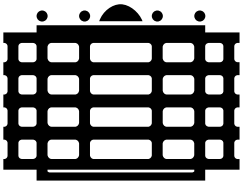
Technische Universität Chemnitz
Fakultät für Informatik
Professur Künstliche Intelligenz



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Diplomarbeit

im Studiengang Angewandte Informatik

Vorgelegt von
Tolleiv Nietsch

—

Skalierbare Item Recommendation in Big-Data und Suchindexen

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende schriftliche Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keinem anderen Prüfer als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Vorname:	Tolleiv
Name:	Nietsch
Matrikelnummer:	172314

Chemnitz, den 29.03.2012

_____

Tolleiv Nietsch

Betreuung und Prüfung durch:

Prof. Dr. Fred Hamker,	Professur Künstliche Intelligenz, TU-Chemnitz
Dr. Johannes Steinmüller	Professur Künstliche Intelligenz, TU-Chemnitz

Technische Universität Chemnitz, Fakultät für Informatik
Straße der Nationen 62, 09107 Chemnitz

Zusammenfassung

tbw

Inhaltsverzeichnis

1 Einleitung

Hinweis dass Ëlementeünd Ätemöft als Synonym verwendet werden, ähnlich wie auch "Ratingünd "Bewertung"gleichzusetzen sind.

2 Grundlagen

Auf welchen Themen und Techniken baut die Arbeit auf.

2.1 Recommendation Konzepte

Die Auswahl von möglichst relevanten Empfehlungen für einen Nutzer kann auf sehr verschiedenen Wegen getroffen werden. Für die zahlreichen bekannten Techniken wird in der Literatur vorwiegend die folgende Gliederung genutzt [Ricci u. a., 2010, Kap. 1] [Burke, 2002] [Jannach u. a., 2010]:

- *Kollaboratives Filtern*, auch *Collaborative filtering (CF)*, gewinnt relevante Elemente aus dem Vergleich des Nutzerprofils mit anderen Profilen.
- *Community-basierte Empfehlungen*, bzw. *Community-based filtering*, nutzen die Ähnlichkeit innerhalb von Gruppen, etwa in sozialen Netzwerken, um relevante Elemente zu finden.
- *Demographisch gestützte Empfehlungen* leiten sich von den Stereotypen, denen ein Nutzer zugeordnet wird, ab.
- *Inhaltsbasierte Empfehlungen* oder *Content-based recommendations*, werden auf der Basis von, am Nutzerprofil gewichteten, Element-Eigenschaften getroffen.
- *Wissensbasierte Empfehlungen* bzw. *Knowledge-based recommendations* werden durch zusätzliches domänenspezifisches Wissen generiert.
- *Utility-basierte Empfehlungen* bestimmen sich durch die Berechnung der Nützlichkeit der Elemente für den Nutzer mit Hilfe der sog. *Utility Function*.
- *Hybride Systeme* kombinieren verschiedene Techniken um die Schwächen der einzelnen auszugleichen.

Die diesen Gruppen zugrunde liegenden Methoden werden in den nächsten Abschnitten näher erläutert. Dazu werden jeweils die zu erhebenden Daten, deren Verarbeitung und die Vor- und Nachteile der Methode beschrieben.

2.1.1 Kollaboratives Filtern

Der Grundgedanke beim kollaborativen Filtern ist, dass Nutzer die in der Vergangenheit gleiche Interessen hatten, diese auch in der Zukunft durch ähnliches Verhalten ausdrücken. So können Empfehlungen für einen Nutzer aus dem Verhalten ähnlicher Nutzer abgeleitet werden. Die Nutzerprofile bilden sich dabei ausschließlich aus Elementbewertungen (*Ratings*), Eigenschaften der bewerteten Elemente fließen nicht ein. Die Ähnlichkeit der Nutzer drückt sich entsprechend durch Gemeinsamkeiten in den Bewertungen aus. [Jannach u. a., 2010, Kap. 2]

Aus den Profilen aller Nutzer ergibt sich eine sog. *User-Item* Matrix, diese ermöglicht es ähnliche Nutzer oder auch ähnliche Elemente im System zu finden. Zur Auswertung dieser Matrix, bzw. zur Generierung von Empfehlungen mit Hilfe dieser Matrix existieren verschiedene Strategien welche in Abschnitt 2.2 näher beschrieben werden.

Die Erhebung der Ratings kann sowohl auf explizite Weise, etwa mit einer 5-Punkte-Likert-Skala, oder implizit, zum Beispiel durch die Aufzeichnung von Browsing-Verläufen, geschehen.

Ein wichtiger Vorteil des kollaborativen Filterns liegt darin, dass Empfehlungen unabhängig von Elementeigenschaften gebildet werden können. Dadurch können auch Elemente deren Inhalt nur schwer oder gar nicht gewonnen werden kann in die Empfehlung einbezogen werden. Die zahlreichen Forschungsarbeiten und die große Zahl der daraus hervorgegangenen Filterstrategien ist ebenfalls ein Vorteil.

Problematisch ist die Verwendung bei Systemen in denen der Nutzer (noch) kein oder nur ein sehr begrenztes Profil hat (*cold start*). Zudem ist es nicht in jedem Fall sinnvoll alle Eigenschaften der Elemente ausser Acht zu lassen, da so ggf. problemspezifische Entscheidungskriterien unbeachtet bleiben. [Ricci u. a., 2010; Burke, 2002]

2.1.2 Community-basierte Empfehlungen

Gemäß [Sinha u. Swearingen, 2001] haben Nutzer ein größeres Vertrauen in Empfehlungen wenn sie von Freunden ausgesprochen werden. Diesem Ansatz folgend werden in community-basierten Systemen Empfehlungen entsprechend der Präferenzen der Freunde eines Nutzers ausgesprochen. Das Nutzerprofil bildet sich daher aus einer Liste von Elementbewertungen und einer Liste von sozialen Verbindungen zu anderen Nutzern.

Da in Vergleichen mit reinen kollaborativen Systemen keine eindeutige Verbesserung der Empfehlungen nachgewiesen werden konnte, stellt das größere Vertrauen in die gebotenen Empfehlungen den wesentlichen Vorteil dieser Methode dar. Die gute Verbreitung und Verfügbarkeit der Daten über öffentliche Schnittstellen von bestehenden sozialen Netzwerken, wie etwa Facebook oder LinkedIn, sind ebenfalls positiv. Die Abwägung zwischen der Aufrechterhaltung der Privatsphäre und dem dadurch resultierenden Verlust an Genauigkeit ist ein wichtiges Problem (vgl. [Machanavajjhala u. a., 2011]). Auch das fehlende theoretische Fundament in anderen Bereichen, etwa beim Aufbau von Vertrauen und Misstrauen zwischen Nutzern, birgt mögliche Probleme bei der Umsetzung. [Victor u. a., 2011]

2.1.3 Demographisch gestützte Empfehlungen

Eine weitere Methode um ähnliche Nutzer zu finden ist die Gruppierung nach demographischen Eigenschaften. So können Gruppen zum Beispiel entsprechend des Alters, der Sprache oder des Geschlechts gebildet werden. Sie können allerdings auch mit Hilfe der Methoden des maschinellen Lernens aus bestehenden Transaktionsdaten gewonnen werden (vgl. [Burke, 2002]). Wie bei den vorangegangenen Methoden bildet sich auch hier das Nutzerprofil zunächst aus einer Liste von Elementbewertungen, ergänzt wird es durch die entsprechenden demographischen Eigenschaften. Die Empfehlungen für den einzelnen Nutzer ergeben sich aus seinen eigenen Präferenzen die endsprechend der Gruppenzugehörigkeit gewichtet werden.

Arbeiten zu reinen demographischen Systemen gibt es kaum. In vielen Fällen, wie etwa [Vozalis u. Margaritis, 2007] werden kollaborative Ansätze ergänzt um eine Verbesserung

der Empfehlungsergebnisse zu erzielen bzw. um die Probleme bei Empfehlungen für neue Nutzer zu verringern. [Burke, 2002]

2.1.4 Inhaltsbasierte Empfehlungen

Bei der inhaltsbasierten Generierung von Empfehlungen werden die Element-Ratings eines Nutzers zur Erzeugung eines "Interessenprofils" genutzt. In diesem Profil drücken sich die Präferenzen des Nutzers für die inhaltlichen Eigenschaften der Elemente aus und so kann es direkt genutzt werden um ihm Elemente mit ähnlichen Eigenschaften zu empfehlen. Hat ein Nutzer also zum Beispiel ein 'Harry Potter" Buch positiv bewertet, so kann man leicht schlussfolgern dass auch andere Fantasy-Bücher empfohlen werden könnten.

Neben der automatischen Erstellung des Profils ist es auch möglich dieses explizit vom Nutzer zu erfragen. Abhängig vom Problemfeld kann dies schneller zu guten Empfehlungen führen und zur Steigerung des Vertrauens in die erzeugten Empfehlungen beitragen, vgl. [Victor u. a., 2011].

Zur Bestimmung ähnlicher Dokumente, bzw. zur Extraktion der relevanten Eigenschaften (*Features*) werden abhängig vom Elementtyp verschiedene Methoden genutzt. Diese reichen von Entscheidungsbäumen über neuronale Netze bis hin zu Vektorraum-Verfahren (vgl. Abschnitt 2.2 und [Jannach u. a., 2010, Kap. 3]). Die große Anzahl der dafür zur Verfügung stehenden Verfahren, die damit verbundenen Erfahrungen und das daraus abgeleitete Problembewusstsein ist einer der Vorteile. Wichtiger noch ist die Tatsache dass inhaltsbasierte Empfehlungen unabhängig von der Größe des Systems bzw. von das Anzahl der Nutzer generiert werden können. Ein weiterer Vorteil ist es dass für die so gewonnenen Empfehlungen auch leichter Erklärungen für den Nutzer generiert werden können, was wiederum ein wichtiger Faktor zur Steigerung des Vertrauens in die Qualität ist.

Schwierigkeiten bei der Erzeugung von Empfehlungen ergeben sich wenn die für den Nutzer relevanten Eigenschaften nicht direkt "messbar" vorliegen. Zum Beispiel des Ästhetik eines Produktes oder die Nutzbarkeit einer Webseite lassen sich nur sehr schwer erfassen, können aber beim Vergleich zweier Elemente wichtiger sein als textuelle Eigenschaften.

Wie auch beim kollaborativen Filtern ist es bei dieser Methode sehr schwer gute Empfehlungen für Nutzer zu generieren, wenn diese kein oder nur ein unvollständiges Profil haben. Eine weitere Schwierigkeit ergibt sich daraus dass Empfehlungen nur aus dem "bevorzugtenÏnteressenbereich des Nutzers gewonnen werden, dies kann zu sehr ähnlichen und kaum "überraschenden" Empfehlungen führen und zu einem Problem was als *more of the same* umschrieben wird. [Jannach u. a., 2010, Kap. 3] [Lops u. a., 2011]

2.1.5 Wissensbasierte Empfehlungen

Wenn die Frequenz mit der Nutzer ein Element brauchen oder konsumieren sehr gering ist, wie es etwa bei Hauskäufen der Fall ist, ergibt sich für die bisher beschriebenen Methoden das Problem dass nur selten umfangreiche Nutzerprofile zur Verfügung stehen oder die darin enthaltenen Informationen schlicht veraltet sind. Oft gibt es zudem in vielen Bereichen Expertenwissen bzw. domänenspezifisches Wissen welches zur Verbesserung von Empfehlungen bzw. zur Einschränkung der Kandidatenliste genutzt werden kann.

Um dieses vorhandene Wissen zur Generierung von Empfehlungen nutzbar zu machen, kann man es in eine Menge von Regeln überführen und mögliche Empfehlungen entsprechend der Regeln filtern. So kann man zum Beispiel aus der Information dass der Nutzer auf der Suche nach einer Wohnung für seine fünfköpfige Familie ist, leicht ableiten dass $40m^2$ Wohnungen nicht empfehlenswert sind und dass solche mit zwei Bädern oder in einer ruhigeren Wohnlage empfohlen werden können.

Form und Inhalt des Nutzerprofils variieren hierbei in Abhängigkeit von der gewählten Wissens- bzw. Regelrepräsentation. Die Einbeziehung von Expertenwissen ermöglicht es auch übliche Standards einzubeziehen und es erleichtert die Vervollständigung des Nutzerprofils durch die Auswahl sinnvoller Fragen bei der Interaktion mit dem Nutzer. Auch in Fällen, in denen keine Vorschläge gefunden werden konnten, haben regelbasierte Systeme Vorteile. Die Information darüber dass keine Empfehlungen gefunden für eine Anfrage gefunden werden konnten, werden Nutzer schneller akzeptieren wenn das System zudem eine Reihe von Vorschlägen unterbreiten kann welche der Regeln ausgelassen werden könnten um neue Empfehlungen zu generieren. Nachteile ergeben sich wenn das Expertenwissen

und die darauf basierenden Regeln nicht an neue Entwicklungen angepasst werden oder wenn für den Nutzer wichtige Features umbewertet bleiben. [Jannach u. a., 2010, Kap. 4]

2.1.6 Utility-basierte Empfehlungen

Ein zweiter Ansatz um domänenspezifisches Wissen zum Ausgangspunkt von Empfehlungen zu machen ergibt sich, indem man die "Nützlichkeit" eines Elements mit Hilfe einer nutzerspezifischen Funktion (*Utility function*) berechnet. Dadurch kann zum Beispiel eine mögliche Toleranz des Nutzers gegenüber gewissen Produktmerkmalen direkt ins Verhältnis zur Dringlichkeit einer Bestellung gesetzt werden. Das Nutzerprofil ergibt sich dabei aus den Parametern der Funktion, welche i.d.R. explizit von Nutzer erfragt werden müssen.

Vor- und Nachteile sind ähnlich gelagert wie im vorangegangene Abschnitt. Vor allem der direkt Einfluss, den der Nutzer auf die Qualität der Ergebnisse hat, kann zur Steigerung das Vertrauens in die generierten Empfehlungen führen. [Ricci u. a., 2010, Kap. 1] [Burke, 2002; Victor u. a., 2011]

2.2 Filtermodelle

Will man die in Abschnitt 2.1.1 beschriebenen kollaborativen Filtermethoden nutzen, stellt sich das Problem wie man die Ähnlichkeit von Nutzern oder Elementen bestimmen kann und wie man dann Empfehlungen für einen Nutzer erzeugt. Die dafür nötigen Modelle sollen in den folgenden Abschnitten näher erläutert werden.

Grundlage der im Folgenden beschriebenen Methoden ist eine *User-Item* Matrix R welche die Bewertung aller Nutzer U für die Elemente (Produkte) P enthält. Die Wahl des Wertebereichs hängt dabei von der Applikation ab. Ein Beispiel für eine solche Matrix wird in Tabelle 1 gezeigt.

	Item1	Item2	Item3	Item4	Item5	Item6	Item7
User1	5.0	3.0	2.5				
User2	2.0	2.5	5.0	2.0			
User3	2.5			4.0	4.5		5.0
User4	5.0		3.0	4.5		4.0	
User5	4.0	3.0	2.0	4.0	3.5	4.0	

Tabelle 1: Beispiel-Matrix für User-Item Ratings

2.2.1 Ähnlichkeitsmaße

Euklidische Distanz Die naheliegendste Form zur Bestimmung der Ähnlichkeit zwischen zwei Spalten oder zwei Zeilen der User-Item Matrix ist es deren Abstand im n-dimensionalen euklidischen Raum, gem. Formel (1) zu nutzen.

$$dist(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{1}$$

$$sim(a,b) = \frac{1}{1 + dist(a,b)} \tag{2}$$

Hierbei ist n die Anzahl der Dimensionen und a_i bzw. b_i beziehen sich auf das i^{te} Attribut der Objekte, resp. die Ratings der Nutzer. Um den Distanzwert zu einem Maß der Ähnlichkeit mit einem Wertebereich von 1 (starke Korrelation) bis 0 (keine Korrelation) umzuformen, kann Formel (2) genutzt werden.

Aus der Verallgemeinerung dieser Berechnung, der sog. *Lr-Norm* bzw. dem *Minkowski Abstand*, ergeben sich weitere Abstandsmaße. Die sog. *L1-Norm* (auch *City-Block-* oder *Manhattan-Distanz*) entspricht $r = 1$, $r = 2$ entspricht dem o.g. euklidische Abstand und $r = \infty$ entspricht dem *Tschebyscheff-Abstand*. [Amatriain u. a., 2009]

$$dist(a,b) = \sum_{i=1}^{n}(|a_i - b_i|^r)^{\frac{1}{r}} \tag{3}$$

Anwendung finden die verschiedenen Abstandsmaße zum Beispiel in XXXXXX

Anwendungsb raussu- chen

Pearson-Korrelation Ein Problem bei der Berechnung mit der euklidischen Distanz ist, dass die Mittelwerte und Varianzen der Bewertungen einzelner Nutzer voneinander abweichen können obwohl sich diese vergleichbare "Interessen" haben (vgl. [Segaran, 2007, Kap. 2]). Dieser Mangel wird mit Hilfe der *Pearson-Korrelation* (4) beseitigt. Ihr Wertebereich reicht von 1 (starke Korrelation) bis -1 (starke negative Korrelation). Vor Allem bei der Bestimmung von nutzerbasierten Ähnlichkeiten konnten mit ihr in vielen Fällen sehr gute Ergebnisse erzielt werden. Zudem existieren zahlreiche Erweiterungen, um zum Beispiel die Gewichtung von Übereinstimmungen bei der Bewertung von kontroversen Elementen stärker hervorzuheben. [Jannach u. a., 2010][Kap. 2.1] [Amatriain u. a., 2009]

$$sim(a,b) = \frac{\sum_{p\in P}(r_{a,p} - \bar{r_a})(r_{b,p} - \bar{r_b})}{\sqrt{\sum_{p\in P}(r_{a,p} - \bar{r_a})^2}\sqrt{\sum_{p\in P}(r_{b,p} - \bar{r_b})^2}} \tag{4}$$

Kosinus-Ähnlichkeit Ein weiterer Ansatz, der sich zum Standardmaß bei der Abbildung von Element- bzw. Item-Ähnlichkeit entwickelt hat, ist die *Kosinus-Ähnlichkeit* (5). Die Distanz zwischen zwei Vektoren entspricht dabei dem zwischen ihnen aufgespannten Winkel, entsprechend steigt die Ähnlichkeit von Vektoren wenn diese in die gleiche Richtung zeigen.

$$sim(a,b) = \frac{a \cdot b}{\|a\|\|b\|} \tag{5}$$

Der Wertebereich des erzeugten Ähnlichkeitsmaßes liegt zwischen 1 (starke Korrelation) und 0 (keine Korrelation) wenn die genutzten Ausgangsvektoren nur positive Werte haben. Dies ist zum Beispiel der Fall bei den oft üblichen 5 Stern Rating-Skalen oder beim Vergleich von Textdokumenten anhand der Vorkommen einzelner Wörter. Das Maß reicht bis -1 für starke negative Korrelationen wenn auch negative Werte genutzt werden. [Jannach u. a., 2010][Kap. 2.2]

Jaccard-Koeffizient Liegen Ratings nur als binäre Werte vor, kann die Ähnlichkeit zweier Elemente durch das Verhältnis der Schnittmenge zur Vereinigungsmenge dieser definiert werden. Der Wertebereich des sog. *Jaccard-Koeffizienten* (6) liegt ebenso zwischen

1 und 0. Verwendung findet er auch wenn die Werte wenig Informationen tragen, die Information ob ein Nutzer eine Bewertung abgegeben hat im Zentrum der Betrachtung steht oder durch die Rating-Werte Beziehungen zwischen Nutzern und Elementen (im Sinne eines Graphen) ausgedrückt werden. Erweitert wird der Jaccard-Koeffizent vom *Tanimoto-* und vom *Dice-Koeffizienten* (vgl. [Bogers u. van den Bosch, 2009]). [Jannach u. a., 2010, Kap. 3.1] [Segaran, 2007]

$$sim(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

Welches der Distanzmaße für eine konkrete Anwendung genutzt werden sollte kann nicht pauschal beantwortet werden. Bei der Bestimmung von nutzerbasierten Ähnlichkeiten stellt in vielen Fällen die Peason-Korrelation einen guten Ausgangspunkt dar, beim Vergleich von Elementen ist die Kosinus Ähnlichkeit oft eine gute Wahl, aber in jedem Fall muss die Wahl eines Maßes immer mit einer entsprechenden Evaluation gegenüber anderen Maßen kontrolliert werden (vgl. Abschnitt 2.6 u. 5).

2.2.2 Nachbarschaftsmodelle

	User2	User3	User4	User5
User1	0.203	0.286	0.667	0.472

Tabelle 2: Aus Tabelle 1 mit der euklidischen Distanz abgeleiteter Ähnlichkeitsvektor für User1

Die Information über die Ähnlichkeit zweier Nutzer kann mit Hilfe von Nachbarschaftsmodellen (*Neighborhood Models*) zur Generierung von Empfehlungen genutzt werden. Um Empfehlungen für einen Nutzer aus den in Tabelle 1 gezeigten Ausgangsdaten abzuleiten, wird mit Hilfe der schon vorliegenden Ratings zunächst die Ähnlichkeit von dieses Nutzers zu anderen berechnet (siehe Tabelle 2). Um den möglichen Wert eines Elements für einen Nutzer aus diesen abzuleiten ($pred(u,p)$), werden die Ratings anderer Nutzer für dieses Element entsprechend der Ähnlichkeit zwischen den Nutzern (siehe Formel 7)) aufsummiert und normiert. Hierbei muss zudem ein weiterer Unterschied zwischen einzelnen Nutzern in Betracht gezogen werden. Auch wenn Nutzer generell ähnliche Interessen oder Meinungen haben, so kann es durchaus sein, dass der Mittelwert der Ratings dieser Nutzer sehr verschieden ist. . Diese Gewichtung am Rating-Mittelwert $\overline{r_u}$ der Nutzer wird aus diesem Grund in der erweiterten Formel (8) beachtet. [Jannach u. a., 2010]

Quelle dazu - Töscher 2008

$$pred(u,p) = \frac{\sum_{b \in U} sim(u,b) * r_{b,p}}{\sum_{b \in U} sim(u,b)} \tag{7}$$

$$pred(u,p) = \overline{r_u} + \frac{\sum_{b \in U} sim(u,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in U} sim(u,b)} \tag{8}$$

[?]

2.2.3 SlopeOne

[Jannach u. a., 2010, S 41]

2.2.4 Matrixfaktorisierung

[Koren u. a., 2009]

2.2.5 Graphen Modelle

2.3 Schwierigkeiten von Recommendern

- Allgemeine Recommendation Challenges und Ansätze (Sparse, Grey Sheep, "rich-gets-richer"...) - Latenzzeit für das lernen (Nutzer hat grad was angeschaut) - Lösung wie dus gemacht hast (Real time learning) - Item based ohne Nutzerid - Zeitliche Aspekte (Weihnachtsgeschäft etc) - (hattest du ja ausgeführt)

2.4 Suchindexe

2.4.1 Dokumentenrepräsentation

2.4.2 Relevanzberechnung

2.5 Skalierungsstrategien

[Linden u. a., 2003] Amazon.com recommendations: item-to-item collaborative filtering

- Recommendation Datenmodelle die ßchnellßind - die meisten arbeiten mit Ïn-Memory"Datenmodellen (mit Optimierte Speicherung für die Ausführung) - Wo sind die Grenzen (user,item anzahl ?) - Lösungen: Precomputing similarities? - Lösungen: Cluster-based recommendation - Wann macht distributed recommendation sinn - Wann stoßen memory based ans limit, welche algorithmen gibt es - Ist Offline recommendation dafür ein Use-Case (Calculation recommendations and send via mail)? - Hadoop vs Sharding - Das lernen der Datenmodelle muss auch skalieren - Hadopp etc...

[Vidal, 2005] [Töscher u. a., 2008] [Linden u. a., 2003]

2.5.1 Online- / Offline-Recommendation

2.5.2 MapReduce basierte Algorithmen

[Chu u. a., 2006]

2.6 Qualitätsmaße

- Successful session [Smyth u. a., 2011, 2005]
- Precision /Recall /F1, [Jannach u. a., 2010][Kap. 7]
- User- und Itemcoverage [Jannach u. a., 2010][S. 183]
- Conversion / Click-through rates

2.6.1 Mittlere Abweichung

2.6.2 Trefferquote und Genauigkeit

2.6.3 Empirische Messung

3 Entwurf

3.1 Anforderungen

.

Potentielle Use-Cases:

A) Personalisierte Suche: Wenn man nach allgemeinen Begriffen wie "GeschenkWeinöder "Kleidßucht, bekommt man bereits durch andere Nutzer gelernte passende Empfehlungen in der Suche höher angezeigt. (Nutzer schaut sich Rotweine an und bekommt bei Suche nach Wein passende Rotweine angezeigt. Im Vergleich dazu ein Nutzer der sich Weißweine angeschaut hat)

B) Personalisiertes-Recommendation Widget: - In der rechten Spalte werden (passend zu den letzten besuchten Items oder durchgeführten Suchen) persönliche Empfehlungen gegeben

C) Context-Recommendation Widget: 1) Alla "Nutzer die x gekauft haben haben auch y gekauftöder simple "Find most simelar items to an item"Hier macht die Kombination von Suchindex und Recommendation auch Sinn weil: - Der Suchindex alle zur Ausgabe relevanten Daten eines Dokumentes hat - (Vorberechnung?) - Es einfach möglich ist die Items über die die Recommendation gemacht werden sollen über inhaltliche (regelbasierte) Suchqueries weiter einzuschränken oder zu boosten (e.g. Items die höheren Preis haben und in gleicher Kategorie sind höher boosten)

2) Crosselling Widget im Warenkorb (Empfehlungen zu den Items im Warenkorb)

3.2 Systemarchitektur

.

-Wenn man von großen Mengen (großer Raum) von Items ausgeht stellt sich das Problem, das die Teilmenge der Items die der Recommender und die Suche zu einer Änfrageßurück geben können disjunkt sein oder nur eine unerhebliche Schnittmenge haben. Teilprobleme wären - "Boosting in der Suchmenge": Recommender bekommt Solr-Menge und macht nur darüber Recommendations (wenn überhaupt möglich - vielleicht kommt man ja in die erste Schleife der user-based algorithmen ï that u has no preference for yet") - "Boosting in der Recommender Menge": Recommender bestimmt Menge (OR query + Optionales query) und Solr boosted nur noch darin. (Use-Case widget) - SSelective Recommendation": Recommender wählt eine zum Query passendes Datenmodell / Datengrundlage aus, die möglichst viele potentielle Schnittmengen mit dem Query hat - Hier könnte man Jobs haben die regelmäßig die User-Item Datensätze nur für Items lernen die von der Suche zu einem bestimmtem Query zurückkommen (e.g. die Top-Querys) - Änreicherungsansatz"Der Recommender könnte zu Ergebnissen in der Suchmenge weitere Ergebnisse einstreuen (flgl "More like these"handler). - "Precomputation and Delegation to Solr Da find ich auch Interessant. Man kann sicherlich durch vorclustern ähnlicher Items oder ähnliche Items pro Usercluster alles direkt in einer Abfrage abwickelt (Durch anreichern der Daten im Index)

- Recommender empfehlen nur neue Items, man will aber ggf auch das bereits geklickte Items in der Suche höher gewichtet werden (so macht es Google ja auch) - Recommendation Algorithmen und Datengrundlage sind sehr verschieden. Wie kann man verschiedene Implementierungen nutzen/ansprechen/auswählen.

3.3 Datenerhebung

4 Realisation

.

.

Verfeinerung des Entwurfs, Schilderung bei der Umsetzung aus dem vorangegangenen Kapitel

4.1 Apache Mahout

4.1.1 Bestandteile

.

.

4.1.2 Datenaufbereitung

.

4.1.3 Skalierung

4.2 Apache Solr

4.2.1 Indexierung

4.2.2 Score-Anpassung

4.3 Searchperience Integration

.

.

5 Evaluation

.

.

.

.

.

Wie wird gemessen, welche Ergebnisse waren zu erwarten, was wurde erreicht. Warum gibt es Abweichungen, welche Probleme enthält die Messmethode.

5.1 Ergebnisse

5.2 Diskussion

6 Zusammenfassung

.

.

Abriss der Arbeit, was wurde erreicht bzw. gelernt. An welcher Stellen kann weitergearbeitet werden.

Annahme dass sich die Präferenzen der Nutzer über die Zeit nicht änder ist ggf. falsch

Abkürzungsverzeichnis

CF	Collaborative filtering

Abbildungsverzeichnis

Literatur

[Amatriain u. a. 2009] Amatriain, X. ; Jaimes, A. ; Oliver, N. ; Pujol, J. M.: Data Mining Methods for Recommender Systems. In: Kantor (Hrsg.) ; Ricci (Hrsg.) ; Rokach (Hrsg.) ; Shapira (Hrsg.): *Recommender Systems Handbook*, Springer, August 2009

[Bogers u. van den Bosch 2009] Bogers, Toine ; Bosch, Antal van d.: *Collaborative and Content-based Filtering for Item Recommendation on Social Bookmarking Websites.* 2009

[Burke 2002] Burke, Robin: Hybrid Recommender Systems: Survey and Experiments. In: *User Modeling and User-Adapted Interaction* 12 (2002), November, Nr. 4, 331–370. http://dx.doi.org/10.1023/A:1021240730564. – DOI 10.1023/A:1021240730564. – ISSN 0924–1868

[Chu u. a. 2006] Chu, Cheng T. ; Kim, Sang K. ; Lin, Yi A. ; Yu, Yuanyuan ; Bradski, Gary R. ; Ng, Andrew Y. ; Olukotun, Kunle: Map-Reduce for Machine Learning on Multicore. In: Schölkopf, Bernhard (Hrsg.) ; Platt, John C. (Hrsg.) ; Hoffman, Thomas (Hrsg.): *NIPS*, MIT Press, 2006, 281–288

[Jannach u. a. 2010] Jannach, D. ; Zanker, M. ; Felfernig, A. ; Friedrich, G.: *Recommender Systems: An Introduction.* Cambridge University Press, 2010. – ISBN 9780521493369

[Koren u. a. 2009] Koren, Yehuda ; Bell, Robert ; Volinsky, Chris: Matrix Factorization Techniques for Recommender Systems. In: *Computer* 42 (2009), August, Nr. 8, 30–37. http://dx.doi.org/10.1109/MC.2009.263. – DOI 10.1109/MC.2009.263. – ISSN 0018–9162

[Linden u. a. 2003] Linden, G. ; Smith, B. ; York, J.: Amazon.com recommendations: item-to-item collaborative filtering. In: *Internet Computing, IEEE* 7 (2003), Nr. 1, S. 76–80

[Lops u. a. 2011] Lops, Pasquale ; Gemmis, Marco ; Semeraro, Giovanni: Content-based Recommender Systems: State of the Art and Trends. Version: 2011. http://dx.doi.org/10.1007/978-0-387-85820-3_3. In: Ricci, Francesco (Hrsg.) ; Rokach, Lior (Hrsg.) ; Shapira, Bracha (Hrsg.) ; Kantor, Paul B. (Hrsg.): *Recommender Systems Handbook.* Springer US, 2011. – ISBN 978–0–387–85820–3, 73-105. – 10.1007/978-0-387-85820-3_3

[Machanavajjhala u. a. 2011] Machanavajjhala, Ashwin ; Korolova, Aleksandra ; Sarma, Atish D.: Personalized social recommendations: accurate or private. In: *Proceedings of the VLDB Endowment* 4 (2011), April, Nr. 7, S. 440–450. – ISSN 2150–8097

[Ricci u. a. 2010] Ricci, F. ; Rokach, L. ; Shapira, B. ; Kantor, P.B.: *Recommender Systems Handbook.* Springer, 2010. – ISBN 9780387858197

[Segaran 2007] Segaran, Toby: *Programming collective intelligence.* First. O'Reilly, 2007. – ISBN 9780596529321

[Sinha u. Swearingen 2001] Sinha, Rashmi R. ; Swearingen, Kirsten: Comparing Recommendations Made by Online Systems and Friends. In: *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001

[Smyth u. a. 2005] Smyth, Barry ; Balfe, Evelyn ; Boydell, Oisin ; Bradley, Keith ; Briggs, Peter ; Coyle, Maurice ; Freyne, Jill: A live-user evaluation of collaborative web search. In: *In IJCAI*, 2005, S. 1419–1424

[Smyth u. a. 2011] Smyth, Barry ; Coyle, Maurice ; Briggs, Peter: Communities, Collaboration, and Recommender Systems in Personalized Web Search. Version: 2011. http://dx.doi.org/10.1007/978-0-387-85820-3_18. In: Ricci, Francesco (Hrsg.) ; Rokach, Lior (Hrsg.) ; Shapira, Bracha (Hrsg.) ; Kantor, Paul B. (Hrsg.): *Recommender Systems Handbook*. Springer US, 2011. – ISBN 978–0–387–85820–3, 579-614. – 10.1007/978-0-387-85820-3_18

[Töscher u. a. 2008] Töscher, Andreas ; Jahrer, Michael ; Legenstein, Robert: Improved neighborhood-based algorithms for large-scale recommender systems. In: *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*. New York, NY, USA : ACM, 2008 (NETFLIX '08). – ISBN 978–1–60558–265–8, 4:1–4:6

[Victor u. a. 2011] Victor, Patricia ; Cock, Martine ; Cornelis, Chris: Trust and Recommendations. Version: 2011. http://dx.doi.org/10.1007/978-0-387-85820-3_20. In: Ricci, Francesco (Hrsg.) ; Rokach, Lior (Hrsg.) ; Shapira, Bracha (Hrsg.) ; Kantor, Paul B. (Hrsg.): *Recommender Systems Handbook*. Springer US, 2011. – ISBN 978–0–387–85820–3, 645-675. – 10.1007/978-0-387-85820-3_20

[Vidal 2005] Vidal, José M.: Trusting Agents for Trusting Electronic Societies. Version: 2005. http://dl.acm.org/citation.cfm?id=2137725.2137737. Berlin, Heidelberg : Springer-Verlag, 2005. – ISBN 3–540–28012–X, Kapitel A protocol for a distributed recommender system, 200–217

[Vozalis u. Margaritis 2007] Vozalis, M. G. ; Margaritis, K. G.: Using SVD and demographic data for the enhancement of generalized Collaborative Filtering. In: *Inf. Sci.* 177 (2007), August, Nr. 15, 3017–3037. http://dx.doi.org/10.1016/j.ins.2007.02.036. – DOI 10.1016/j.ins.2007.02.036. – ISSN 0020–0255

Index