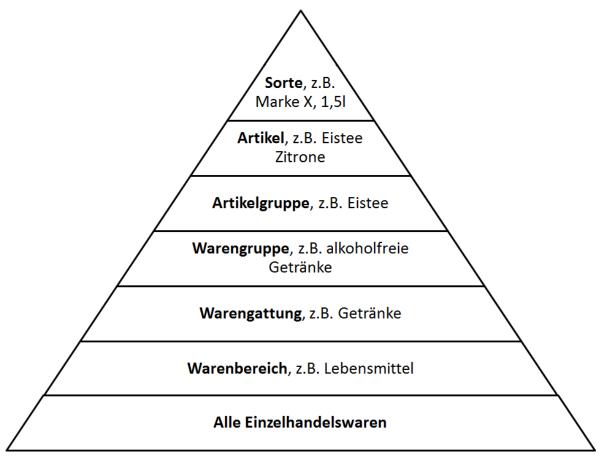


Abbildung 2.1: Sortimentspyramide zur hierarchischen Strukturierung des Sortiments (nach dem Buch "Handelsmanagement"[18])

Die dargestellte Sortimentspyramide nach dem Buch "Handelsmanagement" von Joachim Zentes, Bernhard Swoboda und Thomas Foscht [18] unterteilt sich in mehrere Sortimentsebenen. Diese spezifizieren ein Produkt von unten nach oben mit jeder Ebene genauer. Dabei ist "Sorte" die genaueste Spezifikation und entspricht den Produkten, die im Handel angeboten werden [18].

Strategische Sortimentsplanung

Die strategische Sortimentsplanung hat die Aufgabe das Sortiment zu definieren, also festzulegen welche Produkte im Handel angeboten werden. Jedoch betrachtet die strategische Sortimentsplanung nicht die Sortimentsebene "Sorte", sondern agiert meist auf der allgemeineren Sortimentsebene "Warengruppe" [17]. In dem Buch "Handelsmanagement" [18] sind einige Faktoren beschrieben, die in die strategische Sortimentsplanung einfließen. Durch eine geeignete Sortimentspolitik, die den Anforderungen des Händlers entspricht, wird definiert, wie das Sortiment aussehen soll. Ein Kriterium ist die Sortimentsdimension. Diese bestimmt Tiefe und Breite des Sortiments. Dabei gibt die Anzahl der Sorten die Tiefe des Sortiments an und die Anzahl der Warenarten die Breite des Sortiments [18].

Operative Sortimentsplanung

Für operative Sortimentsplanung sind neben Größe und Platzierung der Produkte auch Verbundbeziehungen zwischen Produkten notwendig, um eine optimale Belegung der Regale zu finden. Verbundbeziehungen können zum Beispiel mit Hilfe der durch den Apriori Algorithmus gefundenen Assoziationsregeln, die in Abschnitt 4.1 beschrieben sind, bestimmt werden. Sie treffen Aussagen wie zum Beispiel "Kunden die Brot kaufen, kaufen in 80% der Fälle auch Käse" [19]. Hierbei ist "Brot" die Prämisse der Regel und "Käse" die Konklusion, 80% ist der Wert der Konfidenz, welcher in Abschnitt 4.1 genauer definiert wird. Prämisse und Konklusion können jeweils aus ein oder mehreren Produkten bestehen. Werden die Produkte aus der Prämisse einer Assoziationsregel im Regal platziert, so sollten bei hoher Konfidenz auch die Produkte aus der Konklusion platziert werden.

Die operative Sortimentsplanung agiert auf den oberen Sortimentebenen "Artikel", sowie "Sorte" [17].

3 Verworfene Ansätze

In diesem Kapitel werden einige mögliche Ansätze zur Umsetzung eines Empfehlungsdienstes im Offline-Handel vorgestellt. Grundlage für die Ideen sind unter anderem die in Abschnitt 2.1 vorgestellten Empfehlungsdienste aus dem Online-Handel. Es wird argumentiert, warum die zwei bekanntesten Filtermethoden für Empfehlungsdienste im Online-Handel [10] im Offline-Handel schwer umsetzbar sind.

3.1 Kollaboratives Filtern im Offline-Handel

Für die Anwendung im Offline-Handel, die in der vorliegenden Arbeit gefordert wird, ist das kollaborative Filtern nicht geeignet. Die wichtigste Grundlage für das kollaborative Filtern ist die Bewertung der Objekte durch Kunden. Bei häufigen Einkäufen von vielen Produkten, wie zum Beispiel Lebensmitteln, erscheint es unwahrscheinlich, dass Kunden die Produkte gewissenhaft bewerten. Hinzu kommt die Frage, wie die Kunden die Produkte bewerten können. Eine Option ist ein Online-Portal, welches für jeden Kunde einen Profil hat, in dem der Kunde seine Einkäufe bewerten kann. Jedoch bleibt das Problem, dass der Kunde nicht alle Produkte zeitnah benutzt (z.B. Insektenspray für den nächsten Urlaub) oder die Produkte gewisse Schwankungen aufweisen (z.B. Obst und Gemüse). Des Weiteren kaufen Kunden im Offline-Handel mehrfach im Monat viele Produkte ein, welche sie alle einzeln bewerten müssen. Der zeitliche Aufwand steht in keinem Verhältnis zu dem gebotenen Vorteil des Empfehlungsdienstes. Aus diesen und weiteren Gründen ist eine explizite Bewertung der Produkte nicht umsetzbar. Auch eine implizite Bewertung ist schwierig umzusetzen. Zwar kann die Häufigkeit der Käufe als Bewertung interpretiert werden, jedoch muss dazu ein Kundenprofil angelegt werden, um die Einkäufe personenbezogen zu speichern. Die vorliegende Arbeit vermeidet es Kundendaten zu speichern. Die Kunden sollen ohne Aufwand und Preisgabe ihrer persönlichen Daten die Vorteile eines personalisierten Empfehlungsdienstes haben. Ein weiteres Problem entsteht für Neukunden. Das sogenannte New-User Problem. Dies bedeutet, dass von einem Neukunden noch keine oder nicht genügend Bewertungen gespeichert wurden und somit keine Korrelationen und Empfehlungen gefunden werden können [20]. In dem Artikel "Getting to Know You: Learning New User Preferences in Recommender Systems" von Al Mamunur Rashid [21] wird ein Ansatz zur Behebung dieses Problems gegeben, welcher jedoch auf Produktbewertungen durch Kunden basiert und somit im Rahmen der vorliegenden Arbeit nicht umsetzbar ist.

3.2 Inhaltsbasiertes Filtern im Offline-Handel

Ein Vorteil des inhaltsbasierten Filterns gegenüber dem kollaborativen Filtern ist die Tatsache, dass keine Produktbewertungen durch Kunden benötigt werden. Um durch inhaltsbasiertes Filtern Empfehlungen zu generieren, werden die Präferenzen des Kunden

benötigt [22]. Diese Präferenzen können explizit oder implizit gesammelt werden [22]. Explizit bedeutet, dass die Kunden ihre Präferenzen angeben oder gekaufte Produkte bewerten. Dieser Ansatz ist jedoch, aus den in Abschnitt 3.1 genannten Gründen, nicht umsetzbar. Hier bietet sich das implizite Sammeln von Kundenpräferenzen an. Dies geschieht durch die Analyse des Verhaltens der Kunden. Im Offline-Handel kann der aktuelle Einkauf des Kunden als Präferenz interpretiert werden. Dies ist offensichtlich, da der Kunde eindeutiges Interesse an diesen Objekten zeigt.

Es tritt jedoch ein neues Problem auf. Die vorhandenen Produkte haben keine Schlagworte oder Metadaten, durch die sie Präferenzen zugeordnet werden können. Im Online-Handel wird das inhaltsbasierte Filtern meist auf Dokumenten verwendet, da hier häufig auftretende Worte den Inhalt des Dokuments gut repräsentieren [22]. Diese Metadaten lassen sich zwar von Hand hinzufügen, bei einem großen Sortiment ist dies jedoch sehr viel Aufwand auf Seiten des Händlers. Des Weiteren wären die Metadaten subjektiv gewählt und repräsentieren eventuell die Objekte nicht ausreichend. So kann eine Schere zum Beispiel das Schlagwort "basteln" enthalten, ebenso aber auch das Schlagwort "Küche". Dies ist abhängig davon, wie die Personen, welche die Metadaten festlegen, die jeweiligen Produkte sehen.

Ein weiteres Problem bildet die Art der Empfehlungen, diese werden nach Kundenvorlieben gegeben. Hat ein Kunde keine Vorliebe für eine Kategorie gezeigt, so werden ihm selten oder nie Produkte aus dieser Kategorie vorgeschlagen, obwohl sie ihn eventuell interessieren [5]. Aus diesen Gründen ist das inhaltsbasierte Filtern für einen Empfehlungsdienst im Offline-Handel nicht geeignet.

3.3 Personalisierte, zeitabhängige Empfehlungen

Eine weitere Idee ist nicht nur ein Empfehlungsdienst, sondern eine Art Einkaufshilfe anzubieten. Der Gedanke ist, ein System zu entwickeln, das aus vergangenen Einkäufen der Kunden ermittelt, in welchem zeitlichen Abstand welche Produkte gekauft werden. Mit Hilfe dieser Informationen kann das System jedem Kunden persönlich beim Einkauf helfen, indem es ihn an eventuell vergessenen Produkte erinnert. Kauft ein Kunde zum Beispiel im Abstand von zwei Wochen regelmäßig Klopapier, dann merkt sich das System zum einen welches Klopapier der Kunde bevorzugt kauft und wann der nächste Kauf bevorstehen wird. Sollte der Kunde zum errechneten Kaufdatum (± einige Tage) kein Klopapier kaufen, so wird ihm seine präferierte Marke vorgeschlagen.

Dieser Ansatz würde dem Kunden den Einkauf von grundlegenden Produkten erleichtern, sodass dieser sich besser auf den Rest des Einkaufs konzentrieren kann. Jedoch tritt auch hier wieder das Problem der Kundenprofile auf. Ohne Informationen über den Kunden und seine vergangen Einkäufe ist dieser Ansatz nicht umsetzbar. Gäbe es jedoch diese Kundenprofile, in denen vergangene Einkäufe gespeichert werden, so müsste der Kunde seinem Profil zugeordnet werden können. Dafür bieten sich Kundenkarten an, die an der Kasse mit gescannt werden. Wird jedoch erst an der Kasse eine Empfehlung bzw. Erinnerung generiert, so tritt erneut das Problem auf, dass der Kunde zum Regal zurücklaufen müsste. Das Sortiment eines Harting Automaten ist zu klein, um für jeden Kunden die persönlichen Lieblingsprodukte zu enthalten. Des Weiteren bräuchte das System einige Zeit, bis sich eine Regelmäßigkeit in den Käufen erkennen lässt, falls eine solche überhaupt existiert.

4 Konzept

In diesem Kapitel werden die Aufgaben der vorliegenden Arbeit konkretisiert. Des Weiteren wird beschrieben, wie sie in der Schnittstelle RAAsP ("Recommendation And Assortment Planning") umgesetzt werden. In Abschnitt 4.1 wird der Apriori Algorithmus erklärt, ohne den die Arbeit schwer umsetzbar ist. Er wird unterteilt in das Finden von häufigen Mengen und die Erstellung von Assoziationsregeln. Diese sind die Basis für die zwei folgenden Abschnitte der Teilsortimentsplanung (Abschnitt 4.2) und des regelbasierten Empfehlungsdienstes (Abschnitt 4.3), welche für den Name RAAsP verantwortlich sind. Im Anschluss wird die Schnittstelle RAAsP genauer betrachtet und erläutert, wie die Anforderungen aus den vorangegangenen Abschnitten umgesetzt werden.

4.1 Apriori Algorithmus

Der Apriori Algorithmus ist ein Big Data Algorithmus, der zur Analyse von großen Datenmengen genutzt wird. Aus großen Mengen von Transaktionen und Mengen von Objekten bestimmt der Algorithmus Teilmengen von Objekten, die häufig zusammen in den Transaktionen auftreten. Aus diesen häufigen Mengen werden im Anschluss Assoziationsregeln gebildet, die Auskunft darüber geben mit welcher Wahrscheinlichkeit diese Objekte abhängig von einander auftreten.

Der Apriori Algorithmus wird häufig zur Warenkorbanalyse genutzt [23]. Dabei wird ermittelt, welche Produkte häufig zusammen gekauft werden. Eine gefundene Assoziation kann zum Beispiel aussagen: "Wenn Müsli gekauft wird, wird in 65% auch Milch dazu gekauft. Die beiden Produkte werden in 3% aller Einkäufe zusammen gekauft.".

Auch in der vorliegenden Arbeit werden Einkäufe analysiert. Hierfür werden dem Algorithmus vergangene Einkäufe übergeben. Diese entsprechen den Transaktionen. Die einzelnen Objekte der Transaktionen sind in diesem Fall die gekauften Produkte. Die so ermittelten Assoziationen geben Auskunft über das Kaufverhalten der Kunden, bzw. welche Produkte häufig in Kombination gekauft werden. Bisher wurden diese Assoziationen genutzt, um Sonderangebote zu bestimmen oder eine optimale Platzierung von Produkten zu finden [24]. Die vorliegende Arbeit nutzt den Apriori Algorithmus für eine neue Idee. Die Assoziationen dienen als Grundlage für einen regelbasierten Empfehlungsdienst. Darüber hinaus werden die Assoziationen genutzt, um ein Teilsortiment auszuwählen, das die Basis für den Empfehlungsdienst bildet.

Begriffe

Hier werden einige Begriffe erläutert, die wichtig sind, um den Apriori Algorithmus zu verstehen.

- Transaktion Eine Transaktion ist eine Menge aus Objekten. Sie ist eindeutig identifizierbar, ebenso wie die Objekte selbst.
- **k-Objektmenge** Eine k-Objektmenge ist eine Menge, die aus k Objekten besteht. So besteht z.B. eine 1-Objektmenge aus einem Objekt.
- Support Der Support bezeichnet die Wahrscheinlichkeit, mit der eine k-Objektmenge in einer Menge von Transaktionen auftritt. Er wird ermittelt aus der absoluten Häufigkeit der k-Objektmenge dividiert durch die Anzahl der Transaktionen.
- Minimaler Support Der minimale Support ist eine, durch den Nutzer definierte, untere Schranke für den Support.
- häufig Eine k-Objektmenge ist häufig, wenn ihr Support über dem minimalen Support liegt.
- Konfidenz Die Konfidenz bezeichnet die Wahrscheinlichkeit, dass eine Assoziationsregel zutrifft. Sie wird berechnet durch die bedingte Wahrscheinlichkeit zwischen Prämisse und Konklusion der Regel.
- Minimale Konfidenz Die minimale Konfidenz ist eine, durch den Nutzer definierte, untere Schranke für die Konfidenz.

Finden der häufigen Mengen

Der Apriori Algorithmus arbeitet auf einer Menge $T = \{T_1, T_2, T_3, ..., T_n\}$ von Transaktionen und einer Menge $I = \{I_1, I_2, I_3, ..., I_m\}$ von Objekten. Zuerst wird eine Tabelle C_1 generiert, mit Kandidaten für häufige 1-Objektmengen. Diese Tabelle enthält eine Zeile für jedes Objekt $I_j \in I$. Zu jedem Objekt I_j werden die Transaktionen gespeichert, in denen es vorkommt. Die Anzahl dieser Transaktionen dividiert durch die Anzahl aller Transaktionen ergibt den Support, der ebenfalls gespeichert wird. Aus dieser Tabelle C_1 von Kandidaten kann nun eine Tabelle L_1 der häufigen 1-Objektmengen ermittelt werden. Das bedeutet, in Tabelle L_1 sind alle 1-Objektmengen gespeichert, die häufiger als der angegebene minimale Support in der Menge der Transaktionen vorkommen. Diese werden durch einen einfachen Vergleich gefunden.

Anschießend werden iterativ für alle weiteren n die Kandidatenmengen C_n ermittelt. Dabei ist zu beachten, dass eine n-Objektmenge nur häufig sein kann, wenn alle ihre Teilmengen häufig sind [6]. Daher werden k+1-Objektmengen aus häufigen k-Objektmengen gebildet. Dabei werden jeweils zwei k-Objektmengen zusammengefasst, die sich in k-1 Objekten überschneiden. Anschließend wird die absolute Häufigkeit der k+1-Objektmengen ermittelt. Dies geschieht jedoch nicht, indem über alle Transaktionen iteriert wird, sondern nur über die Transaktionen, in denen eine der beiden k-Objektmengen vorkommt. So kann die Anzahl der zu betrachtenden Transaktionen reduziert werden. Um die Menge weiter zu reduzieren, werden die Transaktionen der k-Objektmenge mit dem kleineren Support betrachtet. Aus dieser Menge an Transaktionen werden diejenigen Transaktionen ermittelt, in denen die k+1-Objektmenge vorkommt. Daraus entsteht der Support, welcher zusammen mit den Transaktionen in C_{k+1} gespeichert wird. Abschließend werden die häufigen k+1-Objektmengen, deren Support über dem angegebenen minimalen Support liegt, in L_{k+1} übertragen.

Dieser Vorgang wird ausgeführt, bis keine weiteren häufigen Mengen mehr gefunden werden. [7]

Bestimmen der Assoziationsregeln

Assoziationsregeln sind Implikationsregeln, die aus einer Prämisse P und einer Konklusion K bestehen. Es gilt

$$P \cap K = \emptyset$$
 und $P, K \subseteq I$.

Das bedeutet, dass Prämisse und Konklusion disjunkt sind und jeweils Teilmengen der Objektmenge I. Weiterhin gilt, dass $P \cup K$ eine häufige Menge sein muss. Eine Regel beinhaltet somit alle Objekte einer häufigen Menge. Um eine Assoziationsregel $P \to K$ zu bewerten, gibt es zwei Größen.

$$support(P \to K) = support(P \cup K)$$
 (4.1)

$$confidence(P \to K) = \frac{support(P \cup K)}{support(P)}$$
 (4.2)

Zum einen den Support (vgl. Gleichung 4.1) als ein Maß für die statistische Signifikanz [25]. Er sagt aus, dass eine häufige Menge, aus der die Assoziationsregel besteht, in s% der Transaktionen vorkommt. Zum anderen gibt es die Konfidenz (vgl. Gleichung 4.2, die ein Maß für die Stärke einer Regel ist. Sie gibt an, in wie viel Prozent der Fälle die Anwesenheit der Prämisse P in einer Transaktion die Anwesenheit der Konklusion K determiniert.

Aus jeder häufigen Menge lassen sich mehrere Assoziationsregeln ableiten. Da Assoziationsregeln nicht zwangsläufig symmetrisch sind, lassen sich aus der kleinst möglichen häufigen Menge mit zwei Objekten bereits zwei Regeln ableiten. Diese haben den selben Support, können aber unterschiedliche Konfidenzen haben. Ein Beispiel dafür ist der Kauf einer Kamera, der mit hoher Wahrscheinlichkeit den Kauf von Batterien mit sich zieht, andersherum jedoch nicht [26].

Aus den generierten Regeln werden die wichtigen Regeln ermittelt, indem deren Konfidenz mit einer minimalen Konfidenz verglichen wird. Liegt die Konfidenz einer Regel unterhalb der minimalen Konfidenz, so hat sie nicht genug Aussagekraft, um für die weiteren Berechnungen in Betracht gezogen zu werden.

Beispiel

Um den im vorangegangenen Abschnitt erklärten Apriori Algorithmus zu veranschaulichen, folgt ein Beispiel.

Tabelle 4.1 zeigt eine Menge von Transaktionen aus der häufige Objektmengen ermittelt werden sollen.

Das daraus resultierende Ergebnis C_1 wird in Tabelle 4.2 gezeigt. Sie zeigt zu jeder 1-Objektmenge die Transaktionen, in denen die Menge vorkommt und den Support. Für einen besseren Überblick wird der Support als absolute Häufigkeit angegeben. Um eine Wahrscheinlichkeit im Intervall [0, 1] zu erhalten, muss die angegebene Häufigkeit durch die Anzahl der Transaktionen (hier 7) dividiert werden. Unter Annahme eines

| T_i | Objekte |
|-------|-----------------|
| T_1 | I_1, I_3, I_4 |
| T_2 | I_1, I_3, I_4 |
| T_3 | I_1, I_3 |
| T_4 | I_1, I_2 |
| T_5 | I_2, I_3 |
| T_6 | I_2, I_3 |
| T_7 | I_2 |

Tabelle 4.1: Menge von Transaktionen, aus denen der Apriori Algorithmus häufige k-Objektmengen ermittelt

| Objekt | Support | Transaktionen |
|-----------|---------|---------------------------|
| $\{I_1\}$ | 4 | T_1, T_2, T_3, T_4 |
| $\{I_2\}$ | 4 | T_4, T_5, T_6, T_7 |
| $\{I_3\}$ | 5 | T_1, T_2, T_3, T_5, T_6 |
| $\{I_4\}$ | 2 | T_1, T_2 |

Tabelle 4.2: C_1 , Tabelle der Kandidaten für häufige 1-Objektmengen, mit absoluter Häufigkeit als Support

minimalen Supports von 3 wird Objekt I_4 entfernt. Die verbleibenden Objekte werden mit Support und Transaktionen in Ergebnistabelle L_1 übernommen.

Im nächsten Schritt werden alle häufigen 2-Objektmengen bestimmt. Tabelle 4.3 enthält alle Kandidaten für häufige 2-Objektmengen. Nach dem Algorithmus sollen jeweils zwei 1-Objektmengen zu einer 2-Objektmenge zusammengefasst werden, die sich in 0 Objekten überschneiden. Somit werden alle möglichen Kombinationen der häufigen 1-Objektmengen der Größe zwei gebildet. Es ist gewährleistet, dass nur Mengen enthalten sind, deren Teilmengen häufig sind. Im Beispiel werden drei Mengen ermittelt, die Kandidaten für häufige 2-Objektmengen sind: $\{I_1, I_2\}, \{I_1, I_3\}, \{I_2, I_3\}.$

| 2-Objektmenge | Support | Transaktionen |
|----------------|---------|-----------------|
| $\{I_1,I_2\}$ | 1 | T_4 |
| $\{I_1, I_3\}$ | 3 | T_1, T_2, T_3 |
| $\{I_2, I_3\}$ | 2 | T_5, T_6 |

Tabelle 4.3: C_2 , Tabelle der Kandidaten für häufige 2-Objektmengen

Bei der Menge $\{I_1, I_2\}$ ist der Support aller Objekte gleich groß, anders als bei den anderen beiden Mengen, bei denen der Support von I_1 bzw. I_2 kleiner ist. Im Fall von $\{I_1, I_3\}$ wird also die Menge der Transaktionen betrachtet, in denen I_1 vorkommt, und geprüft, ob auch I_3 in diesen Transaktionen enthalten ist. Ist dies, wie bei den Transaktionen T_1, T_2, T_3 der Fall, werden diese in Tabelle C_2 eingetragen und daraus der

Support ermittelt. Wegen des minimalen Supports von 3 bleibt am Ende genau eine häufige 2-Objektmenge, nämlich $\{I_1, I_3\}$.

Die Suche nach häufigen Mengen kann hier beendet werden, da es keine Kandidaten für 3-Objektmengen gibt. Es gibt keine zwei häufigen 2-Objektmengen, aus denen eine 3-Objektmenge gebildet werden kann. In jeder 3-Objektmenge, die aus den vorhandenen Objekten gebildet werden kann, gibt es mindestens eine Teilmenge, die nicht häufig ist. Daher gibt es keine häufigen 3-Objektmengen.

Aus allen häufigen Mengen, die mehr als ein Objekt enthalten, werden Assoziationsregeln gebildet. Im Beispiel gibt es eine häufige Menge mit mehr als einem Objekt. Aus der 2-Objektmenge $\{I_1, I_3\}$ lassen sich zwei Assoziationsregeln bilden. Deren Konfidenz wird berechnet durch

$$confidence(I_1 \to I_3) = \frac{support(I_1 \cup I_3)}{support(I_1)} = \frac{3}{4}$$
(4.3)

bzw.

$$confidence(I_3 \to I_1) = \frac{support(I_3 \cup I_1)}{support(I_3)} = \frac{3}{5}$$
 (4.4)

Durch eine minimale Konfidenz, als untere Schranke für die Konfidenz, lassen sich nützlich Assoziationsregeln finden. Je höher die Konfidenz, desto häufiger ist die Regeln anwendbar.

4.2 Teilsortimentsplanung

Aus einem umfangreichen Sortiment soll ein Teilsortiment ermittelt werden, welches die Basis für einen regelbasierten Empfehlungsdienst bildet. Hierzu wird aus dem gesamten zur Verfügung stehenden Sortiment ein Teilsortiment ausgewählt und an einen Harting Automaten angepasst. Dies ist ein Auswurfautomat mit einer gewissen Anzahl an Fächern einer bestimmten Größe.

In die Planung des Teilsortiments für den Harting Automaten gehen mehrere Faktoren ein. Zum Einen haben sowohl die Fächer des Automaten, als auch die Produkte eine bestimmte Größe. Ein Produkt kann nur einem Fach zugeordnet werden, wenn es in das Fach passt. Außerdem gibt es eine begrenzte Anzahl an Fächern. Der Harting Automat, der im Rahmen der vorliegenden Arbeit in Betrieb genommen werden soll, hat 20 Fächer. Eine möglichst gute Belegung der Fächer zu finden ist NP-hart, da es exponentiell viele Lösungen gibt aus einem großen Sortiment ein kleines auszuwählen. Um das Problem in angemessener Zeit zu lösen, wird ein Planungsalgorithmus verwendet. Dem Planungsalgorithmus wird eine Berechnungszeit mitgegeben, innerhalb derer er eine Lösung finden und optimieren soll.

Des Weiteren werden Produkte, die in der Vergangenheit häufig gekauft wurden, bei der Belegung bevorzugt. Produkte, die durch den Apriori Algorithmus als häufig identifiziert werden, haben Potenzial auch in Zukunft häufiger gekauft zu werden. Damit diese häufigen Produkte durch den regelbasierten Empfehlungsdienst empfohlen werden können, werden außerdem Assoziationen zwischen Produkten betrachtet. Diese Assoziationen werden durch den Apriori Algorithmus, wie in Abschnitt 4.1 erläutert, ermittelt.

Die gleichen Assoziationen werden auch für den regelbasierten Empfehlungsdienst genutzt. Bei der Planung des Teilsortiments werden nur Produkte aufgenommen, die in den Konklusionen der Regeln zu finden sind. Außerdem muss eine Konklusion immer vollständig im Teilsortiment enthalten sein. Diese Kriterien lassen sich durch die Arbeitsweise des in Abschnitt 4.3 erklärten, regelbasierten Empfehlungsdienst begründen.

Durch die Nutzung der gleichen Assoziationen ergänzen sich das Teilsortiment und der regelbasierte Empfehlungsdienst und es wird eine hohe Quote an Empfehlungen erwartet.

4.3 Regelbasierter Empfehlungsdienst

Ein regelbasierter Empfehlungsdienst ist kaum zusätzlicher Aufwand für den Händler und bietet den Kunden personalisierte Empfehlungen, ohne Angabe von persönlichen Daten. Im Allgemeinen bilden Implikationen die Grundlage für regelbasierte Empfehlungsdienste [3]. Im Fall des vorliegenden Systems dienen die Assoziationsregeln, deren Berechnung durch den Apriori Algorithmus in Abschnitt 4.1 erklärt wird, als Basis für den regelbasierten Empfehlungsdienst. Die Regeln bestehen aus einer Prämisse und einer Konklusion, welche jeweils nichtleere Mengen von Produkten sind.

Der Vorteil eines regelbasierten Empfehlungsdienstes, im Vergleich zu den in Abschnitt 2.1 beschriebenen Diensten ist, dass außer den Regeln und dem aktuellen Einkaufswageninhalt keine weiteren Informationen benötigt werden, um einem Kunden Empfehlungen geben zu können. Der Einkaufswageninhalt ist nach dem Scannen an der Kasse bekannt. Da der Prozess der Empfehlungsgenerierung nur ein Abgleich von Regeln ist, kann die kurze Zeit zwischen dem Einscannen der Ware und dem Bezahlen genutzt werden. Im IRL sind die Produkte mit RFID-Funketiketten versehen. Diese erleichtern das Einscannen der Produkte. Der Kunde geht mit seinem Einkaufswagen durch eine RFID-Schleuse, wodurch alle Produkte registriert werden, ohne sie aus dem Wagen zu nehmen. Nach dem Einscannen der Produkte gibt der regelbasierte Empfehlungsdienst passende Empfehlungen. Die Funktionsweise eine regelbasierten Empfehlungsdienstes wird im Folgenden beschrieben.

Der gescannte Einkaufswageninhalt ist die Eingabe für den Empfehlungsdienst. Aus allen Regeln werden diejenigen ermittelt, deren Prämisse eine Teilmenge des Einkaufswageninhaltes ist. Das bedeutet, dass die Regel anwendbar ist, da ihre Prämisse erfüllt ist. Anschließend werden die Konklusionen der anwendbaren Regeln betrachtet. Diese sind nach Konfidenz absteigend sortiert. Tritt eine Konklusion zweimal auf, so wird ihr die höhere Konfidenz zugeordnet. Die Konklusionen werden der Reihe nach abgearbeitet. Alle Produkte der aktuellen Konklusion, die nicht bereits im Einkaufswagen enthalten sind, werden der Menge der Empfehlungen hinzugefügt. Dies geschieht so lange, bis die maximale Anzahl an Empfehlungen erreicht ist, oder keine weiteren Konklusionen mehr zur Verfügung stehen. In der vorliegenden Implementierung ist die maximale Anzahl der Empfehlungen auf sechs begrenzt. So fällt es dem Kunden leichter nicht den Überblick über zu viele Empfehlungen zu verlieren. Außerdem hat der Bildschirm, über den die Empfehlungen angezeigt werden sollen, eine begrenzte Größe.

Da dem regelbasierten Empfehlungsdienst ein zuvor geplantes Teilsortiment zu Grunde liegt, wird die Menge aller ermittelten Assoziationsregeln an das Teilsortiment angepasst. Dies geschieht, indem alle Regeln übernommen werden, deren Konklusion eine

Teilmenge des Teilsortiments ist. Da bei der Planung des Teilsortiments darauf geachtet wird, nur Produkte aus den Konklusionen der Regeln in das Teilsortiment aufzunehmen, befinden sich keine Produkte im Automaten, die nicht empfohlen werden können. Außerdem gibt es zu jedem Produkt des Teilsortiments mindestens eine Assoziationsregel, die das Produkt empfehlen kann. Das ist sichergestellt dadurch, dass Produkte nur im Teilsortiment aufgenommen werden, wenn die gesamte Konklusion enthalten ist oder das Produkt alleine die Konklusion einer Regel bildet.

Durch das Aussortieren der Regeln bleiben nur diejenigen übrig, die aus dem Teilsortiment Empfehlungen geben können. Dadurch ist der Abgleich der Regeln schneller, da nur die Prämisse und nicht auch die Konklusion überprüft werden muss.

Beispiel

Um die Funktionsweise des regelbasierten Empfehlungsdienstes zu veranschaulichen folgt ein Beispiel.

In Tabelle 4.4 ist ein exemplarischer Einkaufswageninhalt dargestellt.

| Einkaufswageninhalt | | |
|---------------------|--|--|
| Brot | | |
| Butter | | |
| Käse | | |

Tabelle 4.4: Exemplarischer Einkaufswageninhalt, als Eingabe für einen regelbasierten Empfehlungsdienst

Der Empfehlungsdienst in diesem Beispiel basiert auf dem Teilsortiment {Milch, Käse, Speck, Wurst} und beinhaltet die folgenden nach Konfidenz absteigend sortierten Regeln.

- 1. {Brot, Butter} $-0.3 \rightarrow \{\text{Wurst, K\"ase}\}\$
- 2. {Käse} $-0, 2 \rightarrow \{Milch\}$
- 3. {Eier} $-0, 1 \rightarrow \{\text{Speck}\}\$

Diese Regeln werden der Reihe nach betrachtet und ausgewertet:

- 1. anwendbar, da {Brot, Butter} eine Teilmenge des Einkaufwageninhaltes ist. Es wird {Wurst} empfohlen, da {Käse} bereits im Einkaufswagen enthalten ist.
- 2. anwendbar, da {Käse} eine Teilmenge des Einkaufswageninhaltes ist. Es wird {Milch} empfohlen.
- 3. nicht anwendbar, da {Eier} keine Teilmenge des Einkaufswageninhaltes ist. Es wird nichts empfohlen.

Es entsteht eine Menge mit zwei Empfehlungen {Wurst, Milch}. Falls maximal ein Artikel empfohlen werden soll wird der genommen, der aus der Regeln mit der höheren Konfidenz stammt, in diesem Fall {Wurst}.

4.4. Filtern Kapitel 4

4.4 Filtern

Alle Produkte, die in den Transaktionen zu finden sind, werden vom Apriori Algorithmus betrachtet und haben die Möglichkeit in den Harting Automaten eingefügt zu werden. Dies kann jedoch bei manchen Produkten ein Problem darstellen. Es gibt Produkte, die gekühlt werden müssen, zerbrechlich sind oder aus anderen Gründen nicht für den Automaten geeignet sind. Um diese nicht nachträglich aus dem Sortiment entfernen zu müssen, können die Produkte direkt zu Beginn gefiltert werden. Alle Produkte werden durch ihre eindeutige European Article Number (EAN, vgl. Abschnitt 9.2) identifiziert. Dadurch können gültige Produkte (oder ungültige, je nachdem welche Menge kleiner ist) vom Nutzer durch eine Liste von EANs definiert werden. Mit Hilfe dieser Liste werden beim Einlesen der Transaktionen alle ungültigen Produkte entfernt. Auf diese Weise fließen ungültige Produkte nicht in die Berechnung der Assoziationsregeln mit ein. Die Anzahl der Produkte und die Größe der Transaktionen und somit die Berechnungsdauer wird so reduziert. Außerdem enthält das geplante Teilsortiment nur Produkte, die vom Nutzer als gültig definiert wurden. Die Anzahl der Transaktionen bleibt jedoch gleich, um Support und Konfidenz nicht zu verfälschen. Es können leere Transaktionen entstehen, welche in die Berechnung des Supports eingehen.

Über den Filter könnten auch Produkte entfernt werden, die aus anderen Gründen nicht im Teilsortiment enthalten sein sollen. Eine Möglichkeit ist Produkte mit Altersbeschränkung, wie Zigaretten und Alkohol, als ungültig zu definieren. Sind diese Produkte im Automaten enthalten und werden minderjährigen Kunden empfohlen, hätte dies sicher einen schlechten Einfluss auf den Ruf des Empfehlungsdienstes und somit auf den Händler. Es ist außerdem sinnvoll die EAN für Flaschen- und Dosenpfand aus den Transaktionen zu entfernen. Diese haben einen hohen Support, da sie zu den meisten Getränken dazu gebucht werden (vgl. Abschnitt 6.1).

4.5 Schnittstelle RAAsP

In diesem Abschnitt wird erläutert, wie die Anforderungen aus den vorherigen Abschnitten in der Schnittstelle RAAsP integriert werden. Hierzu bietet Abbildung 4.1 einen Überblick über die bereitgestellten Methoden in Form eines UML Klassendiagramms. Die Aufgaben der einzelnen Methoden werden in drei Bereiche unterteilt, welche im Folgenden erklärt werden.

Konfiguration

Zu Beginn müssen die Transaktionen, die der Apriori Algorithmus analysieren soll, dem System hinzugefügt werden. Um dem Nutzer die Möglichkeit zu geben Daten jedes Formats zu laden, wird eine Schnittstelle TransactionReader bereitgestellt. Durch Implementieren dieser Schnittstelle mit der abstrakten Methode

readTransaction(Database)

entscheidet der Nutzer wie und von wo die Transaktionen gelesen werden. Wichtig ist nur, dass sie der übergebenen Datenbasis hinzugefügt werden.

«Interface» RAAsP

- +setItemLoader(ItemLoader)
- +setTransactionReader(TransactionReader)
- +addAssortmentSlot (ID, min/max width, min/max height, min/max depth)
- +computeSelectedAssortment (minConf, minSup, maxTime)
- +getSelcetedAssortment ()
- +getRelevantRules ()
- +setUpRecommenderSystem (selectedAssortment, relevantRules)
- +recommender (Set of Items)

Abbildung 4.1: UML Klassendiagramm für die Schnittstelle RAAsP

Sollten die Transaktionen nicht alle notwendigen Informationen über die Objekte enthalten, können beim Laden der Objekte weiter Informationen hinzugefügt werden. Dies geschieht durch Implementieren der Schnittstelle ItemLoader und die abstrakte Methode onItemLoad(Item). Diese Methode kann einem Objekt beim einmaligen Laden in die Datenbank weitere Informationen hinzufügen. ItemLoader und TransactionReader werden mit den folgenden Methoden zum System hinzugefügt.

```
setItemLoader(ItemLoader)
setTransactionReader(TransactionReader)
```

Ebenfalls zur Konfiguration gehört die Methode

addAssortmentSlot (ID, min/max width, min/max height, min/max depth).

Diese fügt dem System die Fächer hinzu, in denen das Teilsortiment aufbewahrt werden soll. Hierzu ist eine eindeutige ID für jedes Fach, sowie die minimale und maximale Größe (bestehend aus Breite, Höhe und Tiefe) des Fachs notwendig. Dem System können beliebig viele Fächer hinzugefügt werden.

Soll die Größe der Fächer und Produkte nicht bei der Planung des Teilsortiments berücksichtigt werden, sollte jedem Fach eine Größe zugewiesen werden, in die jedes Produkt passt. Alternativ kann jedem Produkt eine Größe zugeordnet werden, die in jedes Fach passt. Der Effekt ist der gleiche, nämlich dass ein Sortiment geplant wird, welches maximal so viele Produkte enthält wie Fächer dem System hinzugefügt werden und die Größe kein Ausschlusskriterium ist.

Teilsortimentsplanung

Die Berechnung der Assoziationsregeln durch den Apriori Algorithmus aus Abschnitt 4.1, sowie die Planung des Teilsortiments wie in Abschnitt 4.2 beschrieben, übernimmt die Methode

computeSelectedAssortment (minConf, minSup, maxTime).

Der Methode wird eine minimale Konfidenz (minConf), ein minimaler Support (minSup) und eine maximale Berechnungszeit (maxTime) übergeben. Zunächst wird der Apriori Algorithmus mit den Transaktionen, die durch den TransactionReader der Datenbasis hinzugefügt werden, ausgeführt. Die Parameter minimale Konfidenz und minimaler Support werden dem Apriori Algorithmus ebenfalls übergeben. Das Ergebnis des Apriori Algorithmus ist eine Menge von Assoziationsregeln. Anschließend wird ein Planungsalgorithmus gestartet, der innerhalb der übergebenen maximalen Berechnungszeit ein Teilsortiment planen soll. In die Planung gehen Größe und Anzahl der Fächer, sowie Größe und Häufigkeit der Produkte und die Assoziationsregeln ein. Die Funktionsweise des Planungsalgorithmus ist in Abschnitt 5.2 genauer erläutert.

Der Nutzer hat abschließend die Möglichkeit, die relevanten Regeln für den Empfehlungsdienst und das Teilsortiment mit Hilfe der Methoden

```
getRelevantRules()
getSelectedAssortment()
```

auszulesen. Relevante Regeln sind die Regeln, deren Konklusion nur Produkte aus dem geplanten Teilsortiment enthält. Das Teilsortiment wird in Form eines Plans zurückgegeben, der möglichst jedem Fach ein Produkt zuordnet (Fächer können leer sein). Es ist empfehlenswert diese zwei Ergebnisse zwischen zu speichern, da ansonsten bei jedem Start des Systems Regeln und Sortiment neu berechnet werden müssen. Eine Neuberechnung ist jedoch nur notwendig, wenn sich die Eingaben ändern. Daher ist es zeitsparend, wenn die Daten zwischengespeichert werden, anstatt sie neu zu berechnen.

Empfehlungsdienst

Zuletzt wird der Empfehlungsdienst mit den zuvor gespeicherten Regeln und dem Teilsortiment mit Hilfe von

```
setUpRecommendationSystem(relevantRules, selectedAssortment)
```

gestartet. Ab diesem Moment können Empfehlungen generiert werden, bis das System beendet oder Daten geändert werden sollen. Im ersten Fall kann das System erneut mit den zuvor gespeicherten Daten gestartet werden. In letzterem Fall ist wieder mit der Konfiguration zu beginnen. Die Empfehlungen werden mit der Methode

```
recommender(Set of Items)
```

durch Eingabe einer Menge von Objekten, wie in Abschnitt 4.3 erläutert, berechnet.

5 Implementierung

Im Rahmen der vorliegenden Arbeit wird die Schnittstelle RAASP (Abschnitt 4.5) durch RAASPImpl implementiert. Diese Implementierung wird im IRL integriert, um den Harting Automaten, der dort im Kassenbereich steht, in Betrieb zu nehmen. Zusätzlich enthält die Implementierung eine grafische Benutzeroberfläche, die es dem Kunden ermöglicht Empfehlungen anzunehmen oder abzulehnen. Außerdem sind sowohl der implementierte Apriori Algorithmus, als auch der Planungsalgorithmus unabhängig von der Schnittstelle nutzbar.

5.1 Apriori Algorithmus

Im Rahmen der vorliegenden Arbeit wird der Apriori Algorithmus implementiert. Frühere Implementierungen des Algorithmus in Java sind nicht allgemein genug, um sie für das vorliegende Problem zu adaptieren. Daher wird eine teilweise parallele Implementierung angefertigt, die auf das vorliegende Problem passt, aber trotzdem allgemein ist.

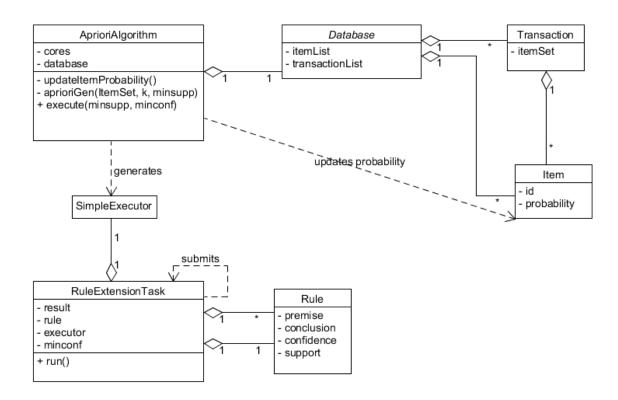


Abbildung 5.1: Vereinfachtes Klassendiagramm des implementierten Aprioi Algorithmus

Abbildung 5.1 zeigt zur Übersicht ein vereinfachtes Klassendiagramm des Apriori Algorithmus. Der implementierte Apriori Algorithmus arbeitet auf einer Datenbasis, der Klasse Database. Diese enthält jeweils eine Liste von eingelesen Transaktionen und Objekten. Dies sind Instanzen der Klassen Transaction und Item, wobei eine Transaction aus mehreren Items besteht. Die Klasse Apriori Algorithmus ermittelt zunächst für jedes Objekt der Klasse Item die Häufigkeit bzw. den Support für alle 1-Objektmengen. Diese Berechnung wird parallel in mehreren Threads ausgeführt. Die Anzahl der Threads kann entweder vom Nutzer bestimmt werden oder entspricht per default der Anzahl der verfügbaren Prozessoren. Um die Aufgaben gleichmäßig zu verteilen und alle Prozessoren auszulasten, wird die Menge der Objekte partitioniert. Die Anzahl der Partitionen entspricht der Anzahl der Threads. Die Threads arbeiten die ihnen zugewiesene Partition ab, wobei jeder Thread für jedes Item über alle vorhandenen Transaktionen iteriert und ermittelt in wie viel Prozent der Transaktionen das Objekt vorkommt. Anschließend werden die häufigen 1-Objektmengen durch Vergleich mit dem minimalen Support herausgefiltert.

Aus den häufigen 1-Objektmengen bzw. im Allgemeinen k-Objektmengen werden k+1-Objektmengen generiert. Diese Berechnung wird parallel ausgeführt. Dafür wird ein Threadpool bereitgestellt mit der zuvor definierten Anzahl an Threads. Jeder Thread bekommt so lange Aufgaben zugewiesen, bis keine weiteren Aufgaben mehr vorhanden sind. Die Aufgabe jedes Threads ist die Generierung aller möglichen k+1-Objektmengen aus der ihm zugewiesenen k-Objektmenge. Dazu wird über die Menge der übrigen k-Objektmengen iteriert und eine Objektmenge gesucht, die sich mit der ursprünglichen k-Objektmenge genau in k-1 Objekten überschneidet. Für jede so gefundene k+1-Objektmenge wird durch den minimalen Support ermittelt, ob sie häufig ist. Dabei wird die Idee von Hassan M. Najadat und seinem Team [7] zur Verbesserung der Laufzeit genutzt, die in Abschnitt 4.1 erläutert wird. Anstatt alle Transaktionen nach der k+1-Objektmenge zu durchsuchen, werden nur die Transaktionen betrachtet, in denen eine der k-Objektmengen vorkommt. Die Anzahl der zu betrachtenden Transaktionen kann so extrem reduziert werden, wodurch die Laufzeit stark verkürzt wird. Je kleiner der minimale Support, desto stärker wird die Laufzeit reduziert [7]. Bei einzelnen Tests mit einem minimalen Support von 0,001 und ca. 300.000 Transaktionen ist eine Laufzeitverkürzung von über 90% festzustellen. In dem Artikel "An Improved Apriori Algorithm for Association Rules" [7] wird bei einem Support aus dem Intervall [0,02, 0,1] von einer durchschnittlichen Laufzeitverkürzung von 67% gesprochen.

Wird auf diese Weise eine häufige Objektmenge gefunden, so wird sie synchronisiert dem Ergebnis hinzugefügt. Sollte das Ergebnis leer sein, so wurden von keinem der Threads neue Objektmengen gefunden und die Berechnung wird beendet. Ist das Ergebnis nicht leer, so werden erneut eins größere Objektmengen gebildet.

Im Anschluss werden aus allen gefunden Objektmengen, die mehr als ein Objekt enthalten, Assoziationsregeln gebildet. Es wird jeweils aus einer Menge eine Regel mit einelementiger Konklusion gebildet. Die restlichen Objekte der Menge sind in der Prämisse enthalten. Von der Klasse RulesExtensionTask, welche die Schnittstelle Runnable implementiert, wird für jede so gefundene Regel eine Instanz erzeugt und an den Threadpool übergeben. Ein RuleExtensionTask prüft zunächst ob die Konfidenz der Regel über der minimalen Konfidenz liegt. Ist dies der Fall, wird die Regel dem Ergebnis hinzugefügt. Falls die Prämisse der Regel mehr als ein Objekt enthält, werden so viele neue Regeln generiert, wie sich Objekte in der Prämisse befinden. Die Regeln haben jeweils

die gleiche Konklusion wie die initiale Regel und zusätzlich ein Objekt aus der Prämisse. Dieses Objekt wird aus der Prämisse entfernt. Für jede so generierte Regel wird ein RuleExtensionTask an den Threadpool übergeben. Dadurch ist gewährleistet, dass jede mögliche Regel mindestens einmal betrachtet wird. Die RulesExtensionTasks verzweigen sich so weit, bis keine weiteren Regeln mehr aus dieser häufigen Menge generiert werden können. Abbildung 5.2 verdeutlicht die Vorgehensweise der RuleExtensionTasks an einem Beispiel. In der vorliegenden Anwendung des Apriori Algorithmus werden

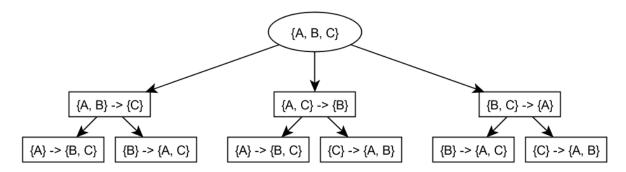


Abbildung 5.2: Arbeitsweise des RuleExtensionTask

höchstens 4-Objektmengen gebildet. Daher kann die mehrfache Berechnung der Regeln vernachlässigt werden.

5.2 Planungsalgorithmus

Für den Planungsalgorithmus wird der OptaPlanner¹ genutzt, eine Open Source Optimierungssoftware. Dieser ist geeignet für NP-harte Probleme. Der Vorteil dieses Planers im Vergleich zu Alternativen ist die Domainsprache. Während andere Planer eine eigene Domainsprache einführen, wie zum Beispiel PDDL (= Planning Domain Definition Language), werden die Planungsprobleme im OptaPlanner in Java spezifiziert. Dies ermöglicht eine einfache Integration in die vorliegende Implementierung. Die Implementierung des Planers ist an das Problem der Teilsortimentsplanung für den Automaten angepasst, kann jedoch auch unabhängig davon genutzt werden.

In Abbildung 5.3 ist ein vereinfachtes Klassendiagramm zur Erläuterung des Planungsalgorithmus zu sehen, welches im Folgenden beschrieben wird. Die Klasse Slot ist eine Planungsentität und entspricht in der vorliegenden Implementierung einem Fach im Harting Automaten. Jeder Slot hat eine eindeutige ID und eine minimale und maximale Größe. Des Weiteren kann jeder Planungsentität eine Planungsvariable zugeordnet werden. Planungsvariablen sind in diesem Fall Instanzen der Klasse PlanningItem, welche den Produkten entsprechen. Auch diese haben eine eindeutige ID und eine Größe und darüber hinaus die Häufigkeit, welche durch den Apriori Algorithmus errechnet wird. Die Klasse Solution repräsentiert eine Lösung des Problems. Das bedeutet, dass die Klasse Solution eine Liste von Slots beinhaltet, denen jeweils ein PlanningItem zugeordnet ist. Jede Lösung enthält somit eine Belegung für den Harting Automaten. Das Feld score der Klasse Solution beinhaltet eine Bewertung, ob die Lösung gültig ist und wie gut sie ist. Diese Bewertung wird berechnet von den Klasse ScoreCalulator

¹http://www.optaplanner.org/

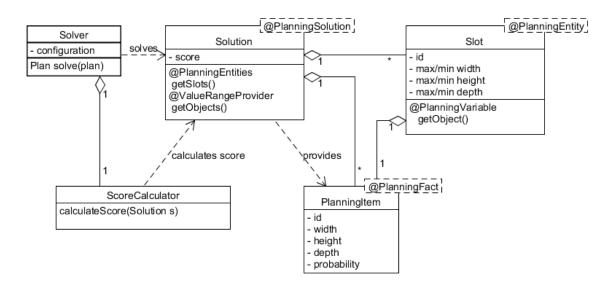


Abbildung 5.3: Vereinfachtes Klassendiagramm des Planungsalgorithmus

und wird im folgenden Abschnitt genauer beschrieben. Abhängig von dieser Bewertung kann die Klasse Solver das Problem lösen. Dafür wird initial eine beliebige Instanz der Klasse Solution übergeben, welche durch den Solver mit Hilfe der Bewertung verbessert wird, bis eine gültige und möglichst gute Lösung gefunden wird. Der Solver betrachtet beliebige Lösungen, also auch ungültige. Allein durch die Bewertung wird sichergestellt, dass am Ende eine gültig Lösung zurückgegeben wird.

Bewertungsfunktion

In der vorliegenden Implementierung wird eine HardSoftScore verwendet. Das hießt, die Bewertung der Klasse Solution besteht aus einer Hardscore und einer Softscore. Sobald die Hardscore negativ ist, ist die Lösung ungültig und wird verworfen. Je höher die Softscore, desto besser die Lösung. In der vorliegenden Implementierung gehen die in Tabelle 5.1 dargestellten Faktoren in die Bewertung ein.

| Hardscore | Softscore |
|------------------------------------|--------------------------|
| - Objekt passt nicht in Fach | - Fach ist leer |
| - Objekt ist mehrfach in Lösung | + Häufigkeit des Objekts |
| - Objekt tritt ohne Konklusion auf | |

Tabelle 5.1: Faktoren, die in die Bewertung der Lösung des Planungalgorithmus eingehen

Die Hardscore wird negativ und damit die Lösung ungültig, sobald einer der aufgelisteten Fälle eintritt. Für jeden Fall wird die Hardscore, die initial 0 ist um 1 verringert. Der erste Fall ist, dass ein Objekt nicht in das Fach passt, dem es zugeordnet ist. Die Lösung wird dadurch ungültig, da die so gefundene Belegung des Automaten unmöglich umzusetzen ist. Der zweite Fall ist, dass ein Objekt mehrfach in der Lösung vorkommt. Mehrfachbelegungen sind in der Lösung nicht erwünscht, da so weniger verschiedene

Objekte im Teilsortiment enthalten sind. Der letzte Fall ist, dass ein Objekt ohne die zugehörigen Objekte einer Konklusion auftritt (vgl. Abschnitt 4.2). Dadurch ist es dem Empfehlungsdienst nicht möglich dieses Objekt vorzuschlagen und es belegt ein Fach im begrenzten Automaten, das sinnvoller genutzt werden kann. Um die so gefundenen gültigen Lösungen zu bewerten ändert sich die Softscore aus den angegeben Gründen. Die Softscore wird für jedes leere Fach, dass sich in der Lösung befindet, um 1.000 verringert. Dies ist notwendig, da Lösungen mit möglichst vielen belegten Fächern besser sind als Lösungen, bei denen Fächer leer bleiben. Je häufiger die Objekte des Teilsortiments in vergangenen Transaktionen vorkommen, desto höher soll die Softscore sein. Objekte die häufig in vergangen Transaktionen vorkommen haben eine höhere Wahrscheinlichkeit auch für folgende Transaktionen relevant zu sein und von dem Empfehlungsdienst vorgeschlagen zu werden. Diese Bewertung wird gewährleistet, indem die Softscore für jedes Objekt um dessen Häufigkeit erhöht wird. Diese Häufigkeit liegt im Intervall [0, 1] und entspricht dem Support der entsprechenden 1-Objektmenge. Das Feld Softscore ist vom Typ int und somit wird die Häufigkeit in allen Fällen auf 0 oder 1 gerundet. Um einen aussagekräftigen Wert zu erhalten, wird die Häufigkeit mit 100 multipliziert. Da die SoftScore für die Häufigkeit um maximal 100 erhöht werden kann und ein leeres Fach die SoftScore um 1.000 verringert, ist gewährleistet, dass Lösungen mit möglichst wenig leeren Fächern gefunden werden. Der Planungsalgorithmus versucht die Lösung schnellstmöglich zu verbessern, was am einfachsten durch Belegung leerer Fächer geht. Das Kriterium der leeren Fächer ist in der Softscore umgesetzt, da eventuell nicht genügend Produkte vorhanden sind, um jedes Fach zu belegen. Der Planungsalgorithmus findet dann trotzdem eine Lösung.

Anpassen an andere Planungsprobleme

Die Implementierung des Planungsalgorithmus ist einfach anzupassen für andere Planungsprobleme. Da die Gültigkeit sowie die Qualität der Lösungen einzig durch die Berechnung der Bewertung definiert wird, kann für andere Planungsprobleme die Berechnung angepasst werden. Hierzu muss das Interface EasyScoreCalculation neu implementiert werden. Diese Implementierung beinhaltet als einziges die abstrakte Methode calculateScore. Durch die Implementierung dieser Methode wird definiert, wann eine Lösung gültig ist und wie gut sie ist.

Noch einfacher ist die Anpassung, falls die Größe der Objekte und Fächer nicht in die Planung mit eingehen soll. Dafür wird jedem Objekt eine Größe zugeordnet, die in jedes der Fächer passt. Auf diese Weise ist die Größe der Produkte kein Ausschlusskriterium bei der Planung.

5.3 Implementierung der Schnittstelle RAAsP

Im Rahmen der vorliegenden Arbeit wird die Schnittstelle RAAsP durch RAAsPImpl implementiert. Die meisten der in Abschnitt 4.5 vorgestellten Methoden der Schnittstelle RAAsP sind intuitiv zu implementieren. Die Herausforderungen sind das Interface TransactionReader und die Methode computeSelectedAssortment, welche das Teilsortiment plant. In der vorliegenden Implementierung fließen in die Sortimentsplanung neben der Größe der Produkte und Fächer und der Anzahl der Fächer auch die Assozia-

tionsregeln und der Support der Produkte ein, den der Apriori Algorithmus berechnet. Dadurch ergänzt das Sortiment den regelbasierten Empfehlungsdienst, der die gleichen Assoziationsregeln verwendet.

BonReader - Implementierung der Schnittstelle TransactionReader

Die Transaktionen, die in der vorliegenden Arbeit analysiert werden, sind im .bon Format gespeichert. Um das Format zu erläutern ist in Abbildung 5.4 eine exemplarische .bon Datei dargestellt.

Abbildung 5.4: Exemplarische .bon Datei

Die erste Zeile enthält einen Zeitstempel bestehend aus Datum und Uhrzeit. Die folgenden Zeilen enthalten die gekauften Produkte. Für den Apriori Algorithmus sind nur zwei Werte relevant. Die zweite Zahl in jeder Zeile ist die meist 13-stellige EAN, die das Produkt eindeutig identifiziert. Mit Hilfe dieser Nummer werden Objekte unterschieden und weitere Informationen, wie Größe, Name und Preis, aus der Datenbank ausgelesen und den Objekten zugeordnet. Es gibt Posten, bei denen keine EAN vorhanden ist, diese werden aussortiert. Ebenfalls wichtig ist die dritte Zahl, sie gibt an, wie oft das Produkt gekauft wurde. Diese Anzahl kann negativ sein bei einem Umtausch oder einer Stornierung, auch dann wird der Posten nicht weiter betrachtet.

Die restlichen Werte geben die Nummerierung der Posten, die Maßeinheit (ST = Stück, KG = Kilogramm), den Preis und einen weiteren Zeitstempel an.

Die Klasse BonReader implementiert das Interface TransactionReader. Diese Implementierung ist für die vorliegende Arbeit auf das .bon Format abgestimmt. Die Klasse BonReader hat neben der zu implementierenden abstrakten Methode readTransaction zwei private Methoden. Die Methode findBon(File folder) arbeitet sich durch die übergebene Ordnerstruktur und ruft für jede gefundene .bon Datei die Methode loadBon(File bon) auf. Diese generiert aus der Datei sowohl Instanzen der Klasse Item, die der Datenbasis hinzugefügt werden, als auch Instanzen der Klasse Transaction. In die Methode sind einige Kriterien eingebaut, wann ein Item gültig ist. Zum einen gilt es zu beachten, dass das Item überhaupt eine EAN besitzt und die Anzahl der gekauften Produkte nicht negativ ist. Des Weiteren kann definiert werden, ob EANs einer gewissen Größe aussortiert werden. In der vorliegenden Implementierung werden EANs, welche aus weniger als 13 Ziffern bestehen aussortiert. Außerdem ist es dem Nutzer möglich einen expliziten Filter hinzuzufügen, der bestimmte EANs aussortiert. Weiter können auch Produkte mit internen EANs aussortiert werden. Interne EANs werden Produkten zugeordnet, die nur in diesem Laden verkauft werden und vor Ort abgewogen werden. Ihre EANs starten mit der Ziffer "2" und werden in der vorliegenden Implementierung aussortiert, da sie nicht für den Automaten geeignet sind.

computeSelectedAssortment - Teilsortimentsplanung

Die Methode

computeSelectedAssortment(minConf, minSupp, maxTime)

führt den Apriori Algorithmus mit den, über den TransaktionReader eingelesen Transaktionen aus. Die Parameter für den minimalen Support und die minimale Konfidenz werden übergeben. Der Apriori Algorithmus generiert wie in Abschnitt 5.1 Assoziationsregeln. Anschließend wird ein Planungsalgorithmus aufgerufen 5.2, der das NP-harte Problem der Teilsortimentsplanung übernimmt. Mit Hilfe der Assoziationsregeln versucht er in der angegeben Zeit maxTime einen Plan zu finden und zu optimieren.

5.4 Grafische Benutzeroberfläche

Für das System wird eine grafische Benutzeroberfläche (GUI, engl. graphical user interface) implementiert, welche über einen Touchscreen dargestellt werden soll und dem Kunden zum Annehmen oder Ablehnen der Empfehlungen dient. Dieser Touchscreen ist Teil des Harting Automaten, der im Kassenbereich stehen soll. Abbildung 5.5 zeigt die grafische Benutzeroberfläche, nachdem ein Kunde in den Kassenbereich getreten ist. Im Zentrum sind die an den Einkauf des Kunden angepassten Empfehlungen zu sehen. Damit der Kunde nicht die Übersicht über die Empfehlungen verliert, werden höchstens sechs Empfehlungen gegeben und angezeigt. Diese werden durch eine Grafik visualisiert und erhalten als Beschriftung Name und Preis. Der Kunde hat nun die Wahl, ob er ein oder mehrere Produkte auswählt oder auf der rechten Seite mit dem "Nein Danke"-Knopf die Empfehlungen verwirft. Der über dem Knopf befindliche Ladebalken hat nach Ablauf von 10 Sekunden den gleichen Effekt wie der "Nein Danke"- Knopf. Er dient dazu die Empfehlungen zu verwerfen, wenn der Kunde nicht interessiert ist, um dem nächsten Kunden Empfehlungen geben zu können. Nach Auswahl einiger Produkte sieht die grafische Oberfläche aus, wie in Abbildung 5.6 gezeigt. Rechts oben sind die gewählten Produkte aufgelistet und die Summe der Preise angegeben. Ist ein Produkt nicht mehr im Automaten verfügbar, wird es als "Ausverkauft" angezeigt. Der zuvor ausgegraute Knopf "Zurücksetzen" ist nun aktiv. Es setzt die Auswahl der Produkte zurück, so dass die Benutzeroberfläche wieder aussieht wie in Abbildung 5.5. Lediglich der Ladebalken wird dann nicht mehr angezeigt, da der Kunde bereits auf die Empfehlungen reagiert hat und so sein Interesse gezeigt hat. Durch Klicken des Knopfes mit der Aufschrift "OK" sollen die Produkte aus dem Harting Automaten ausgeworfen werden und die Summe auf den bisherigen Kaufpreis addiert werden.



Abbildung 5.5: Exemplarischer Zustand der grafischen Benutzeroberfläche nachdem ein Kunde in den Kassenbereich getreten ist



Abbildung 5.6: Exemplarische Benutzeroberfläche nach Auswahl einiger Produkte durch den Kunden

6 Evaluation

Die Evaluation der vorliegenden Implementierung von RAAsP unterteilt sich in zwei Teile. Zum Einen wird die Konfiguration überprüft, das bedeutet, dass verschiedene Werte für die zu übergebenden Parameter getestet werden. Es wird argumentiert, welche Werte für die Parameter am sinnvollsten sind. Zum Anderen wird der Empfehlungsdienst mit Originaldaten der GLOBUS SB-Warenhaus Holding getestet. Diese Daten beinhalten aufgezeichnete Einkäufe aus einer Globus-Filiale.

6.1 Parameter prüfen

Zu prüfende Parameter sind sowohl minimale Konfidenz und minimaler Support des Apriori Algorithmus, als auch die maximale Berechnungszeit, die der Planungsalgorithmus benötigt. Für alle Tests wurden Originaldaten aus der Globus-Filiale in Güdingen verwendet. Hierbei handelt es sich um Einkaufszettel über eine Spanne von vier Monaten. Pro Monat wurden ungefähr 300.000 Einkaufszettel aufgezeichnet.

Minimaler Support und minimale Konfidenz

Um einen Überblick zu erlangen, in welchen Bereichen der minimale Support und die minimale Konfidenz liegen sollten, wird der Apriori Algorithmus mehrfach mit unterschiedlichen Parametern gestartet. Dabei ergeben erste Messungen das in Abbildung 6.1 dargestellte Diagramm.

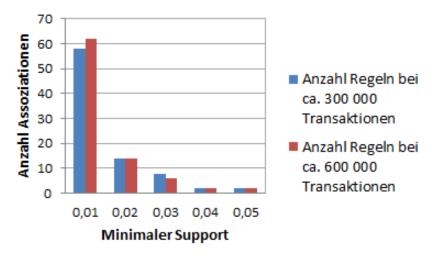


Abbildung 6.1: Anzahl Assoziationsregeln bei verschiedenen minimalen Supports im Intervall [0.01, 0.05], minimale Konfidenz = 0

Für den minimalen Support werden Werte im Intervall [0.01, 0.05] im Abstand von

0.01 gewählt. Die minimale Konfidenz hat den Wert 0. Der Apriori Algorithmus wird mit Einkaufszetteln von einem Monat (≈ 300.000 Transaktionen) und von zwei Monaten ($\approx 600~000$ Transaktionen) gestartet. An der vertikalen Achse ist die Anzahl der Assoziationsregeln abzulesen, an der horizontalen Achse der minimale Support. Es ist deutlich zu sehen, dass der minimale Support im Bereich zwischen 0 und 0,01 liegen sollte, um eine angemessene Anzahl an Assoziationsregeln finden zu können. Die Ausgabe der Konfidenzen der Assoziationsregeln zeigt, dass diese über das gesamte Intervall [0, 1] verteilt sind.

Für die weitere Evaluation wird der minimale Support im Intervall [0.001, 0.01] im Abstand 0.01 gewählt und die minimale Konfidenz im Intervall [0,1] im Abstand 0.05. Auf einen Support von 0 wird verzichtet, da dies bei einer Menge von rund 20.000 Produkten zu mehreren Millionen 2-Objektmengen führen würde. Außerdem würde kein Produkt aussortiert werden, was dazu führt, dass jede jemals gekaufte Kombination von Produkten als häufige Menge erkannt wird.

Mit den so entstehenden 200 Parameter Paaren wird der Apriori Algorithmus gestartet mit Transaktionen von einem, zwei, drei und vier Monaten. In Abbildung 6.2 ist ein Diagramm zu sehen, das die Ergebnisse des Apriori Algorithmus mit ca. 300.000 Transaktionen (1 Monat) darstellt.

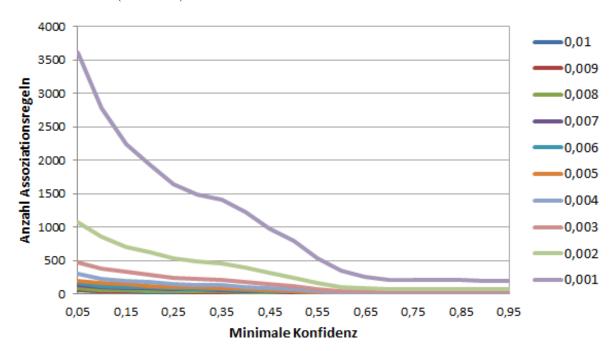


Abbildung 6.2: Anzahl Assoziationsregeln für Transaktionen: ca. 300.000, Produkte: ca. 62.000, minimaler Support: [0,001, 0,01], minimale Konfidenz [0, 1]

An der vertikalen Achse ist die Anzahl der gefunden Regeln aufgetragen, an der horizontalen Achse die minimale Konfidenz. Die einzelnen Linien entsprechen jeweils der Ausgabe für einen minimalen Support, wie er rechts aufgelistet ist. Dabei ist die Linie mit dem kleinsten Support als oberstes im Graph zu sehen. Mit steigendem Support sinkt die Anzahl der gebildeten Regeln, da weniger häufige Mengen gefunden werden. Es wird deutlich, dass die eingetragenen Linien bei einer minimalen Konfidenz im Intervall [0,65, 1] fast konstant bleiben. Das bedeutet, dass es kaum Regeln gibt mit Konfidenz

zwischen 0.65 und 1. Die Regeln mit Konfidenz 1 sind nicht repräsentativ und nicht für die weitere Planung nutzbar. Sie enthalten Flaschen- oder Dosenpfand, der beim Kauf von Getränken immer mit gebucht wird. So entstehen Regeln, die angeben, dass zwei Produkte bzw. in diesem Fall ein Produkt und der zugehörige Pfand, mit 100%iger Wahrscheinlichkeit zusammen gekauft werden. Bei genauerem Betrachten wird deutlich, dass bei einem Support von 0,001 Regeln mit Konfidenzen bis 0,77 zu finden sind. Regeln bis zu dieser Schranke beinhalten nicht nur Getränke und Pfand, sondern auch andere Produkte. Alle Regeln oberhalb dieser Schranke enthalten ausschließlich Getränke und Pfand. Aus diesem Wissen lässt sich schließen, dass die minimale Konfidenz kleiner als 0,65 gewählt werden sollte. Bei Betrachten des Graphen wird deutlich, dass die Anzahl der Regeln für Konfidenzwerte kleiner als 0,2 exponentiell ansteigt. Die Zeit, die der Apriori Algorithmus benötigt, um die Regeln zu ermitteln bleibt jedoch konstant für unterschiedliche minimale Konfidenzen. Auch die Zeit, die der Apriori Algorithmus bei kleinerem Support benötigt, ändert sich kaum. Abbildung 6.3 zeigt die Durchschnittszeit in Minuten, die der Apriori Algorithmus benötigt, um für die unterschiedlichen Werte für den minimalen Support alle Regeln zu berechnen. Die Grafik macht deutlich, wie stark die Laufzeit durch die Verbesserung von Mohammed Hassan M. Najadat [7] verkürzt wird. Während im Intervall [0,01, 0,004] die Zeit in beiden Fällen konstant ist, steigt sie beim unverbesserten Apriori Algorithmus für kleinere Supportwerte exponentiell an. Die Berechnung der Assoziationsregeln benötigt für einen minimalen Support unter 0,004 nur wenige Minuten, wohingegen die Berechnungszeit bei einem minimalen Support von 0,001 durch den unverbesserten Algorithmus bis zu 2 Stunden dauern kann. Der verbesserte Algorithmus benötigt für die gleiche Berechnung wenige Minuten. Interessant ist, dass der verbesserte Apriori Algorithmus für einen minimalen Support von 0.001 weniger Zeit benötigt als für einen minimalen Support von 0.002. Die Zeitangaben sollen eine groben Uberblick geben und sind stark hardwareabhängig. Die vorliegenden Berechnungen wurden auf einem Rechner mit der folgenden Spezifikation ausgeführt: Intel(R) Core(TM) i7-5930K, CPU @ 3.5Ghz, 32GB DDR3 Ram, Windows 8 64bit.

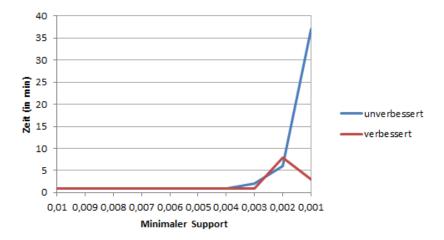


Abbildung 6.3: Vergleich der durchschnittlich benötigten Zeit in Minuten zum Berechnen der Assoziationsregeln zwischen einfachem und verbessertem Apriori Algorithmus

Zusammengefasst sollte die minimale Konfidenz zwischen 0,2 und 0,65 liegen und der minimale Support zwischen 0,001 und 0,01 sein. Es gilt zu beachten, dass ein größerer minimaler Support weniger häufige Mengen findet als ein kleiner. Dagegen erlaubt eine größere minimale Konfidenz weniger Regeln und eine kleine mehr Regeln aus den vorhandenen häufigen Mengen. Tabelle 6.1 gibt einen Überblick, wie sich größere bzw. kleinere Werte der Parameter auf die gefundenen Regeln auswirken.

| | kleiner minimaler Support | großer minimaler Support |
|--------------------------|---------------------------|--------------------------|
| | z.B. 0,002 | z.B. 0,01 |
| kleine minimale Konfi- | viele häufige Mengen | wenig häufige Mengen |
| denz z.B. 0,2 | viele Assoziationen | viele Assoziationen |
| große minimale Konfidenz | viele häufige Mengen | wenige häufige Mengen |
| z.B. 0,65 | wenige Assoziationen | wenige Assoziationen |

Tabelle 6.1: Auswirkung der Parameter minimaler Support und minimale Konfidenz auf die gefundenen Assoziationsregeln

Die Anzahl der Assoziationen ist weiterhin abhängig von der Anzahl der häufigen Mengen, jedoch werden mehr oder weniger Regeln aus den vorliegenden häufigen Mengen abgeleitet, je nachdem wie hoch die Konfidenz gewählt wird.

Für die vorliegenden Daten über 1, 2 und 3 Monate liefert eine minimale Konfidenz von 0,3 und ein minimaler Support von 0,003 innerhalb wenigen Minuten ca. 200 Assoziationsregeln.

Diese scheinen ausreichend für die Planung eines Teilsortiments. Jedoch liefert die Planung des Teilsortiments ein unerwartetes Ergebnis. Im Sortiment befinden sich genau drei Produkte, da nur diese drei Produkte in den Konklusionen der rund 200 Regeln vorhanden sind. Genaueres Betrachten der EANs ergibt, dass es sich um abgewogene Produkte handelt. Die EANs haben alle den Präfix "207" der bei Globus für Produkte aus der Meisterbäckerei verwendet wird.

Da vor Ort abgewogene Produkte nicht für den Automaten geeignet sind, werden alle EANs mit dem Präfix "2" aus den Transaktionen entfernt. Ebenso alle EANs, die weniger als 13 Stellen haben. Mit den so gefilterten Transaktionen werden die folgenden Analysen geführt. Für die gleichen 200 Parameterpaare mit Konfidenzen im Intervall [0,1] und Support im Intervall [0,001, 0,01] wird der Apriori Algorithmus gestartet und die Anzahl der gefundenen Assoziationsregeln ermittelt.

Abbildung 6.4 zeigt die Anzahl der durch den Apriori Algorithmus ermittelten Assoziationsregeln bei ca. 300.000 Transaktionen. Es ist deutlich zu sehen, dass viel weniger Regeln gefunden werden, da nun viel weniger Produkte in den Transaktionen enthalten sind. Für Support größer 0,004 werden keine Regeln gefunden. Es ist erneut eine Schranke bei einer Konfidenz von ca. 0,7 zu sehen, ab der keine Regeln mehr gefunden werden. Die Anzahl der Transaktionen ändert kaum etwas an der Anzahl der Regeln. Zumindest bei der Analyse der vorliegenden Daten, sind die Ergebnisse für 300.000 und 1.200.000 Transaktionen fast identisch.

Somit ergeben sich als bestmögliche Parameter für die vorliegenden Daten ein minimaler Support von unter 0,002 und eine minimale Konfidenz nahe 0. Diese Parameter liefern genug Regeln, um eine Teilsortiment zu planen und das System zu evaluieren. Ob das gefundenen Teilsortiment und damit die Empfehlungen von Kunden angenom-

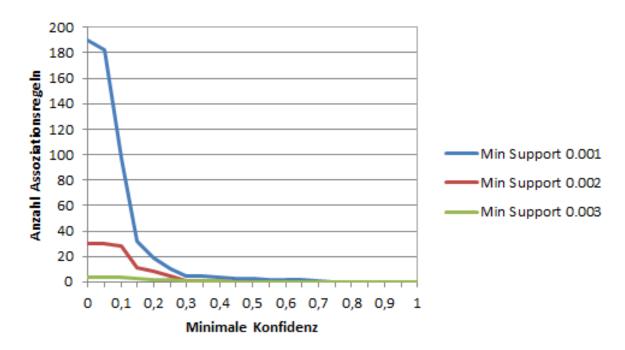


Abbildung 6.4: Anzahl Assoziationsregeln für Transaktionen: ca. 300.000, Produkte: ca. 55.000, minimaler Support: [0,001, 0,003], minimale Konfidenz [0, 1]

men werden, kann im Rahmen der vorliegenden Arbeit nicht getestet werden. Ebenfalls ist es schwierig die geeigneten Produkte in den Transaktionen zu belassen, da keine Zuordnung zwischen EAN und Produktname vorliegt. Durch Anfragen einzelner EANs wurden Schlüsse gezogen (z.B. alle EANs die mit Präfix "2" aussortieren), die eventuell in der praktischen Anwendung zu schlechten Ergebnissen führen. Trotz Allem arbeitet der Apriori Algorithmus in angemessener Zeit und liefert mit den ermittelten Parametern genug Regeln für weitere Evaluationen.

Maximale Berechnungszeit

Um eine qualifizierte Aussage über die maximale Berechnungszeit treffen zu können, werden die Größen der Produkte und Fächer benötigt. Diese liegen jedoch für die Produkte nicht vor. Das bedeutet, dass die Größen bei der Planung ignoriert werden. Im diesem Fall wird jedem Produkt eine Größe zugeordnet, die in jedes der Fächer passt. Mit den optimalen Parametern für den minimalen Support und die minimale Konfidenz werden mit dem Apriori Algorithmus Assoziationsregeln ermittelt. Eine Berechnungszeit von zwei Minuten reicht unter diesen Bedingungen aus, ein Teilsortiment zu finden und er zu verbessern.

6.2 Empfehlungsdienst testen

Um den regelbasierten Empfehlungsdienst in einem angemessenen Zeitrahmen und ohne Nutzerstudie testen zu können, werden gespeicherte Einkaufszettel aus der GLOBUS-Filiale Güdingen über eine Zeitspanne von vier Monaten genutzt. Hierzu werden Einkaufszettel aus einem, zwei und drei Monaten (1 Monat ≈ 300.000 Einkaufszettel) be-

nutzt, um das System zu initialisieren und jeweils Einkaufszettel von einem Tag (ca. 10.000-15.000), um das System zu evaluieren. Die Einkaufszettel zur Evaluation sollten nicht in die Berechnung des Teilsortiments eingehen, damit das Szenario der Realität entspricht. In einer realen Anwendung wird das System mit vorhandenen Einkäufen initialisiert, wobei für neue Einkäufe Empfehlungen generiert werden.

Zur Evaluation werden aufgezeichnete Einkaufszettel als Einkaufswageninhalt eingespeist und es wird notiert, in wie viel Prozent der Einkäufe der Empfehlungsdienst Ausgaben liefert.

Das Teilsortiment besteht aus 20 Produkten, die jedoch keine Größenangaben haben. Somit plant der Algorithmus das Sortiment ohne Berücksichtigung der Größe, nur mit Hilfe der Assoziationsregeln und Häufigkeiten der Produkte. Das System wird für die drei Berechnungen mit Einkaufszetteln von einem, zwei und drei Monaten, jeweils mit 20 Fächern initialisiert. Bei der Berechnung des Teilsortiments werden eine minimale Konfidenz von 0 und ein minimaler Support von 0,001 bzw. 0,002 übergeben. Die maximale Berechnungszeit beträgt 2 Minuten. Die Teilsortimente sind jeweils in Kapitel 9 aufgelistet.

Die Anzahl der Empfehlungen bei 13.281 Einkäufen als Eingabe für den Empfehlungsdienst sind im Kapitel 9 aufgelistet. Eine nennenswerte Empfehlungsquote liefert der Empfehlungsdienst, der mit Transaktionen aus zwei Monaten und einem Support von 0,001 initialisiert wurde. Hier liegt die Quote der Empfehlungen bei 20%. Das bedeutet, dass bei 13.281 Einkäufen in 20%, also 2.658 Einkäufen, mindestens ein Produkt aus dem Automaten empfohlen wird. Insgesamt werden 13.454 Produkte empfohlen. Im Durchschnitt wird somit zu jedem Einkauf etwas mehr als eine Empfehlung gegeben. Werden alle gegebenen Empfehlungen gekauft, so werden die Einnahmen des Tages um ca. 13.000€ erhöht. Genauere Angaben sind mit den vorliegenden Daten nicht möglich, da einige Produkte zu unterschiedlichen Preisen verkauft werden. Hochgerechnet auf 300.000 Einkäufe ergibt das etwa eine Umsatzsteigerung von 300.000€ im Monat.

7 Konklusion

Die vorliegende Arbeit liefert einen neuen Ansatz zur Umsetzung eines Empfehlungsdienstes im Offline-Handel. Dabei ist im Rahmen der vorliegenden Arbeit nicht nur ein theoretischer Ansatz entwickelt worden, sondern auch eine Implementierung angefertigt. Diese setzt den Empfehlungsdienst im IRL praktisch um. Darüber hinaus können sowohl der implementierte Planungsalgorithmus, als auch der teilweise parallele Apriori Algorithmus an andere Probleme angepasst und seperat genutzt werden.

Anstoß für die Entwicklung eines Empfehlungsdienstes im Offline-Handel sind die immer weiterverbreiteten Empfehlungsdienste im Online-Handel. Diese nutzen meist Filtermethoden, die durch Produktbewertungen oder Nutzerpräferenzen Empfehlungen geben. Die zwei bekanntesten Filtermethoden sind das kollaborative Filtern und das inhaltsbasierte Filtern. Beide sind nicht geeignet zur Lösung des vorliegenden Problems, da im Offline-Handel keinerlei Informationen über Produktbewertungen oder Nutzerpräferenzen zur Verfügung stehen. Ein weiteres Problem stellt die Frage, wann die Empfehlung gegeben werden soll und vor allem wie sie in den Einkaufswagen gelangt. Im Online-Handel reicht ein Klick um das empfohlene Produkt in den Warenkorb zu befördern. Die Lösungen dieser Probleme, sowie weitere erreichte Ziele werden im Folgenden aufgelistet und genauer erläutert. Einige Probleme oder Ideen, die sich bei näherem Betrachten des Themas aufgetan haben, konnten aus verschiedenen Gründen nicht umgesetzt werden. In Kapitel 8 wird ein Ausblick auf weiterführende Arbeiten gegeben.

Regelbasierter Empfehlungsdienst

Um ohne Informationen über Produktbewertungen oder Nutzerpräferenzen personalisierte Empfehlungen geben zu können, wird ein regelbasierter Empfehlungsdienst entwickelt. Dieser arbeitet auf einer Menge von Regeln, die ihm Information darüber geben welche Produkte zu einem Einkauf passen. Da die Ermittlung und Wartung dieser Regeln für große Datenmengen ein Problem darstellt, wird ein Big Data Algorithmus zur Datenanalyse genutzt. Es handelt sich um den Apriori Algorithmus, der aus einer Menge von Transaktionen Assoziationen zwischen den Objekten dieser Transaktionen herausfindet. In der vorliegenden Arbeit wird der Algorithmus genutzt, um aus vergangen Einkäufen Assoziationsregeln zu ermitteln, die Informationen darüber geben, welche Produkte wie häufig zusammen gekauft werden. Mit Hilfe dieser Implikationsregeln und der Information über den aktuellen Einkaufswageninhalt eines Kunden gibt der Empfehlungsdienst Empfehlungen. Bei geeigneter Wahl der Parameter für den Apriori Algorithmus, werden aus einer Menge von Transaktionen bis zu 200 Regeln gefunden.

Teilsortimentsplanung

Es ist am sinnvollsten Empfehlungen zu einem Einkauf erst an der Kasse zu geben, da dann der Einkauf beendet ist. So kann es nicht passieren, dass dem Kunden ein Produkt vorgeschlagen wird, dass er ohnehin kaufen will. Damit Empfehlungen direkt in den Einkaufswagen der Kunden gelangen, werden einige Produkte in einem Auswurfautomaten im Kassenbereich des Offline-Handels gelagert. Diese Produkte können von dem regelbasierten Empfehlungsdienst empfohlen werden und anschließend per Knopfdruck ausgeworfen werden. Die Auswahl dieser Produkte ist jedoch ein NP-hartes Problem, da aus mehreren 10.000 Produkten wenige Produkte ausgewählt werden müssen, die in den Automaten passen. Zur Lösung dieses Problems wird ein Planungsalgorithmus verwendet, der sowohl Größe der Fächer und Produkte, sowie die Anzahl der Fächer in Betracht zieht. Damit der Empfehlungsdienst möglichst viele Empfehlungen geben kann, werden außerdem die Assoziationen, die auch als Regeln für den Empfehlungsdienst dienen, mit in die Planung einbezogen. Es werden nur Produkte aus den Konklusionen der Regeln in das Teilsortiment aufgenommen, da nur diese empfohlen werden können. Außerdem treten Konklusionen immer als Ganzes auf. Da Produkte, die in der Vergangenheit oft gekauft wurden, auch in Zukunft oft gekauft werden, haben Produkte die häufig in den Transaktionen vorkommen eine höhere Chance ins Teilsortiment aufgenommen zu werden.

Die vorliegenden Daten zu Evaluation des Systems beinhalten keine Information über die Größe der Produkte. Der implementierte Planungsalgorithmus kann daher nur ohne Beachtung der Größe getestet werden. Für alle getesteten Eingaben liefert der Algorithmus innerhalb weniger Minuten ein Ergebnis, das alle spezifizierten Kriterien erfüllt.

Implementierung

Um das System praktisch umzusetzen wird eine Schnittstelle entwickelt und implementiert. Die Implementierung beinhaltet eine teilweise parallele Implementierung des Apriori Algorithmus, sowie die Implementierung des Planungsalgorithmus mit Hilfe des Opta Planners. Beide Module sind auch unabhängig von der Schnittstelle nutzbar. Dadurch kann der Planungsalgorithmus leicht angepasst werden und andere Kriterien können in die Teilsortimentsplanung einfließen. Ebenso kann der Planungsalgorithmus für andere Probleme genutzt werden. Auch der Apriori Algorithmus kann unabhängig genutzt werden. Das liefert einen großen Vorteil, da er für verschiedene Probleme genutzt werden kann. So wurde er zum Beispiel im Projekt HySociaTea des DFKI genutzt, zur Umsetzung einer ähnlichen Aufgabenstellung wie die der vorliegenden Arbeit (siehe letzter Abschnitt).

Evaluation

Da das System im Rahmen der vorliegenden Arbeit nur im Labor (IRL) praktisch umgesetzt werden kann, muss auf eine Nutzerstudie verzichtet werden. Um das System trotzdem evaluieren zu können stehen Originaldaten der GLOBUS SB-Warenhaus Holding zur Verfügung. Es handelt sich um Einkaufszettel in einer Spanne von vier Monaten. Zusammengefasst sind das ca. 1.200.000 aufgezeichnete Einkäufe. Ein Nachteil der vorliegenden Daten sind die fehlenden Größen der Produkte. Die Größen der

Produkte werden bei der Planung ignoriert, sind also kein Ausschlusskriterium für den Planungsalgorithmus. Daher liegen keine Daten vor, wie viel Zeit dem Planungsalgorithmus zugewiesen werden sollte, zur Berechnung des Teilsortiments. Ohne Betrachtung der Größen benötigt der Plaungsalgorithmus wenige Minuten zur Planung eines Teilsortiments.

Die Evaluation zeigt, dass die Wahl der Parameter des Apriori Algorithmus wichtiger ist als die Anzahl der Transaktionen. Bei gleiche Parametern liefert der Apriori Algorithmus für die dreifache Menge an Transaktionen (ca. 900.000) etwa die gleiche Anzahl an Regeln wie für die einfache Menge (ca. 300.000). Aus 200 Regeln lässt sich innerhalb von wenigen Minuten ein Sortiment mit 20 Produkten finden. Dieses Sortiment als Grundlage für den Regelbasierten Empfehlungsdienst liefert bei ca. 13.000 Einkäufen für ca. 20% aller Käufe mindestens eine Empfehlung. Da zu einem Einkauf mehrere Empfehlungen gegeben werden können, wird im Durchschnitt etwa zu jedem Einkauf eine Empfehlung gegeben. Beim Kauf aller Empfehlungen führt dies zu einer Umsatzsteigerung von ca. 13.000 € an einem Tag.

Integrieren in HySociaTea

HySociaTea ist ein aktuelles Projekt am Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) in Saarbrücken. Der Name HySociaTea leitet sich ab von "Hybrid Social Team for Long-Term Collaboration in Cyber-Physical Environments". In dem Projekt wird die Zusammenarbeit zwischen Menschen, Robotern und virtuellen Charakteren weiterentwickelt, um flüssige und selbstbestimmte Abläufe an Arbeitsplätzen zu sichern. Eine beispielhafte Umgebung besteht aus einem Arbeitsplatz, an dem Menschen arbeiten und einem Lager in dem sich Boxen mit Werkzeug und Bauteilen befinden. Hier werden Roboter eingesetzt, um auf Wunsch des Menschen Boxen zwischen Lager und Arbeitsplatz hin und her zu transportieren. Die virtuellen Charaktere Gloria und Curtis haben die Aufgabe Bestellungen von Menschen entgegenzunehmen und die Roboter zu beauftragen die gewünschte Box zu besorgen. In diesem Rahmen wird die vorliegende Arbeit nützlich integriert. Indem vergangene Bestellungen als Transaktionen angesehen werden kann das System Verbindungen zwischen den einzelnen Boxen knüpfen. Hierfür wird die, im Rahmen dieser Arbeit angefertigte, Implementierung des Apriori Algorithmus genutzt, um Assoziationsregeln zu finden. Dies geschieht auf die gleiche Weise, wie in Abschnitt 4.1 und 4.1 beschrieben. Diese Verbindungen bzw. Regeln dienen als Gehirn für Gloria und Curtis. In Abbildung 7.1 sind sowohl Gloria und Curtis als virtuelle Charaktere, als auch ihr Gehirn in Form eines Graphen dargestellt. Der Graph repräsentiert die Assoziationsregeln, die die Grundlage für den regelbasierten Empfehlungsdienst bilden. Die Knoten entsprechen dabei den unterschiedlichen Boxen, die im Lager stehen, und die Kanten stellen die Verbindungen zwischen den Boxen dar. Die Kanten sind gerichtet, da Assoziationsregeln nicht unbedingt symmetrisch sind. Eine Kante entspricht immer genau einer Regel, wobei der oder die Startknoten die Prämisse bilden und der oder die Zielknoten die Konklusion. Die Konfidenz ist jeweils an der entsprechenden Kante annotiert. Eine Bestellung des Menschen an Gloria wird als Warenkorb interpretiert. Da dem System als Grundlage das ganze Sortiment des Lagers zur Verfügung steht, wird kein Teilsortiment geplant und der regelbasierte Empfehlungsdienst agiert auf allen vorhandenen Boxen. Das bedeutet, dass Gloria und Curtis die bestmögliche Regel aus ihrem Gehirn auf die Bestellung anwenden und daraufhin dem

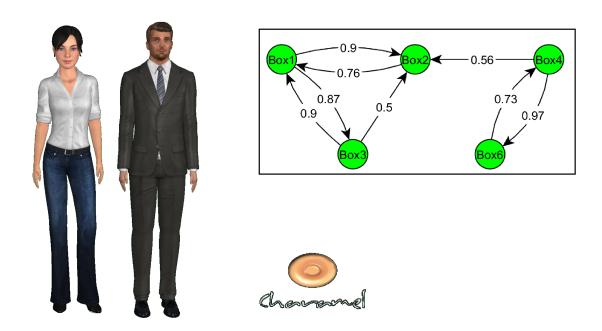


Abbildung 7.1: Virtuelle Charaktere Gloria und Curtis und die Assoziationsregeln für den regelbasierten Empfehlungsdienst in Form eines Graphen

Menschen ergänzende Bestellungen empfehlen. Diese werden direkt an den Roboter mit übergeben, wenn der Mensch dies wünscht.

Nachdem der Roboter die Boxen an den Arbeitsplatz gebracht hat und wieder stillsteht, wird das Gehirn von Gloria mit der letzten ausgeführten Bestellung aktualisiert. Dies geschieht, indem die letzte Bestellung dem System hinzugefügt wird und die Regeln aktualisiert werden. So ist das System immer auf den neusten Stand.

8 Weiterführende Arbeiten

In diesem Kapitel werden einige mögliche weiterführende Arbeiten vorgestellt. Sie konzentrieren sich darauf, das vorgestellte System weiter zu verbessern. In Abschnitt 8.1 basiert die Verbesserung auf einer Produkthierarchie, die in den vorliegenden Daten nicht vorhanden ist und im Rahmen der vorliegenden Arbeit zeitlich nicht umsetzbar ist. Die beiden folgenden Abschnitte 8.2 und 8.3 legen eine Nutzerstudie zu Grunde, welche zum einen zeitlich, aber auch praktisch im Rahmen der vorliegenden Arbeit nicht umsetzbar ist, da das vorgestellte System nur im IRL integriert ist. Zur Umsetzung der Idee aus Abschnitt 8.4 wird eine Datenbank mit Informationen über Produkte benötigt, die im Rahmen der vorliegenden Arbeit nicht angefertigt werden kann. Abschnitt 8.5 erläutert einige Ansätze der Evaluation, die mit umfangreicheren Daten angefertigt werden könnten. Letztendlich beschäftigt sich Abschnitt 8.6 mit der Frage, wie das neuartige System den Kunden näher gebracht werden kann.

8.1 Sortimentsebenen

In Abschnitt 2.2 wird das Konzept der Sortimentsebenen vorgestellt, welches auch für die vorliegende Arbeit nützlich sein kann. Die vorgestellte Implementierung RAAsPImpl unterscheidet und identifiziert Produkte allein durch die eindeutige EAN. Das bedeutet, dass die Unterscheidung auf der Sortimentsebene "Sorte" getroffen wird. Somit sind gleiche Artikel genauso gleich oder verschieden wie unterschiedliche Artikel. Diese Unterscheidung führt dazu, dass eine 0,331 Flasche Apfelsaft der Marke A und eine 0,51 Flasche Apfelsaft der Marke A im implementierten System keinerlei Verbindung zueinander haben. Dies kann zu dem Szenario führen, dass ein Kunde der eine 0,331 Flasche Apfelsaft kauft eine Empfehlung für eine 0,51 Flasche bekommt. Da sich der Kunde jedoch aktiv für eine 0,331 Flasche entschieden hat, erscheint es unwahrscheinlich, dass er eine 0,51 Flasche des selben Herstellers kaufen möchte.

Lösungsansatz

Dieses Problem lässt sich jedoch leicht beheben, indem man die Sortimentsebenen bei den Empfehlungen mit betrachtet. Dazu müssen für jedes Produkt alle Sortimentsebenen spezifiziert und für das System einsehbar sein. Es wird determiniert bis zu welcher Sortimentsebene bei den Empfehlungen unterschieden werden soll (vgl. folgender Abschnitt). Wird zum Beispiel bis zur Ebene "Artikel" unterschieden, bedeutet das, dass zwei Produkte dann als unterschiedlich angesehen werden, wenn ihnen unterschiedliche Werte der Ebene "Artikel" zugewiesen sind. Wird eine Empfehlung ausgesprochen und es befindet sich bereits ein Produkt der gleichen Artikelebene im Einkaufswagen des Kunden, so wird die Empfehlung verworfen.

In "Mining Generalized Association Rules" von Ramakrishnan Srikant und Rakesh Agrawal [19] werden Algorithmen vorgestellt, die es ermöglichen verallgemeinerte Assoziationsregeln zu finden. Hierzu wird eine Taxonomie (= Klassifikation) betrachtet, die eine hierarchische Ordnung auf den Produkten einführt. Verallgemeinerte Assoziationsregeln erstrecken sich über mehrere Ebenen der Taxonomie. So kann zum Beispiel aus den Regeln "Milch \rightarrow Müsli" und "Joghurt \rightarrow Müsli" die verallgemeinerte Regel "Milchprodukte \rightarrow Müsli" generiert werden. Deren Support muss jedoch nicht die Summe der Supports der zwei speziellen Regeln sein, da es auch Kunden geben kann, die Milch, Joghurt und Müsli zusammen kaufen. Allerdings können auf diese Weise mehr Assoziationen gefunden werden, da der Support der speziellen Regeln eventuell nicht über dem minimalen Support liegt und die Regeln somit verworfen werden. Die verallgemeinerten Regeln bieten einen Ansatz zur Umsetzung der Sortimentsebenen, welche eine Taxonomie darstellen.

Möglichkeit zur Verifizierung

Eine Unterscheidung auf der Ebene "Artikel" erscheint sinnvoll, da so gleiche Produkte von unterschiedlichen Herstellern als nicht unterschiedlich angesehen werden. Dies ergibt Sinn, da gleiche Produkte meist im gleichen Regal stehen und der Kunde somit beim Kauf eines Produktes von Hersteller A ebenso das Produkt von Hersteller B gesehen hat und sich dagegen entschieden hat. Um die Sortimentsebene festlegen zu können sollte jedoch eine Studie geführt werden, die im Rahmen der vorliegenden Arbeit nicht umsetzbar ist.

Für die Studie müssen die Sortimentsebenen spezifiziert werden, die Unterscheidung jedoch auf der Sortimentseben "Sorte" bleiben. Es soll geprüft werden ob eine Unterscheidung der Produkte bis Sortimentsebene X sinnvoll ist. Nun wird bei jeder Empfehlung analysiert, ob bereits Produkte im Einkaufswagen liegen, denen die gleiche Ebene X zugewiesen ist, wie der Empfehlung. Aus diesen Daten lässt sich ableiten, wie oft Empfehlungen vom Kunden abgelehnt wurden, wenn sich bereits ein Produkt mit der gleichen Ebene X im Einkaufswagen befindet, und wie oft sie angenommen wurden. Mit diesem Wissen wird entschieden ob die Unterscheidung bis Ebene X sinnvoll ist oder eine höhere oder tiefere Ebene getestet werden soll.

8.2 Allgemeine Empfehlungen

Bei kleinen oder ungewöhnlichen Einkäufen kann es vorkommen, dass der regelbasierte Empfehlungsdienst keine Empfehlungen aussprechen kann, da keine passenden Regeln vorhanden sind. Ebenso kann es sein, dass nicht die maximale Anzahl an Empfehlungen gegeben wird und noch Platz für weitere Empfehlungen ist. In beiden Fällen bleibt die Möglichkeit eine allgemeine Empfehlung auszusprechen. Dies kann im wesentlichen auf zwei Arten geschehen, über den Verkaufsschlager oder über Produkte, die neu im Sortiment sind.

Verkaufsschlager

Ein Ansatz ist die Empfehlung des Verkaufsschlagers, also des am besten verkauften Produktes aus dem Automaten. Dies lässt sich leicht umsetzen, wenn gespeichert wird wie häufig ein Produkt vom System empfohlen und gekauft wird. Es kann das Produkt empfohlen werden, das häufig empfohlen wird und in den meisten Fällen anschließend auch gekauft wird. So ist gewährleistet, dass das Produkt für viele Kunden interessant ist und von den meisten dieser Kunden auch angenommen wird. Da auch die Quote betrachtet wird, wie oft das Produkt angenommen wird, entscheidet das System automatisch, ob der bisherige Verkaufsschlager bleibt oder durch zu häufiges Ablehnen von einem anderen Produkt abgelöst wird.

Neu im Sortiment

Ein weiterer Ansatz ist die Empfehlung von neuen Produkten. Diese können durch den Händler nachträglich in das geplante Teilsortiment eingefügt und als neu im Sortiment definiert werden. Sie werden immer vorgeschlagen, wenn noch Platz für weitere Empfehlungen ist. Dadurch wird sichergestellt, dass immer mindestens eine Empfehlung ausgesprochen wird. Zusätzlich wird das New-Item Problem [20] umgangen. Wenn ein Produkt neu im Sortiment ist, wurde es vorher noch nie gekauft und ist somit nicht in den Einkaufszetteln der vergangenen Zeit zu finden. Da die Regeln für den Empfehlungsdienst durch die Analyse der vergangenen Einkaufszettel ermittelt werden 4.1, gibt es keine Regel, die das neue Produkt empfiehlt. Das Empfehlen der neuen Produkte macht die Kunden aufmerksam, was dazu führen kann, dass mehr Kunden das Produkt kaufen und es so schneller für folgende Teilsortimentsplanungen berücksichtigt wird.

8.3 Anpassen des Teilsortiments

Ein weiterer, interessanter Punkt ist die Frage, ob das gewählte Teilsortiment wirklich optimal ist oder ob es nachträglich angepasst werden muss. Dieses Problem kann von zwei verschiedenen Seiten betrachtet werden. Entweder werden einzelne Produkte des Sortiments durch andere ersetzt oder das Teilsortiment wird neu berechnet, wenn es nicht optimal ist. In beiden Fällen dient eine Studie als Entscheidungshilfe, ob das Teilsortiment angepasst werden soll. Es wird aufgezeichnet, wie oft jedes Produkt empfohlen wird und wie oft es auf Grund dessen gekauft wird. Wird ein Produkt selten empfohlen oder trotz Empfehlung selten gekauft, so wird das Teilsortiment angepasst, da angenommen wird, dass es eine bessere Belegung gibt.

Teilweises Ersetzen

Angenommen die auszutauschenden Produkte sind durch die Studie ermittelt worden, dann bleibt noch die Frage, wie geeignete Produkte zum Austauschen bestimmt werden. Eine Möglichkeit bietet der Planungsalgorithmus, mit dem das Teilsortiment geplant wurde (vgl. Abschnitt 5.2). Die Implementierung des Planungsalgorithmus kann unabhängig genutzt und die Klasse EasyScoreCalculation, welche die Bewertung berechnet, leicht angepasst werden. Die Assoziationsregeln, die für die Planung des Sortiments genutzt werden, sollten auch für die Anpassung des Sortiments genutzt werden. Um das

angepasste Sortiment zu planen wird der Planungsalgorithmus leicht abgewandelt mit den vorhanden Assoziationsregeln gestartet. Dabei werden die Produkte, die im Automaten bleiben sollen, als feststehend definiert. Das bedeutet, ein Plan ist nur dann relevant, wenn alle der feststehenden Produkte in ihm enthalten sind, jedoch keines der auszutauschenden Produkte. Dies kann durch die Hardscore umgesetzt werden, indem diese um eins verringert wird, wenn sich ein auszutauschendes Produkt im Automaten befindet oder eins der feststehenden Produkte fehlt. Dadurch wird die Hardscore negativ und die Lösung ungültig. Je nachdem ob mehr Wert darauf gelegt wird feststehende Produkte zu behalten oder auszutauschende Produkte zu entfernen, kann eines der Kriterien (oder beide) über die Softscore definiert werden. Die Position der feststehenden Produkte kann sich ändern, wenn dadurch insgesamt eine bessere Belegung erzielt wird. Innerhalb einer festgelegten Zeit ermittelt der Planungsalgorithmus einen Plan und optimiert diesen. Wenn der Algorithmus ein Ergebnis findet, so ist dies eine Belegung für den Automaten, in der die feststehenden Produkte beibehalten wurden und die auszutauschenden durch bestmögliche Alternativen ersetzt wurden.

Neuberechnung

Eine genaue Angabe über die Zeit, die benötigt wird um ein Teilsortiment zu planen, kann nicht getroffen werden. Hierzu fehlen in den vorliegenden Daten die Größen der Produkte. Somit entsprechen die wenigen Minuten, die der Planungsalgorithmus benötigt, eventuell nicht der Berechnungszeit die er unter Berücksichtigung der Größen benötigt. Dennoch kann mit Sicherheit gesagt werden, dass sowohl das Laden der Transaktionen, als auch das Ausführen des Apriori Algorithmus einige Zeit dauert. Bei einer Transaktionsmenge mit ca. 300.000 Transaktionen und ca. 55.000 Produkten benötigt das Laden der Daten und die Berechnung des Teilsortiments zusammen wenige Minuten. Bei ca. 900.000 Transaktionen und 80.000 Produkten liegt die Zeit, bei etwa 30 Minuten. Diese Daten sind hardwareabhängig, geben jedoch einen groben Überblick.

8.4 Produktergänzende Empfehlungen

Auch beim Kauf von Backmischungen kann ein Empfehlungsdienst nützlich sein. Zu einer Backmischung werden einige ergänzende Produkte benötigt, um das entsprechende Rezept umzusetzen. Da diese Produkte auf der Packung aufgelistet sind und nicht auf dem persönlichen Einkaufszettel stehen, können sie leicht vergessen werden.

Um dies zu verhindern sollte für jede Backmischung und alle ähnlichen Produkte, die Ergänzungen benötigen, eine Datenbank angelegt werden. In dieser sollte vermerkt sein welche Produkte zu den jeweiligen Produkten gekauft werden müssen. Mit diesem Wissen, welches aus von Hand angelegten Implikationsregeln besteht, kann ein regelbasierter Empfehlungsdienst ableiten, ob der Kunde eventuell ein Produkt vergessen hat. Dieses kann der Empfehlungsdienst mit den anderen Empfehlungen an der Kasse anzeigen, mit dem Hinweis, dass das Produkt für die Backmischung nötig ist.

Es ist eher unwahrscheinlich, dass diese Produkte im Teilsortiment des Automaten vorhanden sind, da es sich meist um Milchprodukte oder Eier handelt, die nicht für den Automaten geeignet sind. Der Kunde muss also zu dem Regal zurück laufen, um das Produkt zu bekommen. Da das Produkt notwendig ist für die gekaufte Backmischung,

ist dieser Aufwand jedoch gerechtfertigt.

Eine Alternative bieten die Einkaufswagen des IRL. In diese ist die Möglichkeit integriert elektronische Einkaufslisten über einen Bildschirm am Wagen anzuzeigen. Sobald ein Produkt von der Einkaufsliste in den Wagen gelegt und über das RFID-Funketikett erkannt wird, wird es von der Liste entfernt. Beim Einlegen einer Backmischung könnten die ergänzenden Produkte automatisch aus der Datenbank abgerufen und vorgeschlagen werden. Der Kunden hat dann die Möglichkeit einzelne oder alle Produkte auf die elektronische Einkaufsliste zu übernehmen. So kann der Kunde entscheiden welche Produkte er noch zu Hause hat und welche er benötigt. Außerdem wird der Kunde direkt aufmerksam auf die fehlenden Produkte und die Wahrscheinlichkeit, dass er eines davon vergisst, wird geringer.

8.5 Weitere Evaluationen

Mit umfangreicheren Daten lassen sich weitere Evaluationen führen, die einem realen Szenario ähnlicher sind. Mit einer Zuordnung zwischen Produktnamen und EANs können geeignetere Filter eingeführt werden. Eventuell ist es sinnvoll das Teilsortiment auf Produkte zu reduzieren, die zum Spontankauf verleiten. Das bedeutet grundlegende Produkte, wie Mehl und Zucker, ersetzen durch zum Beispiel Süßigkeiten.

Durch die Namen können einige Produkte zu Kategorien zusammengefasst oder sogar die Sortimentsebenen spezifiziert werden. Dies benötigt jedoch einige Zeit und eine geeignete Kategorienhierarchie. Eventuell kann die Hierarchie des IRL übernommen und erweitert werden. Diese besteht zur Zeit aus 605 Kategorien, wobei davon 232 auf der Ebene Artikel zu finden sind.

Außerdem ist die Frage interessant, ob und wie sich das Kaufverhalten der Kunden in den verschiedenen Jahreszeiten ändert. Wenn sich das Verhalten ändert, sollte das Teilsortiment des Automaten angepasst werden. Stehen aufgezeichnete Einkäufe über ein oder mehrere Jahre zur Verfügung, kann geprüft werden, ob ein gewähltes Teilsortiment über das ganze Jahr aktuell ist. Hierbei kann eine Simulation hilfreich sein. Die Simulation könnte einen Offline-Handel widerspiegeln, in dem das System der vorliegenden Arbeit integriert ist. Dabei würde das System Kunden durch aufgezeichnete Einkäufe simulieren und Empfehlungen geben, die sie, abhängig von einigen Faktoren, zufällig annehmen oder ablehnen. Ein möglicher Faktor, der in die Entscheidung eingehen kann, ist die Tageszeit. So haben Kunden die während der Woche vormittags einkaufen, wahrscheinlich mehr Zeit und Geduld, als Kunden, die zur Feierabendzeit (16-18 Uhr) einkaufen.

Diese Ideen der Evaluation sind im Rahmen der vorliegenden Arbeit zeitlich nicht umsetzbar. Außerdem werden genauere Informationen für die einzelnen Produkte benötigt. Jedoch verspricht eine genauere Analyse der Daten interessante Ergebnisse. Gerade für die GLOBUS SB-Warenhaus Holding sollten diese Ergebnisse interessant sein, da sie aus ihren Daten ermittelt wurden.

8.6 Werbemöglichkeiten

Da die Idee der vorliegenden Arbeit neuartig ist und so noch nicht umgesetzt wurde, kann es sein, dass Kunden das System nicht annehmen. Durch einfache Werbeaktionen kann das System dem Kunden näher gebracht werden. Hier bieten sich zwei Ansätze an.

Der erste und einfachste Weg ist das Verschenken empfohlener Artikel. Das bedeutet, dass ein Kunde, dem ein Produkt empfohlen wird, die Möglichkeit hat dieses als Geschenk anzunehmen. Dadurch wird der Kunde positiv auf das neue System aufmerksam und kann sich von der Qualität der Empfehlungen überzeugen. Ist der Kunde erst an die Empfehlungen gewöhnt, wird die Entscheidung ein empfohlenes Produkt zu kaufen einfacher fallen.

Die zweite Variante sind Kombi-Käufe. Das bedeutet, dass einige Angebote an die Empfehlungen gekoppelt sind. So bekommt ein Kunde beim Kauf eines empfohlenen Produktes aus dem Automaten einen Rabatt auf dieses. Das motiviert die Kunden sich die Empfehlungen anzuschauen, da sie auf ein Angebot hoffen.

Danksagung

An dieser Stelle möchte ich allen danken, die mich bei dieser Arbeit unterstützt haben. Ganz besonders meinem Betreuer Christian Bürckert, der mich über die ganze Zeit auf seine Art motiviert hat und mir bei allen Problemen zur Seite stand. Außerdem möchte ich dem DFKI, speziell Prof. Dr. Antonio Krüger, und der GLOBUS SB-Warenhaus Holding danken, die mir die nötigen Mittel zur Verfügung gestellt und so die Arbeit möglich gemacht haben. Darüber hinaus danke ich Freunden und Familie, die meine Arbeit korrekturgelesen haben und mich durch Ablenkung unterstützt haben. Zu guter Letzt möchte ich auch meinen Kommilitonen danken, die mich (meist nur zeitweise) in meinem Studium begleitet haben.

Saarbrücken 30. September 2015

Eva Maria Gressung

9 Appendix

9.1 Evaluationsergebnisse

Die Analyse der vorliegenden Daten ergibt, dass die Produkte aus der folgenden Tabelle in den Teilsortimenten auftreten. Diese werden im Folgenden jeweils mit der ID aus der Tabelle referenziert. Wie die Sortimente bei bestimmten Spezifikationen aussehen und wie häufig die Produkte durch den regelbasierten Empfehlungsdienst empfohlen werden, zeigen die folgenden Abschnitte. Der Empfehlungsdienst wird für Transaktionen aus einem, zwei und drei Monaten getestet.

| ID | EAN | Produktname |
|----|---------------|--|
| 1 | 2001609606004 | Roggenmischbrot |
| 2 | 2001609644006 | Baguette |
| 3 | 2001609657037 | Unbekannt (wahrscheinlich Brot) |
| 4 | 4000007018301 | Crex Nektarinen |
| 5 | 4003490035121 | gut & fein Deutsche Markenbutter |
| 6 | 4003490037200 | Hochwald Milchwerke Schlagsahne Pappbox 200 |
| 7 | 4008400401928 | Ferrero - Nutella, Nuss-Nougat Creme, 800 g |
| 8 | 4009044152122 | Schlaufentragetasche Globus Chips |
| 9 | 4009044153396 | Schlaufentragetasche Globus Eistee |
| 10 | 4251003507225 | Unbekannt |
| 11 | 4260079661038 | Saarlandeier 10er Gr. L Bodenhaltung |
| 12 | 4304218408839 | korrekt Emmentaler, gerieben, 45% |
| 13 | 4304218409119 | korrekt Rollsaftschinken, Hunsrücker Art |
| 14 | 4304218417367 | korrekt Mozzerella classic |
| 15 | 4304218417534 | korrekt Eier, Bodenhaltung, Gr. M |
| 16 | 4304218417558 | korrekt Eier, Freilandhaltung, Gr. M-L |
| 17 | 4304218418432 | korrekt Schmand, 24 % Fett |
| 18 | 4304218418494 | korrekt Speisequark, Magerstufe |
| 19 | 4304218418593 | korrekt Schlagsahne, frisch, 30 % Fett |
| 20 | 4304218418616 | korrekt Frische Voll-Milch, 3,5 % Fett |
| 21 | 4304218418630 | korrekt Frische fettarme Milch, 1,5 % Fett |
| 22 | 4304218419255 | korrekt H-Milch, haltbar, 1,5 % Fett |
| 23 | 4304218608536 | natuvell Toilettenpapier Premium, 4-lagig |
| 24 | 4304218608550 | korrekt Küchentücher, 3-lagig |
| 25 | 4304218608659 | natuvell Taschentücher Box, 4-lagig |
| 26 | 4304218608673 | natuvell Toilettenpapier, Classic, 3-lagig |
| 27 | 4304218610447 | natuvell Toilettenpapier, Recycling, 3-lagig |
| 28 | 4304218713469 | korrekt Raffinade Zucker, feine Körnung |
| 29 | 4304218713506 | korrekt Weizenmehl Type 405 |

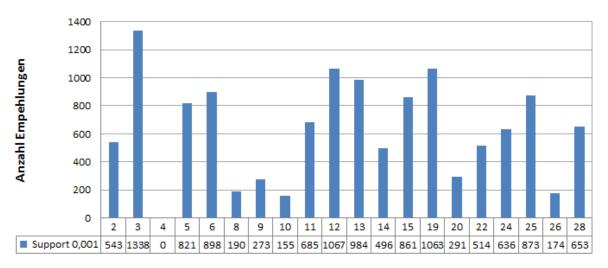
1 Monat

| Anzahl Einkaufszettel | 292.383 |
|--------------------------------------|---------|
| Anzahl Produkte | 58.160 |
| Anzahl Fächer | 20 |
| Anzahl Einkaufszettel zur Evaluation | 13281 |

 $\begin{array}{c} \text{Support} & 0,001 \\ \text{Konfidenz} & 0,0 \end{array}$

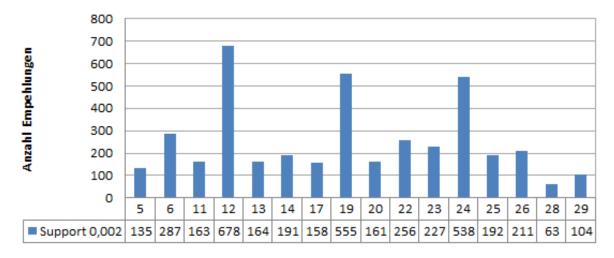
Anzahl der Einkäufe mit mind. einer Empfehlungen 2.668 (20%) Anzahl der Empfehlungen insgesamt 12.515 (94%)

Belegte Fächer 20



Produkt ID

Support 0,002 Konfidenz 0,0 Anzahl der Einkäufe mit mind. einer Empfehlungen 1.533 (11,5%)Anzahl der Empfehlungen insgesamt 4.083 (30,5%)Belegte Fächer 16



Produkt ID

2 Monate

■ Support 0,001

| Anzahl Einkaufszettel | 628.744 |
|--------------------------------------|---------|
| Anzahl Produkte | 73.395 |
| Anzahl Fächer | 20 |
| Anzahl Einkaufszettel zur Evaluation | 13.281 |
| | |
| Support | 0.001 |

Support 0,001 Konfidenz 0,0

Anzahl der Einkäufe mit mind. einer Empfehlungen 2.658 (20%)Anzahl der Empfehlungen insgesamt 13.454 (101,5%)Belegte Fächer

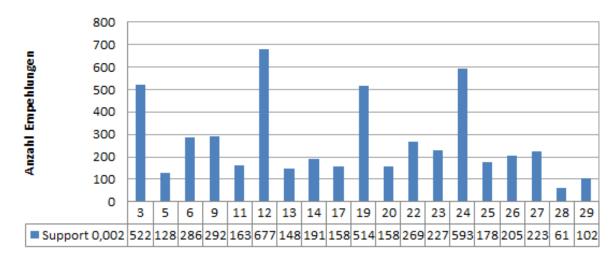
Anzahl Empehlungen

Produkt ID

748 395 765 1042 152 646 939 203 1022 1123 1327 167

| Support | 0,002 |
|--|-------------------|
| Konfidenz | 0,0 |
| Anzahl der Einkäufe mit mind. einer Empfehlungen | $2.182\ (16,5\%)$ |
| Anzahl der Empfehlungen insgesamt | 5.095 (38,5%) |
| Belegte Fächer | 19 |

543 1289 859 802



Produkt ID

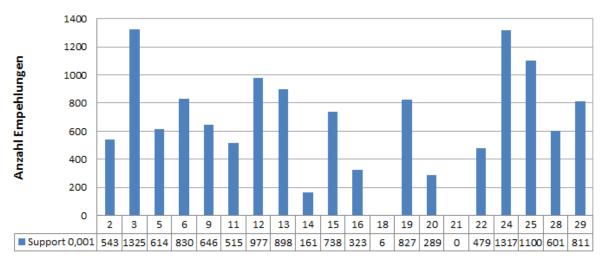
3 Monate

| Anzahl Einkaufszettel | 946.642 |
|--------------------------------------|---------|
| Anzahl Produkte | 84.317 |
| Anzahl Fächer | 20 |
| Anzahl Einkaufszettel zur Evaluation | 13.281 |

 $\begin{array}{c} \text{Support} & 0,001 \\ \text{Konfidenz} & 0,0 \end{array}$

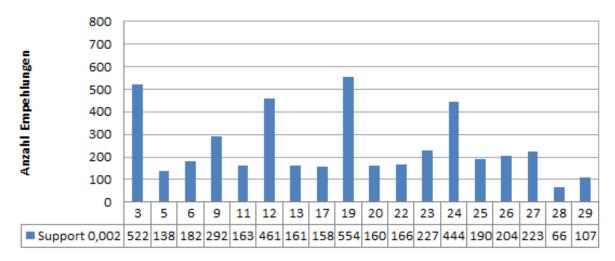
Anzahl der Einkäufe mit mind. einer Empfehlungen 2.687 (20%)
Anzahl der Empfehlungen insgesamt 13.000 (98%)

Belegte Fächer 20



Produkt ID

Support 0,002 Konfidenz 0,0 Anzahl der Einkäufe mit mind. einer Empfehlungen 2.157 (16%)Anzahl der Empfehlungen insgesamt 4.418 (33%)Belegte Fächer 18



Produkt ID

9.2 European Article Number

Die European Article Number (EAN) ist eine 8 bzw. 13 stellige Zahl, die Handelsprodukte international eindeutig kennzeichnet. Die ersten sieben bis neun Ziffern sind die Basisnummern bestehend aus einem dreistelligen Länderpräfix (400-440 für Deutschland) und einer Unternehmensnummer. Es folgen drei bis fünf Ziffern für die Artikelbezeichnung. Je nachdem wie lang die Basisnummer ist variiert die Länge der Artikelnummer, so dass die EAN insgesamt 13 Stellen hat. Die letzte Ziffer ist eine Prüfziffer.

Im Einzelhandel kann der Länderprafix durch ein spezielles Präfix ausgetauscht werden. Dieses wird intern benutzt, um vor Ort abgewogene Waren wie Obst, Gemüse und Fleischwaren zu kennzeichnen. Die EAN besteht dann aus "2xx" anstatt der Ländernummer, dann folgen vier Stellen für die Artikelnummer und fünf Stellen für Gewicht, Menge oder Preis.¹

¹https://de.wikipedia.org/wiki/European_Article_Number

Literaturverzeichnis

- [1] Dettling, Selina: Verschiedene Arten und Einsatzmöglichkeiten von Bewertungssystemen sowie die Anwendung im mki-Web 3.0, 2014.
- [2] SARWAR, BADRUL, GEORGE KARYPIS, JOSEPH KONSTAN und JOHN RIEDL: Analysis of Recommendation Algorithms for E-Commerce. In: Proceedings of the 2nd ACM conference on Electronic commerce. ACM, 2000.
- [3] Voss, Jakob: 2. Formen von Empfehlungsdiensten, 2008.
- [4] RESNICK, PAUL und HAL R. VARIAN: <u>Recommender Systems</u>. Communications of the ACM, 40(3):56–58, 1997.
- [5] KUZDAS, SABINE: Schwachstellen von Recommender Systemen, 2005.
- [6] AGRAWAL, RAKESH und RAMAKRISHNAN SRIKANT: <u>Fast Algorithms for Mining Association Rules</u>. In: <u>Proc. 20th int. conf. very large data bases, VLDB</u>, Band 1215, Seiten 487–499, 1994.
- [7] NAJADAT, HASSAN M, MOHAMMED AL-MAOLEGI und BASSAM ARKOK: <u>An Improved Apriori Algorithm for Association Rules</u>. International Research Journal of Computer Science and Application, 1(1):01–08, 2013.
- [8] SINGH, JAISHREE, HARI RAM und DR. J.S. SODHI: <u>Improving Efficiency of Apriori Algorithm Using Transaction Reduction</u>. International Journal of Scientific and Research Publications, 3(1):1–4, 2013.
- [9] DUTT, SHALINI, NAVEEN CHOUDHARY und DHARM SINGH: An Improved Apriori Algorithm based on Matrix Data Structure. Global Journal of Computer Science and Technology, 14(5), 2014.
- [10] PAZZANI, MICHAEL J.: A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review, 13(5-6):393–408, 1999.
- [11] Basilico, Justin und Thomas Hofmann: <u>Unifying Collaborative and Content-Based Filtering</u>. In: <u>Proceedings of the twenty-first international conference on Machine learning</u>, Seite 9. ACM, 2004.
- [12] LEE, JOONSEOK, MINGXUAN SUN und GUY LEBANON: A Comparative Study of Collaborative Filtering Algorithms. arXiv preprint arXiv:1205.3193, 2012.
- [13] SARWAR, BADRUL, GEORGE KARYPIS, JOSEPH KONSTAN und JOHN RIEDL: Item-Based Collaborative Filtering Recommendation Algorithms. In: Proceedings of the 10th international conference on World Wide Web, Seiten 285–295. ACM, 2001.

- [14] HOHFELD, STEFANIE und MELANIE KWIATKOWSKI: Empfehlungssysteme aus informationswissenschaftlicher Sicht-State of the Art. Information Wissenschaft und Praxis, 58(5):265, 2007.
- [15] Muno, Stefan: <u>Recommender Systeme</u>. Seminararbeit am Lehrstuhl für Informationsysteme der Universität Duisburg-Essen, 2008.
- [16] Suhl, Uwe H: IT-gestützte, operative Sortimentsplanung. In: IT-gestützte betriebswirtschaftliche Entscheidungsprozesse, Seiten 175–194. Springer, 2001.
- [17] Syring, Dietmar: Bestimmung effizienter Sortimente in der operativen Sortimentsplanung. Doktorarbeit, Freie Universität Berlin, Germany, 2004.
- [18] ZENTES, JOACHIM, BERNHARD SWOBODA und THOMAS FOSCHT: Handelsmanagement. Verlag Franz Vahlen GmbH, 2012.
- [19] SRIKANT, RAMAKRISHNAN und RAKESH AGRAWAL: Mining Generalized Association Rules, 1995.
- [20] Spitzer, Ursula: Recommender Systeme im E-Commerce. Bachelor Thesis, Wirtschaftsuniversität Wien, 2005.
- [21] RASHID, AL MAMUNUR, ISTVAN ALBERT, DAN COSLEY, SHYONG K. LAM, SEAN M. McNee, Joseph A. Konstan und John Riedl: Getting to Know You: Learning New User Preferences In Recommender Systems. In: Proceedings of the 7th international conference on Intelligent user interfaces, Seiten 127–134. ACM, 2002.
- [22] VAN METEREN, ROBIN und MAARTEN VAN SOMEREN: <u>Using Content-Based</u>
 <u>Filtering for Recommendation</u>. In: <u>Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, Seiten 47–56, 2000</u>.
- [23] WAKABI-WAISWA, PETER P. und VENANSIUS BARYAMUREEBA: Extraction of Interesting Association Rules Using Genetic Algorithms. International Journal of Computing and ICT Research, 2(1):26–33, 2008.
- [24] BOLLINGER, TONI: Assoziationsregeln–Analyse eines Data Mining Verfahrens. Informatik-Spektrum, 19(5):257–261, 1996.
- [25] AGRAWAL, RAKESH, TOMASZ IMIELIŃSKI und ARUN SWAMI: Mining Association Rules between Sets of Items in Large Databases. In: ACM SIGMOD Record, Band 22, Seiten 207–216. ACM, 1993.
- [26] COSKUN, MEHMET E: Shelf Space Allocation: A Critical Review and a Model with Price Changes and Adjustable Shelf Heights, 2012.