



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# **Hausarbeit**

**Eduard Weigandt**

**Master Seminar**

Eduard Weigandt

**Master Seminar**

Hausarbeit eingereicht im Rahmen der Vorlesung MS

im Studiengang Master of Science Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüfer: Prof. Dr. Kai von Luck & Prof. Dr. Bettina Buth

Eingereicht am: 28. Februar 2015

## **Eduard Weigandt**

### **Thema der Arbeit**

Master Seminar

### **Stichworte**

Visualisierung, Visual Analytics, Konzepte, Statistik, Kontext-basierte Empfehlungen, Hybrid-Empfehlungssystem, Features, Factorized-Matrix, Content-based, Collaborative-based, Constraints, Master Outline

### **Kurzzusammenfassung**

In der hier vorliegenden Arbeit wird ein *Master Thesis Outline* für eine mögliche Masterarbeit im Themenbereich der Empfehlungs-Systeme vorgeschlagen. Die Basis besteht aus der Problembeschreibung und der Machbarkeit des Themas anhand von recherchierter Literatur. Der wissenschaftliche Wert wird durch relevanten Fragestellungen festgelegt. Diese werden dann mit einer bestimmten Methodik, während der Umsetzungszeit auf ihre Richtigkeit untersucht. Die Evaluierung erfolgt durch Prüfen der aufgezählten Ziele und Fragen.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Problemdefinition . . . . .	1
1.2	Motivation und Möglichkeiten . . . . .	3
<b>2</b>	<b>Empfehlungs-Systeme</b>	<b>4</b>
2.1	Verfahren und Literatur . . . . .	4
2.1.1	Collaborative Filtering . . . . .	4
2.1.2	Content-based . . . . .	6
2.1.3	Knowledge-based . . . . .	6
2.1.4	Hybrid . . . . .	7
2.1.5	Kontext . . . . .	8
<b>3</b>	<b>Methodik</b>	<b>9</b>
3.1	Umsetzung . . . . .	9
3.2	Evaluierung . . . . .	11

# 1 Einführung

In der hier verfassten Ausarbeitung werden jeweils wichtige Aspekte zur Umsetzung einer Masterarbeit der Domäne von Empfehlungs-Systemen vorgestellt. Als Grundlage werden alle vorangegangenen Ausarbeitungen aus *AW1* und *AW2* im Bereich des *Visual Analytics* vorausgesetzt. Zu den vorgestellten Aspekten zählen einmal die akademischen Fragen, die das wissenschaftliche Arbeiten vorantreiben sowie die zu untersuchenden Hauptthesen. Dadurch soll ein klares Bild vermittelt werden, was für Aufgaben in der Masterarbeit zu bearbeiten sind. Durch die Vorarbeit in *AW1* und *AW2* liegt hier der Fokus mehr auf der Einführung in die Empfehlungs-Systeme.

**Aufbau der Arbeit** Der Aufbau dieser Ausarbeitung gestaltet sich wie folgt. Zu aller erst sind die Fragen definiert und die Motivation dahinter diese in Form einer Masterarbeit nachzugehen. Danach wird eine Einführung in den thematischen Kontext der Empfehlungs-Systeme gegeben, die den derzeitigen Stand der Dinge in Verbindung mit der relevanten Literatur markieren. Darüber hinaus werden auch die bekannten Risiken aufgezählt. Zum Schluss wird die eigentliche Methodik während der Umsetzung beschrieben.

## 1.1 Problemdefinition

Heutzutage gibt es fast keine Webseite im Internet, die nicht auf die eine oder andere Art der Empfehlung zurückgreift. In Zeiten von Big-Data sehen sich die Menschen dem Problem einer großen Datenflut gegenübergestellt. Die verschiedensten Dienste bieten Information oder Produkte an, die von Kunden konsumiert werden können. Die Online-Händler z.B. wollen, dass ihre Kunden die richtigen Vorschläge bekommen um die Wahrscheinlichkeit eines Kaufs zu erhöhen. Es gibt eine große Anzahl an Verfahren mit denen man dieses Problem in Angriff nehmen kann. Dabei sind sowohl bestimmte Balanceakte zwischen sich untereinander ausschließenden Eigenschaften und Zielen zu beachten. Als auch viele Arten diese einzelnen Verfahren miteinander zu kombinieren. Die Schwierigkeiten bestehen in der Balance zwischen z.B. der Genauigkeit oder Vielfalt in den Empfehlungen. Je nachdem ob der Benutzer zum Zeitpunkt etwas bestimmtes sucht oder einfach nur explorativ die Seite durchstöbert. Eine weitere wichtige Eigenschaft ist, wenn man ausschließlich populäre anstatt neuer Artikel vorschlägt. Bei den populären Artikeln besteht eine hohe Wahrscheinlichkeit, dass diese von dem derzeitigen Benutzer für relevant befunden werden. Man verpasst jedoch die Chance neue Waren anzupreisen, wenn der aktuelle Benutzer eigentlich was anderes in seinem Kontext (Adomavicius u. Tuzhilin, 2011) sucht.

**Ziele** Die Entwicklung eines ausbalancierten und lernenden Empfehlungs-Systems ist eine komplexe Aufgabe. Ich möchte die Erstellung eines solchen Systems durch den Einsatz von verschiedenen *Visual Analytics* Techniken vereinfachen. Im groben soll ein Werkzeug entwickelt werden, welches bei der Erstellung eines Empfehlungs-Systems und der Kombination von bekannten Techniken unterstützende Hilfe leisten soll. Ein Ziel dabei ist es die Bewertung und den Vergleich von eingesetzten Verfahren zu erleichtern. Ein großer Teil davon soll durch passende Visualisierungen ermöglicht werden, die auf Basis von Mengenbeziehungen die berechneten Relationen aufzeigen sollen. So soll die "Blackbox" eines Empfehlungs-Systems aufgebrochen werden, um ein besseres Verstehen über die Algorithmen und die Nutzer zu ermöglichen. Bei der Bewertung des Systems spielt das eine wichtige Rolle, weil dadurch das explorative Erforschen der berechneten Daten ermöglicht werden kann. Die verwendeten Verfahren zur Berechnung von Empfehlungen sollen bestimmte Feature<sup>1</sup> der Produkte sowie den Kontext<sup>2</sup> der Benutzer berücksichtigen. Das Ergebnis wird ein Hybrid-System (Burke, 2002) sein, welches *Collaborative Filtering* (Sarwar u. a., 2001) und *Content-based Filtering* (Pazzani u. Billsus, 2007) vereint. Durch die Wahl der Features und des Nutzerkontextes kann man leicht unterschiedliche Anpassungen mit dem System testen. Dadurch soll die Untersuchung der Relevanz einzelner berücksichtigter Aspekte für die Empfehlungen vereinfacht werden. Zu dem soll dabei die Theorie von Hybrid-Empfehlungs-Systemen in Verbindung mit *Visual Analytics* betrachtet werden. Speziell gesehen das Design im Aufbau und in der Komposition der eingesetzten Verfahren eines solchen Systems.

**Fragen** Wie schon erwähnt ergeben sich viele Fragen, wie man ein passendes Empfehlungs-System für die gesteckten Ziele und Aufgaben entwickelt. Dazu zählt z.B. welche Ansätze für die Berechnung von Empfehlungen gewählt werden und wie gut die erbrachten Ergebnisse gegenüber dem alten System sind. Diese Fragen sollen durch das zu entwickelnde Werkzeug einfacher zu beantworten sein. Nachdem man eine Auswahl von Verfahren bestimmt hat, die man im Zusammenspiel untersuchen will, muss man eine geeignete Visualisierung finden, die zu einem besseren Verstehen der Arbeitsweise führen soll. Welche Visualisierung sich für diesen Zweck am besten eignet muss dabei näher evaluiert werden. Die erbrachten Erkenntnisse im Zusammenspiel sollen dann iterativ in den Erstellungs-Prozess zurückfließen, um z.B. ausgewählte Techniken zu verwerfen oder anzupassen. Es wäre von Interesse einen definierten Analyseprozess zu finden und etablieren, der dem Domänen-Experten hilft eigens aufgestellte Hypothesen über das Kundenverhalten in Verbindung mit den Algorithmen zu testen. Anschließend muss die Bewertung des Werkzeugs und die damit erzielten Ergebnisse im eigentlichen Empfehlungs-

---

<sup>1</sup>Beispiel Feature: Preis, Artikeltyp, Genre, etc.

<sup>2</sup>Beispiel: Alter, Jahreszeit, Geschlecht, Mausbewegungen etc...

System erfolgen. Um eine klare Aussage treffen zu können sind dafür Schlüsselmerkmale zu definieren, wie die Zufriedenheit eines Kunden über die präsentierten Empfehlungen unter Einbeziehung seines Kontextes bzw. des eigentlichen Vorhabens. Dies alleine reicht jedoch nicht aus, weil der Domänen-Experte eine Einschätzung treffen muss inwieweit er seine Aufgaben besser oder schlechter durch den Einsatz des Werkzeugs erfüllen konnte. Traditionell werden Empfehlungs-Systeme anhand von Trainingsdaten oder statistischen Relevanz in ihrer Erfolgsquote untersucht. (Jannach u. a., 2010) Es stellt sich die Frage: "Wie bewertet man das aktuelle System?" Wenn ein Domänen-Experte das Arbeiten mit dem Werkzeug als hilfreich empfindet, der Kunde jedoch am Ende unzufrieden ist. Außerdem inwieweit steht eine Änderung am System und das gemessene Ergebnis am Ende im Zusammenhang miteinander?

**Hypothesen** In der Masterarbeit sollen die folgenden Punkte untersucht werden. Die Feature-basierte Empfehlungserstellung kann durch das Anreichern von Kontext ein besseres Ergebnis in Form von einer höheren Zufriedenheit der Nutzer liefern. Die Analyse und Bewertung des Designs in diesem Beispielsystem soll mit Hilfe von *Visual Analytics* vereinfacht werden. Um dieses Ziel umzusetzen sollen die Daten explorativ erforschbar gemacht werden, um neue Erkenntnisse zu gewinnen und mögliche Verbesserungen zu identifizieren. Eine Verbesserung könnte z.B. sein, dass anhand der visualisierten Modelle eine neue Regel abgeleitet werden kann. Damit soll die Arbeitsweise in der Bewertung des Systems für den Analysten vereinfacht werden. Zu dem soll gezeigt werden, wie sich die vorgeschlagenen Algorithmen ergänzen.

### 1.2 Motivation und Möglichkeiten

Das Interesse an guten Empfehlungen kommt einerseits aus eigener Erfahrung mit solchen Systemen im Alltag, andererseits auch über die weiterführenden Möglichkeiten die sich ergeben. Ein häufiges Szenario, was ich antreffe sind z.B veraltete oder falsche Vorschläge. Kauft man ein Buch über Gartenpflege als Geschenk für jemanden führt das in den meisten Fällen dazu, dass weitere Vorschläge für Bücher dieser Art angesteuert werden. Die Annahme im Kontext, dass man ausschließlich Bücher über Gartenpflege kaufen will ist in dem Fall falsch. Es ist wichtig den Kunden nicht in einer Kategorie in Form eines *Filter Bubble* (Nagulendra u. Vassileva, 2010) einzuschließen. Eine weitere Quelle der Motivation besteht in der großen Nachfrage der Wirtschaft den Kunden besser zu verstehen, um sich an seine Bedürfnisse anzupassen. Dadurch könnten neue Strategien, wie man mehr Produkte effektiver an den Endkunden bringen kann, entdeckt werden. Vom akademischen Standpunkt her ist es genau so spannend, wenn man Empfehlungen verbessert, um z.B. größere Mengen an Daten zu erforschen. Ein mögliches Beispielszenario könnte ein System sein, welches eigenständig Vorschläge über interessante Orte in den Daten liefern kann. Für sich allein gesehen sind die derzeitigen Systeme für so was noch nicht

ausgelegt, weil sie nicht ohne vorher gesammelte Bewertungen oder Metadaten funktionieren können. Durch die Definition und das Anlernen von bestimmten Merkmalen in Form von Features in Verbindung mit weiteren bekannten Metainformationen, soll sich das System besser an neue Gegebenheiten anpassen. Zusammen mit der Unterstützung, die dem Analysten dadurch gegeben wird, sollte man in der Lage sein schneller zu entscheiden, ob eine zuvor gemachte Veränderung für sinnvoll zu erachten ist.

## 2 Empfehlungssysteme

### 2.1 Verfahren und Literatur

Die nachfolgenden Abschnitte beschäftigen sich mit den derzeitigen Techniken um das Thema der Erstellung von Empfehlungen. Darüber hinaus werden die Schwierigkeiten und Barrieren zu diesen Herangehensweisen beschrieben. Zum großen Teil stammen die hier aufgeführten Grundlagen aus Ricci u. a. (2011) und Jannach u. a. (2010).

#### 2.1.1 Collaborative Filtering

Für die Berechnung von Empfehlungen kann man zwei Arten von Vorschlägen identifizieren, die personalisierten, welche die Daten von ähnlichen Benutzern verwenden und die nicht-personalisierten. Bei den letzteren, wird grob gesagt, ein Durchschnitt über alle bekannten Nutzer errechnet.

*Collaborative Filtering* zählt zu den traditionellen Verfahren, die mit Hilfe von Benutzerdaten zu Produkten arbeiten. Im Grunde gibt es zwei Klassen von Verfahren. Zum einen die den Fokus auf Relationen zwischen Artikeln oder Benutzern auf Grund von Nachbarbeziehungen legen und zum anderen die Methoden, wo die genannten Beziehungen in Form von Faktoren in einer N-Dimensionalen Matrix abgebildet werden. Bei diesem Ansatz werden oft statistische Modelle verwendet. (Adomavicius u. Tuzhilin, 2005) Dabei kann die x-Achse den Kunden darstellen, die y-Achse die verschiedenen Artikel und die z-Achse die Zeit. In der Arbeit von Rendle u. a. (2011) wird gezeigt wie man durch den Kontext der Kunden die *Matrix Factorization* zu besseren Ergebnissen führen kann. Ein wichtiger Punkt besteht darin, dass man die beiden Klassen miteinander kombinieren kann (Koren, 2008). Dies ist ein Faktor der in der geplanten Masterarbeit entscheidend ist. Zu dem kann man mit *Matrix Factorization* verborgene *Features* in den Beziehungen entdecken.

**Benutzer-basierte** Bei diesen Verfahren werden Empfehlungen durch die abgegebenen Bewertungen der aktiven Kunden ermittelt. Der Ablauf sieht im einzelnen wie folgt aus:

1. Ein Datensatz von Bewertungen muss vorhanden sein.

2. Die Nachbarschaftsbeziehungen über die Ähnlichkeit zwischen den Benutzern müssen berechnen.
3. Die gut bewerteten Produkte von den identifizierten ähnlichen Benutzern aus Punkt 2 müssen gefunden werden.
4. Es werden Empfehlung erstellen über die Top N am besten bewerteten Produkte aus Punkt 3 und an die Kunden von Punkt 1 angesteuert.

Damit das ganze gut funktioniert braucht man einen aktiven Kundenstamm im Verhältnis zur Anzahl an Artikeln im System. Zu dem besteht die Schwierigkeit in der Berechnung von Beziehungen bei einer wachsenden Nutzer- und Artikelanzahl. In einem realen System bewertet der Durchschnitts-Kunde nicht sehr viele Artikel und aus diesem Grund ist die wahre Komplexität etwas kleiner, jedoch trotzdem ein bestehendes Problem. (Linden u. a., 2003)

**Produkt-basiert** Komplementär zum Benutzer-basierten Ansatz steht die Untersuchung von Relationen unter Produkten. Bei dieser Variante ermittelt man die Ähnlichkeit zwischen bewerteten Produkten vom Kunden, um ähnliche vorzuschlagen. Bei dieser und der vorherigen Variante kann man durch relativ wenig Arten von Daten gute Ergebnisse liefern. In beiden Fällen machen die Kunden meistens explizite Angaben, was ihnen gefällt oder nicht. Alternativ kann jedoch auch das Benutzerverhalten studiert werden, um daraus neue Erkenntnisse abzuleiten. (siehe Unterabschnitt 2.1.5)

**Clustering & Klassifizierung** Typischerweise werden über das *Clustering* die Produkte oder Benutzer in bestimmte Gruppen eingeteilt. Dabei besitzen die Dinge in einer gemeinsamen Gruppe eine große Ähnlichkeit zueinander. Es wird oft nach einer Hashing-Funktion <sup>1</sup> gesucht, die das Objekt in solch eine Gruppe einordnen kann. Die Empfehlungen aus solch einem Cluster sind dann nicht einfach auf andere übertragbar. Diese Methode eignet sich gut zur Steigerung der Performance für den Ansatz der Nachbarschaftsbeziehungen. Eine weitere Art der Einordnung von Dingen ist die *Klassifizierung*, die ein Benutzermodell anhand von Trainingsdaten berechnet. Ein Vorteil dieser besteht in der Vielseitigkeit und Flexibilität in Kombination mit anderen Verfahren. Jedoch muss man einen guten Satz an Trainingsdaten besitzen, um gute Ergebnisse zu erzielen. Des Weiteren kann es passieren, dass man einen zu spezialisierten Klassifizierer anlernt, welcher nicht mit neuen Artikeldaten funktioniert oder nur populäre Artikel vorschlägt und die neuen außen vorlässt. Das *Kalt-Start* Problem ist ein weiteres Hindernis mit dem die *Collaborative Filtering* Verfahren ihre Schwierigkeiten haben. Dies tritt bei einem neuen System ohne viele Benutzer oder Bewertungen, die man zur Analyse nutzen kann, auf. Eine Teillösung zu diesem Problem wird in Liu u. a. (2011) durch eine angepasste Suche von Faktoren im Benutzer-

---

<sup>1</sup>k-means (siehe AW2), Locality-sensitive hashing, LDA etc.

modell umgesetzt. Die üblichen Implementierungen in der Klassifizierung basieren auf *Support Vector Machines* oder *Bayesian Networks*, die bewährte Verfahren aus der Statistik darstellen.

### 2.1.2 Content-based

Die zuvor aufgezählten Problematiken in der Vielfalt oder Genauigkeit des Empfehlungssystems sind bei den *Content-based* Verfahren kein Hindernis. (Pazzani u. Billsus, 2007) Der Kerngedanke besteht darin die Artikel in Form von *Feature-Vektoren* zu beschreiben. Der Nutzer wird dann durch Klassifizierung für bestimmte *Feature* Merkmale eingeordnet. Dazu werden jedoch gut gepflegte Datenbestände zu den Artikeln benötigt, was nahelegt dass sich textlastige Beschreibungen dafür am besten eignen. Der Vorteil in diesem Verfahren ist das einfachere Verstehen der Empfehlungen, weil man aufgrund der ausgesuchten Features einen Bezug zum Benutzer herstellen kann. Das Festlegen von geeigneten Attributen zur Beschreibung eines Artikels wird entweder manuell oder automatisch ausgeführt. In Ronen u. a. (2013) wird zu diesem Zweck ein Algorithmus vorgestellt, welcher auf einer faktorisierten Matrix arbeitet. Dieser berechnet jeweils eine Punktzahl anhand der vorhandenen Historie für alle bekannten Features, um so die relevanten Attribute für neu hinzugefügte zu definieren. Eine gute Auswahl an Attributen ist für die Genauigkeit und den vielseitigen Einsatz wichtig. (Chakrabarti, 2003) (Manning u. a., 2008) Ein großer Kritikpunkt in diesen einzelnen Verfahren ist die relativ *flache* Analyse über Begriffe, die auch zur Beschreibung eines qualitativ minderen oder zu einem sehr ähnlichen Produkt führen. Zu dem ist das *Kalt-Start* Problem von der Abhängigkeit zu Bewertungshistorie von Nutzern auf die des aktuellen Benutzers verschoben. Anfänglich muss also der aktuell untersuchte Kunde eine Anzahl an Artikeln bewerten.

### 2.1.3 Knowledge-based

Im Gegensatz zu *Collaborative Filtering* und *Content-based* Empfehlungen, die Schwierigkeiten haben eine Veränderung von bestimmten Verhalten oder Vorlieben zu identifizieren, ergänzen die *Knowledge-based* Algorithmen das System um eine interaktive und leitende Rolle im Zusammenspiel mit dem Nutzer. Bei speziellen Konstellationen von Anforderungen braucht man mehr als nur Nachbarschaftsbeziehungen oder Abhängigkeiten zwischen Artikelattributen. Eine Möglichkeit besteht in der Erfüllung von einem *Constraint* Problemen. Bei diesen wird versucht einen Artikel zu finden, der die größte Anzahl an Anforderungen gleichzeitig erfüllt. In der Arbeit von Zanker u. a. (2010) wird dazu ein empirischer Beweis erbracht, dass man durch gewichtete Anforderungen (*Soft-Constraints*) die besten Ergebnisse bekommt, weil man in den meisten Fällen nie alle Anforderungen gleich gut erfüllen kann. Deswegen werden bestimmte Anforderungen vom System verworfen, um jedoch die meisten zu erfüllen. Das Gute an dieser Art von Algorithmen ist die Arbeitsweise, welche ohne Vorwissen ein Resultat liefern kann.

### 2.1.4 Hybrid

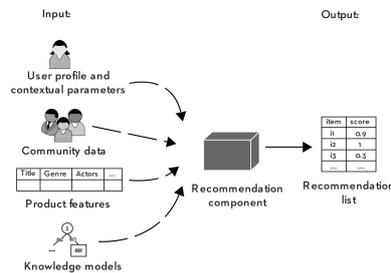


Abbildung 2.1: Aufbau eines Hybrid-Systems. (Jannach u. a., 2010)

In Abbildung 2.1 sieht man die Blackbox eines Empfehlung-Systems, welche als Ergebnis eine Liste von Vorschlägen ausgibt. Die Daten bei der Eingabe sind dabei jeweils alle bisher genannten Verfahren und deren Ergebnisse aus Kapitel 2. Dieser Zustand soll verändert werden und mit der hier vorgestellten Arbeit zu einem klareren und verständlicheren System führen. Da die einzelnen Techniken ihre Vor- und Nachteile besitzen, wurden viele Kombinationen und Verfeinerungen (Burke, 2002) entwickelt, jedoch mit noch sehr wenig Fokus auf die Theorie dahinter. Hybrid-Systeme benutzen die Information der ähnlichen Benutzer, das Finden von vergleichbaren Artikeln zu den eigenen bewerteten und alle weiteren zusätzlichen Informationen in Form von Regeln oder dem Kontext, um Vorschläge zu generieren. Nach Burke (2002) kann man das Design eines solchen Systems in sieben Kategorien einordnen:

**Weighted** Empfehlungen und Relevanz werden separat erstellt und am Ende zu einem gewichteten Durchschnitt zusammengerechnet.

**Switching** Das System entscheidet welcher Algorithmus für die Empfehlung genutzt wird.

**Mixed** Alle Ergebnisse der Algorithmen werden vom System aufgezeigt.

**Cascade** Die gelieferten Empfehlungen können von Ebene zu Ebene neu gefiltert werden.

**Meta-Level** Es wird ein einheitliches Modell als Grundlage genommen.

**Feature Combination** Algorithmen in einen zusammenführen.

**Feature Augmentation** Pipeline von Empfehlungs-Algorithmen aufbauen.

Für den Zweck der Masterarbeit fallen die letzten beiden Strategien raus, weil sie eine monolithische Architektur des Systems begünstigen und damit die Flexibilität des Systems einschränken.

### 2.1.5 Kontext

Wie schon zu Anfang erklärt wird sich ein Teil des Empfehlungs-Systems mit dem Benutzerkontext beschäftigen. Die Nutzung von mehr als nur Bewertungen ist ein vielseitiger Zweig, in dem Potential für die Suche und Evaluierung von relevanten Kontextinformationen steckt, da man mit diesen Methoden bessere Empfehlungen generieren kann. (Panniello u. a., 2014) In der Arbeit von Adomavicius u. Tuzhilin (2011) sind jeweils drei Gruppen von Methoden mit denen man den Kontext berücksichtigen kann, aufgelistet:

**Contextual pre-filtering** Die kontextuellen Informationen werden dazu genutzt vor der Berechnung der Empfehlungen irrelevante Bewertungen herauszufiltern.

**Contextual post-filtering** Das Resultat der Empfehlung wird mit Hilfe von kontextuellen Informationen nach-gefiltert.

**Contextual modeling** Der Kontext wird als eine weitere Größe ins Modell mit den Bewertungen und Artikeln einberechnet.

**Effektivität und Performance** In Panniello u. a. (2014) wurden jeweils diese drei Varianten in ihrer Genauigkeit und Vielseitigkeit untersucht, dabei wurde kein klarer Sieger festgestellt, jedoch wurden drei Faktoren in Zusammenhang mit der Performance ermittelt. Es sind die Aufgabenart<sup>2</sup>, die Kontextgranularität und der Empfehlungsdatentyp dabei am entscheidendsten. Unter dem Begriff des Kontextes werden jeweils zwischen den Informationen über das System, der Benutzer und der Artikel unterschieden. Zuvor wurden in Unterabschnitt 2.1.2 das Problem einer *flachen* Analyse durch die Produkt-basierten Techniken beschrieben. Der Kontext kann als eine Komponente dazu genutzt werden, um dieses Manko in den Empfehlungen zu entschärfen.

**Architektur** Für die Masterarbeit wird der Fokus auf den Vergleich zwischen den Kontextmethoden und nicht auf das Auffinden von relevanten Daten gelegt. Zum letzteren bietet die Arbeit von Said (2010) einen guten Einstieg und zeigt zusätzlich wie man den Kontext in der Architektur des Systems berücksichtigt. Dieses Wissen und das von Hussein u. a. (2014) wird wiederum in der Masterarbeit zur Einbettung des Kontextes genutzt werden.

**Maus-Tracking** Von Guo u. a. (2012) wird ein Modell vorgestellt, welches die Interaktion des Benutzers auf Suchergebnisseiten besser einfängt und den Erfolg einer Suchanfrage vorhersagen kann. Auf Basis dieser Erkenntnisse wird eine einfachere Auswertung der Benutzerzufriedenheit

---

<sup>2</sup>Die Top N Artikel finden oder alle.

auf Basis von Maus-Trackingdaten ermöglicht. In Lagun u. Agichtein (2011) werden solche Daten im Ergebnis mit Tests unter Laborbedingungen mit einem *Eye-Tracker* gleichgestellt, welcher den Aufwand bei der Evaluierung eines Systems vergrößert. In Abbildung 2.2 sieht man Beispieldaten zur Interaktion eines Benutzers in Zusammenspiel mit dem Blick eines Babies und des Werbetextes dazu. Durch diese Platzierung leitet man den Blick auf den Werbetext.

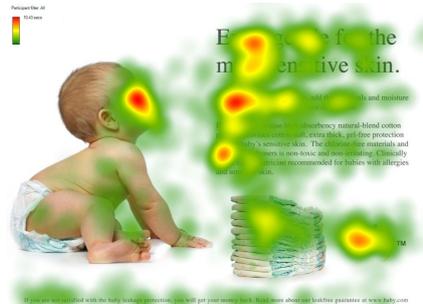


Abbildung 2.2: Kontext über Eye-Tracking der Blickdauer. ©usableworld.com.au (08.11.14)

## 3 Methodik

### 3.1 Umsetzung

Die Einführung in Kapitel 2 zu den Verfahren hat die bestehenden Vor- und Nachteile aufgezeigt. Darüber hinaus wurden Kombinationsmöglichkeiten der einzelnen Verfahren erläutert. Diese werden in der Masterarbeit in dem vorgestellten Umfang dann auch aufgegriffen. Die wissenschaftliche Neuerung, die das System bringen wird, besteht im flexiblen und offenen Zusammenspiel der Algorithmen für das bessere Verstehen. Die Implementierung des Systems soll in Java geschehen, als Basis soll dafür das *Lenskit*<sup>1</sup> Framework verwendet werden, weil ein Teil der genannten *Collaborative Filtering* Verfahren schon vorhanden ist. Das Projekt besitzt eine aktive Community und eine gut gepflegte Dokumentation. Das Framework muss jedoch um die *Content-based* Techniken sowie dem Benutzerkontext in Form von Interaktion<sup>2</sup> mit der Seite und einem Benutzerprofil<sup>3</sup> erweitert werden. Hier besteht das Risiko, dass die bestehende Architektur von *Lenskit* vielleicht einen Umbau benötigt. Ein weiteres nicht zu verschweigenes Risiko stellt die Aufgabengröße mit der Evaluierung aller Verfahren dar. Auch wenn die Machbarkeit in der Kombinationen gezeigt wurde, kann man trotzdem unbekannte Probleme in der speziellen Konstellation nicht ausschließen. Die eigentliche Testumgebung wird auf einem Test- oder Realsystem laufen, je nachdem was organisatorisch machbar ist. In den meisten Fällen

<sup>1</sup><http://lenskit.org/>

<sup>2</sup>Maus-Trackingdaten

<sup>3</sup>Alter, Geburtstag, etc

werden Empfehlungssysteme mit offline Trainingsdaten evaluiert, weil die meisten Firmen ihre Daten nicht preisgeben. Dies ist bisher das größte Risiko, was die Umsetzung sehr erschweren könnte. Im Notfall könnte man, wie die meisten Arbeiten, auf frei verfügbare Filmbewertungen<sup>4</sup> zurückgreifen, was jedoch die Auswertung des Kontextes erschweren wird.

**Feature-Vektor** Für die Bewertung von Produkten wird die Arbeit von Han u. Karypis (2005) implementiert und durch den Benutzerkontext ergänzt. Die Verfeinerung dieses Algorithmus mit dem Kontext, kann in den drei gezeigten Formen dann getestet werden. (siehe Unterabschnitt 2.1.5)

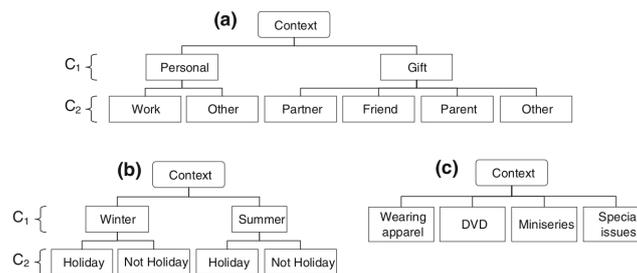


Abbildung 3.1: Kontexthierarchie

**Kontext (Knowledge-based)** Bei den Kontextdaten sollen keine weiteren gesucht, sondern die aufgezählten untereinander in ihren Ergebnissen verglichen werden. Genau so wie die anderen Verfahren werden diese dann als flexible Bausteine im fertigen System in der Komposition evaluiert. Zu den genannten Maus-Trackingdaten sollen die in Abbildung 3.1 zu sehende Profilhierarchien hinzugefügt werden. (Panniello u. a., 2014) (Panniello u. Gorgoglione, 2012) Die neuen Erkenntnisse daraus können dann für weitere Arbeiten genutzt werden.

**Visualisierung** Die Visualisierung wird in Form von Mengendarstellungen umgesetzt, die dann auf unterschiedlichen Ebenen angewendet werden. Das bedeutet, dass man in die Daten reinzoomen kann. So soll der Zusammenhang einer Empfehlungsmenge zu den Algorithmen hergestellt werden. Angefangen wird eine Darstellung des ganzen Systems, welche man in Abbildung 3.2 (links) sieht. Die unterste Stufe beim Zoom ist dann ein einzelnes Benutzermodell und die für diesen speziellen Kunden empfohlenen Artikel. Diese Art der Analyse, die durch diese Arbeitsweise geboten wird, ist mit Hybriden-Empfehlungssystemen ein von mir vorgeschlagener Lösungsweg. Für die Implementierung der Mengen werden die Visualisierungen von Parra u. a. (2014) und Verbert u. a. (2013) eingesetzt. Je nachdem wie gut die Analyse mit dem Werkzeug klappt, kann es sein, dass sich diese Visualisierungsstrategie nur für Teile des Systems eignet.

<sup>4</sup><http://grouplens.org/datasets/movielens/>

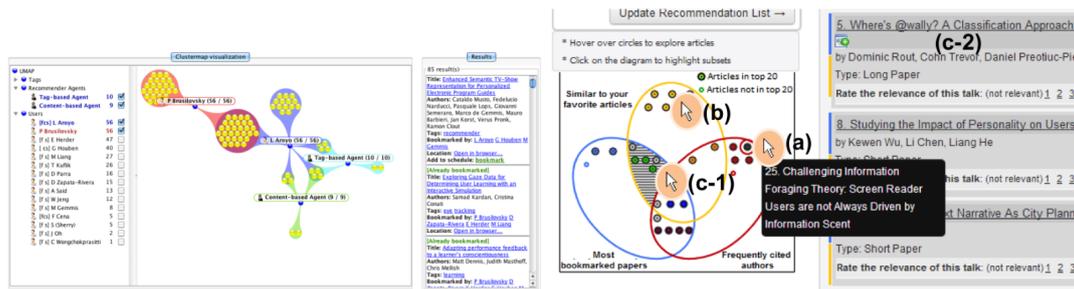


Abbildung 3.2: Relationen zwischen Empfehlungen und den Algorithmen.

### 3.2 Evaluierung

Für die Evaluierung gibt es mehrere Ansätze die verfolgt werden können. Einmal die Nutzung eines realen Systems mit einer *A/B Test*<sup>5</sup> Methode, bei der man einen kleinen Teil der Nutzer die Empfehlungen des gebauten Systems sehen lässt. Hier besteht das Risiko, dass sich kein Online-Händler bereit erklärt solche Tests auszuführen. Als Alternative kann man sich Domänen-Experten oder Analysten suchen, die auf Trainingsdaten mit dem Werkzeug zusammen arbeiten. Jannach u. a. (2010) Am Ende muss dann evaluiert werden inwieweit, dass Werkzeug bei der Erfüllung der Aufgaben geholfen hat oder die Arbeitsweise erleichtert hat. Ein wünschenswertes Ergebnis wäre ein Erkenntnisgewinn für den Analysten über die Arbeitsweise der Verfahren. Als letzte Variante kann man einen Testshop mit künstlichen Artikeln und Probanden aufsetzen, die man auf eine virtuelle Einkaufstour schickt. Die Auswertung geschieht dann in der Form eines Fragebogens und der Benutzerzufriedenheit zu den Empfehlungen oder der Arbeitsweise. Als eine Alternative zu den realen Testprobanden kann man auch durch die Trainingsdaten den Erfolg mit statistischen Methoden<sup>6</sup> analysieren, jedoch klappt das nicht für alle Empfehlungsarten, weil z.B. die auf dem Kontext aufbauenden zu speziell auf einen Benutzer geschnitten sind und keine statistische Relevanz für die globalen Daten haben.

<sup>5</sup>[http://en.wikipedia.org/wiki/A/B\\_testing](http://en.wikipedia.org/wiki/A/B_testing)

<sup>6</sup>[http://en.wikipedia.org/wiki/Root-mean-square\\_deviation](http://en.wikipedia.org/wiki/Root-mean-square_deviation) oder [http://en.wikipedia.org/wiki/Mean\\_absolute\\_error](http://en.wikipedia.org/wiki/Mean_absolute_error)

## Literaturverzeichnis

- [Adomavicius u. Tuzhilin 2005] Adomavicius, Gediminas ; Tuzhilin, Alexander: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. In: *Knowledge and Data Engineering, IEEE Transactions on* 17 (2005), Nr. 6, S. 734–749 4
- [Adomavicius u. Tuzhilin 2011] Adomavicius, Gediminas ; Tuzhilin, Alexander: Context-aware recommender systems. In: *Recommender systems handbook*. Springer, 2011, S. 217–253 1, 8
- [Burke 2002] Burke, Robin: Hybrid recommender systems: Survey and experiments. In: *User modeling and user-adapted interaction* 12 (2002), Nr. 4, S. 331–370 2, 7
- [Chakrabarti 2003] Chakrabarti, Soumen: *Mining the Web: Discovering knowledge from hyper-text data*. Morgan Kaufmann, 2003 6
- [Guo u. a. 2012] Guo, Qi ; Lagun, Dmitry ; Agichtein, Eugene: Predicting Web Search Success with Fine-grained Interaction Data. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York, NY, USA : ACM, 2012 (CIKM '12). – ISBN 978-1-4503-1156-4, 2050–2054 8
- [Han u. Karypis 2005] Han, Eui-Hong S. ; Karypis, George: Feature-based recommendation system. In: *Proceedings of the 14th ACM international conference on Information and knowledge management* ACM, 2005, S. 446–452 10
- [Hussein u. a. 2014] Hussein, Tim ; Linder, Timm ; Gaulke, Werner ; Ziegler, Jürgen: Hybreed: A software framework for developing context-aware hybrid recommender systems. In: *User Modeling and User-Adapted Interaction* 24 (2014), Nr. 1-2, S. 121–174 8
- [Jannach u. a. 2010] Jannach, Dietmar ; Zanker, Markus ; Felfernig, Alexander ; Friedrich, Gerhard: *Recommender systems: an introduction*. Cambridge University Press, 2010 3, 4, 7, 11
- [Koren 2008] Koren, Yehuda: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* ACM, 2008, S. 426–434 4
- [Lagun u. Agichtein 2011] Lagun, Dmitry ; Agichtein, Eugene: ViewSer: Enabling Large-scale Remote User Studies of Web Search Examination and Interaction. In: *Proceedings of the 34th*

- International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA : ACM, 2011 (SIGIR '11). – ISBN 978-1-4503-0757-4, 365–374 9
- [Linden u. a. 2003] Linden, Greg ; Smith, Brent ; York, Jeremy: Amazon. com recommendations: Item-to-item collaborative filtering. In: *Internet Computing, IEEE* 7 (2003), Nr. 1, S. 76–80 5
- [Liu u. a. 2011] Liu, Nathan N. ; Meng, Xiangrui ; Liu, Chao ; Yang, Qiang: Wisdom of the better few: cold start recommendation via representative based rating elicitation. In: *Proceedings of the fifth ACM conference on Recommender systems* ACM, 2011, S. 37–44 5
- [Manning u. a. 2008] Manning, Christopher D. ; Raghavan, Prabhakar ; Schütze, Hinrich: *Introduction to information retrieval*. Bd. 1. Cambridge university press Cambridge, 2008 6
- [Nagulendra u. Vassileva 2010] Nagulendra, Sayooran ; Vassileva, Julita: Understanding and controlling the filter bubble through interactive visualization: a user study. (2010) 3
- [Panniello u. Gorgoglione 2012] Panniello, Umberto ; Gorgoglione, Michele: Incorporating context into recommender systems: an empirical comparison of context-based approaches. In: *Electronic Commerce Research* 12 (2012), Nr. 1, S. 1–30 10
- [Panniello u. a. 2014] Panniello, Umberto ; Tuzhilin, Alexander ; Gorgoglione, Michele: Comparing context-aware recommender systems in terms of accuracy and diversity. In: *User Modeling and User-Adapted Interaction* 24 (2014), Nr. 1-2, S. 35–65 8, 10
- [Parra u. a. 2014] Parra, Denis ; Brusilovsky, Peter ; Trattner, Christoph: See what you want to see: Visual user-driven approach for hybrid recommendation. In: *Proceedings of the 19th international conference on Intelligent User Interfaces* ACM, 2014, S. 235–240 10
- [Pazzani u. Billsus 2007] Pazzani, Michael J. ; Billsus, Daniel: Content-based recommendation systems. In: *The adaptive web*. Springer, 2007, S. 325–341 2, 6
- [Rendle u. a. 2011] Rendle, Steffen ; Gantner, Zeno ; Freudenthaler, Christoph ; Schmidt-Thieme, Lars: Fast context-aware recommendations with factorization machines. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* ACM, 2011, S. 635–644 4
- [Ricci u. a. 2011] Ricci, Francesco ; Rokach, Lior ; Shapira, Bracha ; Kantor, Paul B.: *Recommender systems handbook*. Bd. 1. Springer, 2011 4
- [Ronen u. a. 2013] Ronen, Royi ; Koenigstein, Noam ; Ziklik, Elad ; Nice, Nir: Selecting content-based features for collaborative filtering recommenders. In: *Proceedings of the 7th ACM conference on Recommender systems* ACM, 2013, S. 407–410 6

- [Said 2010] Said, Alan: Identifying and utilizing contextual data in hybrid recommender systems. In: *Proceedings of the fourth ACM conference on Recommender systems* ACM, 2010, S. 365–368 8
- [Sarwar u. a. 2001] Sarwar, Badrul ; Karypis, George ; Konstan, Joseph ; Riedl, John: Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web* ACM, 2001, S. 285–295 2
- [Verbert u. a. 2013] Verbert, Katrien ; Parra, Denis ; Brusilovsky, Peter ; Duval, Erik: Visualizing recommendations to support exploration, transparency and controllability. In: *Proceedings of the 2013 international conference on Intelligent user interfaces* ACM, 2013, S. 351–362 10
- [Zanker u. a. 2010] Zanker, Markus ; Jessenitschnig, Markus ; Schmid, Wolfgang: Preference reasoning with soft constraints in constraint-based recommender systems. In: *Constraints* 15 (2010), Nr. 4, S. 574–595 6