Benjamin Hartwich

Evaluierung von Empfehlungssystemen

26. Februar 2014

Inhaltsverzeichnis

1 Einleitung

Wir alle haben ein Bedürfnis nach Information: ob tagesaktuelle Nachrichten, Facebook-Posts der Freunde oder die aktuellen Angebote auf Zalando. Für jede Informationsquelle gibt es zu viele Alternativen, als dass sie je in der verfügbaren Zeit erfahrbar sein werden. Suchmaschinen helfen vermutlich Relevantes zwischen diesem Überangebot zu finden. Doch was bedeutet Relevanz?

Personenbezogene Daten werden gesammelt: Welche IP-Adresse hat wann auf welche Ressource zugegriffen und wie lange? Über welche andere Ressource wurde der Inhalt gefunden, welches Betriebssystem, welcher Browser etc. wurde verwendet? Diese grundlegenden Informationen werden von fast jedem Server gespeichert. Ob Google oder die Webseite eines Newsportals: Man ist auf diese rudimentären Daten angewiesen. Doch dann zieht der Besucher weiter, dabei gäbe es noch viele weitere tolle Produkte oder Infos, die man ihm auf der eigenen Plattform anbieten könnte.

Empfehlungssysteme sind dafür zuständig, einzelnen Besuchern einer Webseite, die Produkte in der ein oder anderen Form anbietet - sei es ein e-Commerce-Portal wie *Amazon* oder eine Nachrichtenseite wie *Spiegel Online* - mehr Anreize zu geben, länger zu verweilen. Schließlich geht es meist um Geld: Egal ob Werbeanzeigen, die eventuell angeklickt werden oder um den virtuellen Einkaufswagen, der sich noch um ein Produkt mehr füllen könnte. Solche Empfehlungssysteme sind inzwischen Alltag, genauso wie ihre Verwendung durch den Nutzer. Schließlich ist diese Art individuell Interessantes zu finden ansatzweise aus realen Situation gewohnt, z.B. durch Werbekataloge. Der Einsatz digitaler Empfehlungssysteme beschränkt sich nicht nur auf die Webseiten selbst, sondern findet auch im E-Mail-Marketing Anwendung. Was früher der teure Versand von Katalogen erledigt hat, geht heute mit automatisierten Systemen, die Mails an ihre Kunden verschicken, in denen nach dem letzten Kaufverhalten auf die Person zugeschnittene Produkte empfohlen werden.

Aber kann ein Empfehlungssystem wirklich wissen, was der Einzelne möchte - nur weil einmal etwas gekauft wurde? Ist überhaupt sichergestellt, dass der Kunde nicht nur Informationen erhält, die lediglich das beinhalten, was er schon weiß? Sind solche Systeme nicht Teil der Informationsblase, die uns das Gefühl der eigenen Welt vermitteln soll, aber uns damit von möglichen weiteren Interessen abschottet?

In dieser Arbeit soll es nicht so sehr um die geisteswissenschaftliche Kritik an Empfehlungssystemen und ihrer Wirkung auf die Konsumenten gehen, sondern um die technische Evaluierung solcher Systeme. Grundlegend dafür ist eine kurze Einführung in die Grundbegriffe, was ein Empfehlungssystem ist und was unter *kollaborativen Filteralgorithmen* verstanden wird. Eine exemplarische Vorstellung solcher Algorithmen bildet den Anfang dieser Arbeit. Außerdem ist es wichtig, die Handlungsabläufe der Nutzer (*User Tasks*) auf Portalen, die solche Systeme einsetzen, zu verstehen sowie die Motivation, warum Produkte bewertet werden (*Rating Tasks*).

Der Hauptteil beschäftigt sich mit den Evaluierungsmetriken für einen erfolgreichen Test solcher Algorithmen. Daher werden grundlegend die Arten ei-

nes Datensets zum Test beschrieben und darauf aufbauend die Gütekriterien von Empfehlungssystemen. Ein weiterer Schwerpunkt dieser Arbeit wird in der Differenzierung zu neuen Metriken liegen, die nicht die Güte messen, sondern sich zentral an den Nutzerbedürfnissen orientieren. Zur Veranschaulichung wird das Filmempfehlungsportal *MovieLens* gewählt.

Ziel dieser Arbeit ist das Nachvollziehen der einzelnen Schritte einer Evaluierung von Empfehlungssystemen, die z.B. mit kollaborativen Filteralgorithmen arbeiten.

2 Grundbegriffe

Grundlegend für das Verständnis der weiteren Kapitel sind die Definitionen von Empfehlungssystem und kollaborativem Filteralgorithmus.

> *„Recommender systems use the opinions of a community of users to help individuals in that community more effectively identify content of interest from a potentially overwhelming set of choices.“*[5]

Grundlage für die Sinnhaftigkeit eines Empfehlungssystems ist zunächst einmal ein Überangebot an Informationen, z.B. ein Reiseportal, deren Anzahl an verschiedenen Reiseangeboten im vier- oder fünfstelligen Bereich liegt. Ob eine Suchmaschine in dieses Raster fällt, ist nicht eindeutig, denn es werden zwar Daten über Nutzer gesammelt und sicherlich auch Vergleiche zu anderen Besuchern gezogen, allerdings beinhaltet z.B. Google keine direkte Feedback- oder Bewertungsfunktion, die es dem Nutzer erlauben würde, Geschmacksprofile zu erstellen, wie das eher bei Shoppingportalen gewünscht ist. Dass eine Suchmaschine allerdings gar keinen Bezug zu einem Empfehlungssystem hat, ist auch nicht ganz richtig, denn eine Bewertung kann beispielsweise auch indirekt aus den angeklickten und nicht angeklickten Suchergebnissen generiert werden.

In dieser Arbeit konzentriere ich mich vor allem auf Portale, die ein Empfehlungssystem im klassischen Sinne verwenden: Amazon, MovieLens etc.

Ein kollaborativer Filteralgorithmus ist nur eine Möglichkeit, mit der ein Empfehlungssystem gestaltet werden kann. Darunter ist eine Berechnung zu verstehen, die interessante Empfehlungen für den Einzelnen anhand der Ähnlichkeit seines Geschmacks zu anderen Personengruppen auflistet.[1] Eine weitere Möglichkeit wäre z.B. ein *Incremental Collaborative Filtering*[2], der eine Weiterentwicklung des kollaborativen Filteralgorithmus darstellt:

> *„CF requires computations that are very expensive and grow polynomially with the number of users and items in a system. [...] To address this scalability problem, we present an incremental CF method, based on incremental updates of user-to-user similarities which is also able to recommend items orders of magnitude faster than classic CF [...].“* [8]

[1] Vgl. Breese, John S.; Heckerman, David; Kadie, Carl, 1998, S. 43.

[2] Vgl. Agarwal; Chen (2001): *ICML'11 Tutorial on Machine Learning for Large Scale Recommender Systems.*

Der *Multi-armed Bandit* Algorithmus sei noch kurz als weiteres Beispiel erwähnt, der im Konzept mit den Spielautomaten im Casino vergleichbar ist.[3].

Ich werde für diese Arbeit kollaborative Filteralgorithmen als Beispiel für die Evaluierung verwenden, da sie auch am einfachsten zu verstehen sind. Die Durchführung einer Evaluierung ist allerdings nicht an bestimmte Algorithmenarten gebunden, sondern sie läuft immer ähnlich ab.

3 Kollaborative Filteralgorithmen

Es können zwei Arten von kollaborativen Filteralgorithmen unterschieden werden: *memory-based* und *model-based* Algorithmen. Erstere arbeiten mit der gesamten Nutzerdatenbank und den darin gespeicherten Ratings, letztere lernen anhand der Daten ein Schema, das anschließend für die Vorhersagen verwendet wird. Ich möchte nun exemplarisch das Konzept zweier Algorithmen aus den zwei Sektoren vorstellen.[4]

Generell wird bei einem *memory-based* Algorithmus die Nutzerbewertung eines Items anhand der wenigen Informationen über den Nutzer vorhergesagt und mit einer Auswahl an Daten aus der Datenbank gewichtet. Dabei werden zum einen die Ratings des aktiven Nutzers mit den Ratings eines anderes Nutzers korreliert. Zum anderen kann das Konzept der Vektorähnlichkeit bei Suchergebnissen, das die Ähnlichkeit zweier Dokumente zum Ausdruck bringt, auf Empfehlungssysteme übertragen werden. Nutzer stellen hier die Dokumente da, Items die Worte und Bewertungen wären die Worthäufigkeit. Als Erweiterung könnte die *inverse user frequency* dienen:

> *„The idea is that universally liked items are not as useful in capturing similarity as less common items.“*[2]

Das *Bayesian Network Model* ist ein Beispiel für ein *model-based* Algorithmus. Hier wird ein Entscheidungsbaum (siehe Abbildung 1) aufgebaut, dessen Knoten aus den jeweiligen Items bestehen. Der Status eines jeden Knotens beinhaltet die möglichen Bewertungen für jedes Item.

> *„In the resulting network, each item will have a set of parent items that are the best predictors of its votes.“*[2]

[3] Vgl. Vermorel, Joann‘es; Mohri, Mehryar, 2005, S. 437f.
[4] Vgl. Breese, John S.; Heckerman, David; Kadie, Carl, 1998, S. 44ff.

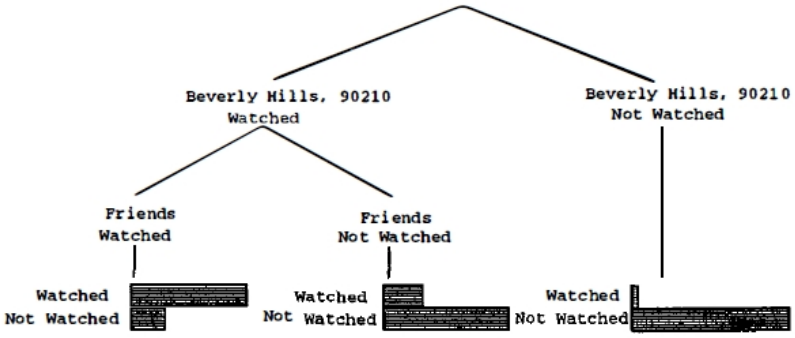

Abb. 1. Beispiel für einen Entscheidungsbaum. Die Skala am Ende des Baumes repräsentiert die Wahrscheinlichkeit von gesehen und nicht gesehen.[2]

Von jedem Knoten hängt die Wahrscheinlichkeit des nächsten Knotens ab. In einer Untersuchung von Breese und Heckerman hat sich gezeigt, dass das *Bayesian Network* und die Vektorähnlichkeit für verschiedenartige Datensätze sehr gut abgeschnitten haben.[5]

4 Handlungsabläufe und Motivation

Da Evaluierung mithilfe von Akkuranzmetriken zu kurz greift - wie ich noch darstellen werde - ist es wichtig, sich kurz mit den Nutzerinteressen bei Empfehlungssystemen zu beschäftigen. Diese Interessen beziehen sich einmal auf die Art der Empfehlung und zum zweiten auf die Motivation, warum überhaupt Produkte in einer Community bewertet werden.

4.1 User Tasks

Nutzer von Empfehlungssystemen haben ein Informationsbedürfnis, sonst würden sie diese Systeme nicht nutzen. Nun stellt sich die Frage, wie dieses Interesse genau befriedigt wird, denn letztlich kommt es bei der Evaluierung auch immer auf den Kontext des eingesetzten Algorithmus an und damit auf Handlungsabläufe des Nutzers. Wenn beides nicht aufeinander abgestimmt ist, werden keine guten Ergebnisse bei der Analyse erzielt. Ich werde nun nach Herlocker[6] die laut ihm wichtigsten *User Tasks* darstellen. Ich verwende dabei die englischen Fachbegriffe.

[5] Vgl. ebd., S. 48f.
[6] Vgl. Herlocker et. al., 2004, S. 9f.

- **Annotation in Context**: Ausgangspunkt für diesen Task ist die Bewertungsfunktion von Beiträgen in Foren. In diesem Fall geht es darum, dass gute Items für den Nutzer aus Sicht der Mehrheit ins Auge fallen sollen, ohne dass die bewerteten Items an sich aus ihrem Kontext gebracht werden, z.B. neu sortiert werden, wie das bei dem Forum *stackoverflow* der Fall ist. Für die Vorhersage einer Empfehlung ist es in diesem Zusammenhang besonders wichtig herauszufinden, wie gut der Nutzer anhand der Empfehlungen zwischen gewollter und nicht gewollter Information unterscheiden kann.
- **Find Good Items**: Ein Beispiel für ein solches Szenario könnten Artikelempfehlungen auf journalistischen Portalen sein. Dabei ist es wichtig, dass die ausgegebene Liste mit den Empfehlungen eine Rangordnung hat, so dass Items, die der Nutzer mit höherer Wahrscheinlichkeit mögen wird, oben in der Liste stehen werden. *Annotation in Context* und *Find Good Items* sind die am meisten untersuchten Nutzerhandlungsprozesse und werden auch dementsprechend häufig als Grundlage für Empfehlungssysteme eingesetzt.
- **Find All Good Items**: Bei *Find Good Items* kommt es darauf an, dass so viele gute Items wie möglich gefunden werden und der Nutzer eine fast vollständige Liste an relevanten Informationen zu einem Thema erhält. Wenn beispielsweise Rechtsanwälte in einem Online-Filter für Gerichtsurteile nach den Präzedenzfällen suchen, ist es äußert wichtig, dass der Algorithmus sehr liberal mit Bewertung von Relevanz und Nicht-Relevanz umgeht.
- **Recommend Sequence**: Ein klassisches Beispiel für einen solchen Handlungsprozess wäre *Last.fm*. Hier ist es von Nutzern gewünscht, dass sie eine auf sie zugeschnittene Wiedergabeliste abspielen können und nicht nur einzelne Songs vorgeschlagen bekommen. Es wird eine Sequenz an Empfehlungen ausgegeben. Dieser Task dürfte hauptsächlich im digitalen Audiobereich Anwendung finden, ist wissenschaftlich aber noch nicht so stark untersucht.
- **Just Browsing**: Während die bisher genannten *User Tasks* ein bekanntes Suchinteresse voraussetzen, geht es bei *Just Browsing* um den wahrscheinlichen Fall, dass ein Nutzer sich einfach nur mit den Inhalten einer Plattform beschäftigt, ohne z.B. ein festes Kaufinteresse zu haben. Dieser Task ist vergleichbar mit dem Stöbern in einem Bücher- oder Kleiderladen. Das Interesse besteht lediglich darin, zu sehen, was angeboten wird.
- **Find Credible Recommender**: Dass ein Nutzer einem unbekannten Empfehlungssystem traut, ist nicht von vornherein gegeben. Daher ist es nicht verwunderlich, dass auch der *Recommender* an sich getestet und mit anderen verglichen wird. Manche Nutzer gehen sogar soweit, dass sie mehrere Profile anlegen und beobachten, ob es bei den Empfehlungen zu Verzerrungseffekten kommt.

Wie die einzelnen *Tasks* erkennen lassen, wird es in der Wirklichkeit niemals genau trennscharfe Nutzerinteressen geben, da diese ja meist auch nicht vollbewusst durchgeführt werden, sondern nach den derzeitigen Bedürfnissen des Nutzers. Bedürfnisse sind überwiegend *stimulus*-gesteuert, was eine objektive Aussage schwer macht.[7] Es ist davon auszugehen, dass sich *User Tasks* über die

[7] Vgl. Schweiger, 2007, S. 60ff.

Zeit auch ändern können, da sie auch trendorientiert sein können. Das gilt es in Beobachtungen herauszufinden.

4.2 Rating Tasks

Grundlage für das Funktionieren von Empfehlungssystemen auf Basis von kollaborativen Filteralgorithmen ist das Bewerten von bestimmten Items durch Nutzer. Doch welche Motivation haben z.B. Besucher eines e-Commerce Portals, um dessen Produkte zu bewerten? Ich werde nach Herlocker[8] die für ihn wichtigen *Rating Tasks* darstellen. Auch hier werde ich die englischen Fachbegriffe dafür verwenden.

- **Improve Profile**: Da Nutzer gute bzw. bessere Empfehlungen erhalten möchten und diese auf ihrem Nutzerprofil aufbauen, liegt die Motivation genau in diesem Zusammenhang. Ein Beispiel wäre die Filmdatenbank *MovieLens*, bei der mit der Anzahl an bewerteten Filmen auch die richtige Vorhersagequalität für noch unbekannte, aber interessante Filme steigt.
- **Express Self**: In Bezug zu diesem *Rating Task* wäre der *User Task* „Annotation in Context“zu sehen, da Bewertungen z.B. in Foren immer die Möglichkeit zum Ausdruck einer eigenen Meinung darstellen, genauso wie das Bewerten von Restaurants. Da die persönliche Meinung gerade im Internet sehr direkt und unverblümt geäußert wird, spielt das Level an Anonymität eine wichtige Rolle zur Grundmotivation des Partizipierens an solchen *Rating Tasks*.
- **Help Others**: Manche Nutzer motiviert zum Bewerten, dass sie andere vor einem Produkt beispielsweise warnen können, da es schlechte Qualitätsmerkmale besitzt. E-Commerce Portale wie Amazon profitieren von diesem Motiv, was auch eng mit dem letzten *Rating Task* verknüpft ist, denn letztlich handelt es sich bei Hilfestellungen immer auch um das Interesse, das Gegenüber von einer bestimmten Meinung zu überzeugen.
- **Influence Others**: Um beim Beispiel von *Amazon* zu bleiben, bezieht sich dieses Motiv darauf, andere Nutzer zum Kauf eines bestimmten Produkts zu bewegen, z.B. durch falsche bzw. geschönte Bewertungen. Gerade bei diesem *Rating Task* ist es wichtig, dass das System solche Einflussversuche erkennt und diesen vorbeugen kann - beispielsweise durch eine kompliziertere Weise, sich zu registrieren, um dann erst Bewertungen abgeben zu können.

Die vorgestellten *Tasks* sind für die Evaluierung entscheidend, da ein Empfehlungssystem immer nur bestimmte *Rating Tasks* zulässt und bei der Auswahl der Testmethoden darauf geachtet werden muss, welche Handlungsabläufe von Nutzern ausgeführt werden und welche Analyseoperatoren sich am besten zur Bestimmung der Qualität des Systems eignen. Grundlage dafür ist die richtige Auswahl des Algorithmus, der sich auch am Nutzerinteresse und der Art an Daten orientieren muss.

[8] Vgl. Herlocker et. al., 2004, S. 11f.

5 Start einer Evaluierung

Bevor ich auf die einzelnen Kriterien der Evaluierung eingehe, möchte ich eine kurze Übersicht geben, was die Kernpunkte bei der Durchführung bzw. Planung sind.[9]

Wie bereits erwähnt, soll eine Evaluierung Aufschluss darüber geben, ob ein Algorithmus, der die Empfehlungen berechnet, das auch qualitativ wertvoll für den jeweiligen Nutzer macht. In dieser Aussage stecken bereits erste Anforderungen: Sie setzt voraus, dass der Zweck der Plattform, auf der der Algorithmus eingesetzt wird, eindeutig definiert ist. Dazu muss bekannt sein, was das Interesse der Nutzer generell und im speziellen an der Plattform ist (siehe 4.1).

Um überhaupt etwas bewerten zu können, muss klar sein, was eine gute Bewertung ist und die kann sich aus folgenden Faktoren verschieden gewichtet ergeben: der Nutzer und seine Interessen, der Plattformbetreiber und seine Interessen, die technischen Voraussetzungen und die Information, die bewertet bzw. vorgeschlagen wird. Wenn beispielsweise *Find all good items* Ziel einer Suche ist, muss die Analyse der Ergebnisse an der Genauigkeit ansetzen. Sollte es *nur* um Filmvorschläge gehen, sind Fehler im Empfehlungssystem eher zu akzeptieren und fallen auch dem Nutzer weniger auf.

Es gilt immer zwei Seiten zu betrachten: Was ist für den Besucher einer Webseite eine gute Empfehlung, was ist für das System - den Algorithmus - eine gute Empfehlung? Dieses Verhältnis zu finden ist die Anwendung einer abgeschlossenen Evaluierung. Daher ist das Ziel einer solchen entscheidend: Soll ein Algorithmus für eine noch nicht existente Webseite gefunden werden oder soll er auf einer bereits seit mehreren Jahren existierenden Webseite verbessert werden? Je nach Ausgangspunkt müssen hier die Analysekriterien anders gesetzt und bewertet werden. Doch anhand welcher Daten?

6 Datensets

Ein Algorithmus kann nur so gut sein, wie das Datenset, das ihm zur Verfügung steht bzw. ihn trainiert. Das heißt allerdings auch, dass es für eine Evaluierung keinen Sinn macht, einen Algorithmus mit den Bewertungsschemata von z.B. dem Forum *stackoverflow* zu trainieren, wenn er später bei einem e-Commerce Portal zum Einsatz kommt. Da die meisten Plattformen, die einen Empfehlungsalgorithmus einsetzen, ein *Cold Start* Problem haben, wird es immer zu leichten Verzerrungen kommen.

Generell ist die Dichte der Daten zu beachten - das Verhältnis zwischen Nutzerzahl und Anzahl der Bewertungen. Wie stark muss der Algorithmus mit neuen Daten für Empfehlungen umgehen können? Wie werden die Daten gesammelt: Explizit - z.B. Bewertungen der Nutzer - oder implizit - z.B. Auswerten des Nutzerverhaltens durch Serverlogs? Diese und weitere Anforderungen (siehe 5) werden auch durch die Art des zustande gekommenen Datensets bestimmt.

[9] Vgl. Herlocker et. al., 2004, S. 13f.

6.1 Studien über den Nutzer

Gerade bei ganz neuen Plattformen, die bisher bekannte Prinzipien und Trends im Internet aufbrechen oder völlig neue Nutzeroberflächen entwickelt haben, für die sind Studien über den Nutzer sinnvoll. Das sind Daten, die mithilfe von Beobachtungen oder Experimenten mit Testpersonen ermittelt werden - qualitative Datenerhebungen.

> *„Qualitative Verfahren beschreiben ein komplexes Phänomen in seiner ganzen Breite.“*[3]

Ein typisches Szenario wäre: Es werden Testpersonen mit bestimmten soziodemographischen und speziellen Merkmalen angeworben. Diese führen anhand einer möglichen Vorgabe Operationen am System aus und währenddessen werden Daten ermittelt, z.B. wie lang bestimmte Operationen gedauert haben, ob ein Ziel erreicht wurde, welche Probleme oder Fehler aufgetreten sind etc.[10] Solche Experimente bzw. Beobachtungen dienen in erster Linie zur Evaluierung der Nutzeroberfläche und dem Herausfinden der Nutzerinteressen in Abhängigkeit von der Plattform. Beispielsweise könnte so untersucht werden, ob ein neuer Empfehlungsalgorithmus das Nutzerverhalten ändert und wenn ja, wie.

Eine anschließende Befragung kann weitere Intentionen geben und auf bisher ungeahnte Probleme aufmerksam machen. Allerdings ist eine solche Studie sehr speziell, d.h. es können keine standardisierten Daten in großer Zahl über mehrere Merkmale gesammelt werden, womit die Interpretation der Daten aufwendig und nicht immer verallgemeinerbar ist. Das macht diese Art der Untersuchung auch kostenintensiver als andere Modelle. Daher kann auch nur eine Auswahl an Aktionen im System evaluiert werden, wodurch ein Pre-test zur Sicherung des intendierten Forschungsziels ratsam ist. Wichtigstes Kriterium für eine erfolgreiche Analyse ist die Auswahl der Testpersonen. Hier können leicht Bias-Effekte etc. entstehen, wenn die Personen nicht die spätere Zielgruppe der Plattform repräsentieren.[11]

Eine Evaluierung sollte nicht nur aus den Daten einer Nutzerstudie bestehen, sondern entweder mit Offline oder Online Datensets kombiniert werden.

6.2 Offline Datenset

Ein Offline Datenset wird meist dazu verwendet, um einen Algorithmus auf ein Schema zu trainieren, so dass seine Vorhersagekraft gemessen werden kann. Ein solches Set an Daten wären z.B. die Filmbewertungen der Nutzer von *MovieLens* innerhalb eines bestimmten Zeitraums. Mit diesen Bewertungen könnte man nun verschiedene Algorithmen für eine Online-Videothek testen - jedoch eher weniger für Restaurantvorschläge.

> *„[...] We assume that the user behavior when the data was collected will be similar enough to the user behavior when the recommender system is*

[10] Vgl. Shani, Guy; Gunawardana, Asela, 2011, S. 7.
[11] Vgl. ebd., S. 8.

> *deployed, so that we can make reliable decisions based on the simulation.“*[11]

Bei einer Evaluierung wird ein Zeitpunkt in dem Datenset ausgewählt, ab dem alle zukünftigen Ereignisse ausgeblendet werden. Es werden nicht alle Daten verwendet, sondern Fälle gezogen. Der Algorithmus wird mit den Daten bis zu diesem Zeitpunkt trainiert und soll anschließend die ausgeblendeten Daten neu berechnen. Je nach Stärke der Übereinstimmung zwischen den ausgeblendeten Daten und dem, was berechnet wurde, eignet sich der Algorithmus weniger oder mehr für Empfehlungen. Dabei können Bewertungen oder Nutzeraktionen ausgeblendet werden.[12]

Ein wesentlicher Nachteil dieses Vorgehens ist die Frage, inwieweit ein aus bestimmten Nutzeraktionen aggregierter Datensatz als Evaluierungsinstrument für den Untersuchungszweck in einer anderen Umgebung verwendet werden kann. Im Gegensatz zu Studien über den Nutzer könnte eine Evaluierung mit einem Offline Datenset ausreichend sein.

6.3 Online Datenset

> *„In many realistic recommendation applications the designer of the system wishes to influence the behavior of users.“*[11]

Online Datensets dienen in erster Linie dazu, bei der Weiterentwicklung einer Plattform das Nutzerinteresse besser zu treffen und dadurch evtl. zu manipulieren. Zu diesem Zweck wird ein kleiner Prozentsatz der Nutzer via Losverfahren auf eine Seite umgeleitet, bei der das Empfehlungssystem auf irgendeine Weise anders gestaltet wurde. Die Nutzerdaten, die auf diese Weise gesammelt werden, bilden den Online Datensatz. Dieser kann nun mit dem Datensatz, der eigentlichen Plattform verglichen werden, um Performance oder Qualitätsunterschiede feststellen zu können.

Ein gutes Empfehlungsergebnis hängt allerdings nicht nur vom Algorithmus selbst ab, sondern von der Intention und Erfahrung des Nutzers sowie dem Design der Nutzeroberfläche. Eine Evaluierung sollte nicht nur mit einem Online Datenset durchgeführt werden.[13]

7 Metriken

Wie wird nun ein Datensatz evaluiert? Ich werde verschiedene Arten von Metriken vorstellen. Allgemein habe ich diese in *Akkuranzmetriken*, *Ranking Metriken* und *Neue Metriken* unterteilt, wobei erstere nach dem klassischen *Information Retrieval* Konzept arbeiten, *Ranking Metriken* die Qualität eines Rankings bestimmen und letztere mehr die Nutzerintentionen berücksichtigen. Ich werde im Folgenden die englischen Fachbegriffe verwenden.

[12] Vgl. ebd., S. 5f.
[13] Vgl. ebd., S. 10f.

7.1 Akkuranzmetriken

> *„An accuracy metric empirically measures how close a recommender system's predicted ranking of items for a user differs from the user's true ranking of preference.“*[5]

Es wird die Genauigkeit der Vorhersage gemessen. Beispielsweise kann so bei der Filmdatenbank *MovieLens* ermittelt werden, wie nah das empfohlene Rating zu einem Film, den der Nutzer noch nicht gesehen hat, an dem tatsächlichen Rating des Nutzers ist. Diese Genauigkeit wird allgemein über den *Mean Absolute Error* berechnet, der die durchschnittliche Standardabweichung zwischen dem vorhergesagten und dem wahren Nutzerrating misst.[14]

$$|E| = \frac{\sum_{i=1}^{N} |p_\mathrm{i} - r_\mathrm{i}|}{N} \tag{1}$$

Der *Mean Absolute Error* kann je nach dem, wie genau die Ergebnisse in richtig und falsch eingeteilt werden, schnell sehr groß werden oder weniger stark ausfallen. Die Detailgenauigkeit entscheidet damit auch über die Akzeptanz von Fehlern.

Klassifizierung Welches Ergebnis ist korrekt im Sinne der Empfehlung, welches nicht? Da die Akkuranzmetriken ein relativ altes Bewertungsmodell darstellen, ist es zum Verständnis besser, von einem endlichen Datensatz auszugehen, z.B. einer Art Dokumentensuchmaschine, die prüft, wie gut ein Ergebnis mit der Anfrage zusammenpasst. Um das bewerten zu können, müssen alle Einträge nach einem Muster klassifiziert werden können.

	Ausgewählt	**Nicht ausgewählt**	**Total**
Relevant	wahr-positiv	wahr-negativ	N_R
Nicht relevant	falsch-positiv	falsch-negativ	N_{NR}
Total	N_A	N_{NA}	N

Tabelle 1. Klassifizierung von Ergebnissen

Die Optionen *Ausgewählt* und *Nicht Ausgewählt*, was im Kontext von Empfehlungssystem mit *Empfohlen* und *Nicht Empfohlen* übersetzt werden muss, beschreiben die Entscheidung des Systems, welches Item für die Ergebnisliste definiert wurde oder nicht. Mit *Relevant* und *Nicht Relevant* ist die Sicht des Nutzers gemeint: Ist das Ergebnis für ihn relevant? Daraus ergeben sich verschiedene Kombinationen, die sich mit *Precision und Recall* rechnerisch bewerten lassen.

[14] Vgl. Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.;Riedl, John T., 2004, S. 20f.

Das Hauptproblem dieser Klassifizierung ist im Falle einer Empfehlung, z.B. ein Kinofilm, die Übertragung einer Geschmacksbewertung: 5 Sterne Rating in ein binäres Schema. Wie wird festgelegt, welches Rating der Option *Relevant* entspricht? Noch unklarer wird es bei Mehrfachbewertungen für ein Item, z.B. bei einem Hotel. So ergeben sich Probleme wie Unschärfe der Bewertung, der Klassifizierung von noch nicht bewerteten Items, der Umgang mit neuen Daten oder ein schlechter Vergleich zwischen Items mit sehr hoher Bewertung.[15]

Precision und Recall Precision und Recall sind Kennwerte, die sich aus der vorhergehenden Klassifizierung zusammensetzen. Voraussetzung dafür ist, dass alle Items der Ergebnisliste betrachtet und bewertet werden, was in der Regel nicht der Fall ist. Hier setzen die *Neuen Metriken* an (siehe 7.3). Precision beschreibt das Verhältnis von relevanten Dokumenten im Kontext zu allen gefundenen Ergebnissen.

$$Precision = \frac{wahr - positiv}{wahr - positiv + falsch - positiv} \tag{2}$$

Recall ist die Anzahl an gefundenen relevanten Ergebnissen in Abhängigkeit von der Gesamtheit an relevanten Items.

$$Recall = \frac{wahr - positiv}{wahr - positiv + wahr - negativ} \tag{3}$$

Dieses Modell ist zwar eine Grundlage für Tests mit Evaluierungsalgorithmen, allerdings vom Prinzip her veraltet:

> *„Most information retrieval evaluation has focused on an objective version of relevance, where relevance is defined with respect to a query, and is independent of the user.“*[5]

Für Empfehlungssysteme macht diese Vorgehensweise keinen Sinn, da Geschmack und Intention eines Nutzers nicht pauschalisierbar sind. Ein Beispiel war bereits die Codierung eines Ratings in ein binäres System - auch hier kann die Relevanz subjektiv interpretiert werden. Für eine Evaluierung mithilfe von Precision und Recall müsste grundlegend ein Konsens zwischen allen Nutzern über die Einteilung ihres Geschmackes und der Relevanz eines Items zu diesem Geschmack herrschen, denn Recall unterstellt die Bekanntheit aller relevanten Items für eine Empfehlung.

Dabei bedingen sich beide Werte gegenseitig: Wenn Recall zunimmt, sinkt Precision und die Länge der Ergebnisliste beeinflusst die Werte. Bei Betrachtung mehrerer Berechnungen hilft der *F Score* bzw. *F1 Score*, der Precision und Recall als eine Zahl ausdrückt. Der *F1 Score* ist zur Berechnung mit einem harmonischem Verhältnis gedacht, beim *F Score* können die beiden Werte gewichtet werden. [16]

$$F1 = \frac{2PR}{P + R} \tag{4}$$

[15] Vgl. ebd., S. 21f.
[16] Vgl. ebd., S. 23ff.

ROC Curves Zur besseren Analyse der Berechnungen im vorigen Kapitel können die Ergebnisse auch visualisiert werden: mit einer Precision Recall Kurve oder mit ROC Kurven. Erstere stellen den Anteil zwischen empfohlenen und gemochten Items dar, letztere das Verhältnis zwischen nicht gemocht, aber empfohlenen Items. So kann z.B. mit einer Precision Recall Kurve evaluiert werden, ob der Recommender gute Filmvorschläge liefert, während eine ROC Kurve eher darauf ausgelegt ist, einen womöglich hohen Anteil an falschen Empfehlungen zu visualisieren. Einsatzgebiet könnte hier das E-Mail Marketing sein.[17] Auf ROC Kurven werde ich kurz näher eingehen.

ROC steht für *receiver operating characteristic* und beschreibt bei einem System, wie gut es beim Messen zwischen Signal und Nicht-Signal unterscheiden kann. Übertragen auf Empfehlungssysteme kann diese Kurve zur Darstellung von relevant und nicht-relevant verwendet werden. Was die ROC Kurve so besonders macht, ist der *Cutoff*. Er repräsentiert die Stelle, an der der Nutzer das Lesen der Ergebnisliste abbricht - es wird damit die Länge der Liste festgesetzt. Vereinfacht ausgedrückt entsteht die Kurve durch folgendes System: Zeichne einen Strich nach oben in einem Koordinatensystem, wenn das Item relevant ist und zeichne einen Strich horizontal nach rechts, wenn es nicht relevant ist. Wie die einzelnen Items dabei angeordnet sind, hat keinen Einfluss auf die Kurve.[18]

[17] Vgl. Shani, Guy; Gunawardana, Asela, 2011, S. 17f.

[18] Vgl. Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.;Riedl, John T., 2004, S. 25ff.

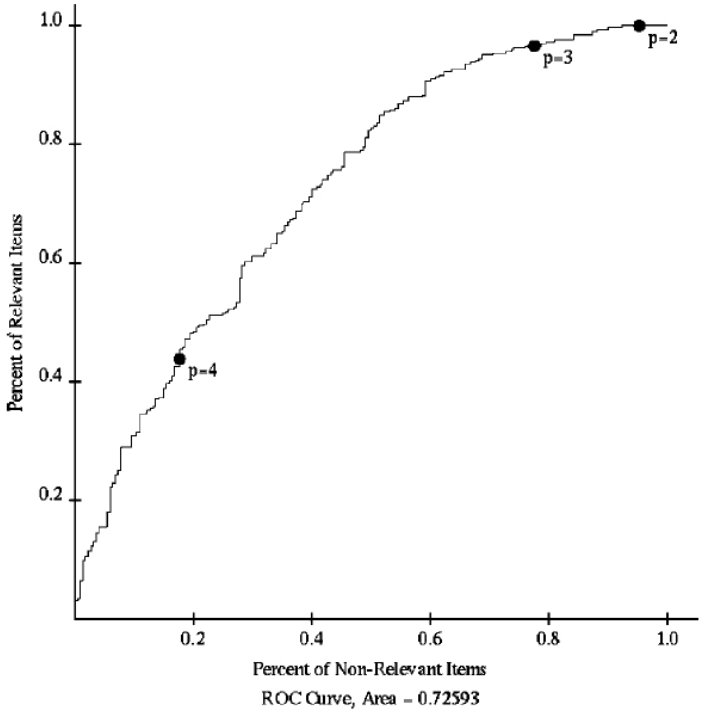

Abb. 2. Beispiel einer ROC-Kurve. Die p-Werte symbolisieren den Cutoff an Items, die mit einer Vorhersage von z.B. dem Rating 4 versehen wurden.[5]

Abbildung 2 zeigt ein Beispiel für die Relevanzverteilung der Empfehlungen in einer Filmdatenbank. An der y-Achse sind die Werte für *Relevanz*, an der x-Achse für *Nicht-Relevanz* eingezeichnet. Für p=4 würden etwas mehr 40% relevante Items gefunden werden, aber auch ca. 20% an nicht-relevanten.

7.2 Ranking Metriken

Ranking Metriken messen die Fähigkeit eines Algorithmus, Empfehlung nach der Ordnung zu sortieren, wie es dem Geschmack des Nutzers entspricht. Im Gegensatz zur Klassifizierung kann hier ein Algorithmus besser evaluiert werden, da der Nutzertask nicht mehr bei *Annotation in Context* liegt, sondern mehr bei *Find Good Items*. Generell unterscheidet man zwischen *Reference Ranking* Metriken und *Utility Based* Rankings.[19] Bei einem *Reference Ranking* wird grundlegend eine Korrelation zwischen der empfohlenen Liste und der vom Nutzer präferierten Liste berechnet. Mathematische Modelle hierfür wären die *Pearson Korrelation*,

[19] Vgl. Shani, Guy; Gunawardana, Asela, 2011, S. 19ff.

Spearman's ρ und *Kendall's Tau*. Beim ersten wird ein linearer Zusammenhang zwischen beiden Variablen berechnet, bei den anderen Modellen, inwieweit beide Rankings unabhängig von aktuellen Werten korrelieren.[20] Genauer werde ich die mathematischen Modelle in dieser Arbeit nicht beschreiben.

Das Konzept der *half-life utility metric* evaluiert die Nützlichkeit der geordneten Liste als Unterschied zwischen dem allgemeinen Rating eines Items und der Bewertung des Nutzers.

> *„The likelihood that a user will view each successive item is described with an exponential decay function, where the strength of the decay is described by a half-life parameter. [...] The half-life is the rank of the item on the list such that there is a 50% chance that the user will view that item.“*[5]

Es handelt sich damit um ein Beispiel für ein *Utility Based* Ranking. So lässt sich dieses Modell eher weniger für *Find All Good Items* einsetzen, während die Korrelationsmodelle die Länge der betrachteten Liste nicht berücksichtigen. In einer weiteren Online-Evaluierung kann bestimmt werden, welche Darstellung relevanter Items am besten geeignet ist.

7.3 Neue Metriken

Bisher habe ich Verfahren beschrieben, die Wahrscheinlichkeiten für das Treffen eines Nutzergeschmacks zu berechnen. Diese Verfahren sind jedoch an endlichen Datensets orientiert - anders könnte die Relevanz nicht bestimmt werden. Die Evaluierung richtet sich hauptsächlich an die Qualität des Systems bzw. des Algorithmus. Was nicht berücksichtigt wird, ist die Intention des Nutzers in verschiedenen Themenkontexten. Wenn beispielsweise Reiseorte empfohlen werden sollen, wie gelangen Orte, die der Nutzer nicht kennt, in die Empfehlung?[21]

Neue Metriken evaluieren mehr die Nutzerinteressen und sind als Erweiterung zu den Akkuranzmetriken zu verstehen. Im Folgenden werde ich die wichtigsten nach Herlocker[22] und Shani[23] kurz vorstellen. Ich werde auch hier die englischen Fachbegriffe verwenden.

Coverage Es kann zwischen *Item Space* und *User Space* Coverage unterschieden werden. Mit *Item Space Coverage* kann gemessen werden, über welchen Anteil der Items vom Gesamtsystem eine Vorhersage getroffen werden kann. Das wird auch als *Catalog Coverage* bezeichnet. Bei einer Evaluierung kann so z.B. der

[20] Vgl. Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.;Riedl, John T., 2004, S. 29ff.

[21] McNee, Sean M.; Lam, Shyong K.; Guetzlaff, Catherine; Konstan, Joseph A.; Riedl, John, 2003, S. 1098.

[22] Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.;Riedl, John T., 2004, S. 39ff.

[23] Vgl. Shani, Guy; Gunawardana, Asela, 2011, S. 24ff.

Prozentsatz an Items ermittelt werden, der empfohlen wird. Die Güte der Coverage kann mit dem *Gini Index* oder der *Shannon Entropie* bestimmt werden.

Bei der *User Space Coverage* geht es hingegen um den Prozentsatz an Nutzern vom Gesamtsystem, für die eine Empfehlung erstellt werden kann. Eine Möglichkeit der Untersuchung ist die Detailgenauigkeit eines Nutzerprofils. Je nach der Nutzeranzahl, die abgedeckt wird, beeinflusst das natürlich auch die Akkuranzmetriken - der richtige Trade-off ist hier entscheidend. *Cold start* Probleme fallen ebenfalls in den Bereich der Coverage, da es sich dabei um neue Items oder Nutzer handelt, über die noch keine Aussagen, Bewertung bzw. Empfehlung getroffen wurde. Es muss lediglich bestimmt werden, was *Cold Items* bzw. *Cold User* sind. So kann ein möglicher Unterschied zwischen der Akkuranz des *Cold start* und des Gesamtsystems berechnet werden.

Learning Rate Learning Rate schließt an Coverage an und beschreibt die Eigenschaft eines Algorithmus anhand veränderter oder neuer Daten das Empfehlungsmodell anpassen zu können. Es gibt die *Overall Learning Rate*, die die Qualität einer Empfehlung als Funktion über alle Ratings im System beschreibt. Die *Per Item Learning Rate* bestimmt die Qualität einer Empfehlung für ein Item als Funktion über die Anzahl der vorhandenen Ratings und die *Per User Learning Rate* definiert die Qualität einer Empfehlung für einen Nutzer als Funktion über die Anzahl der Ratings, die der Nutzer gemacht hat.

Um die Learning Rate zu evaluieren, wird meist ein Graph aus der Qualität und der Anzahl an Bewertungen aufgespannt, allerdings ist dieses Thema in der Literatur bisher wenig diskutiert. Dieses Modell ist jedoch für die meisten Empfehlungssysteme notwendig, die mit wenigen Daten arbeiten müssen.

Confidence Confidence beschreibt die Sicherheit des Systems darüber, wie stark die gegebene Empfehlung für den jeweiligen Nutzer gültig ist. Um die Qualität der Confidence zu verbessern, benötigt das System schlicht mehr Daten für den Empfehlungsfall, in dem es unsicher ist. Das kann einerseits mittels einer Art T-Test evaluiert werden, d.h. die Wahrscheinlichkeit zu messen, ob der wahre Vorhersagewert innerhalb eines 95% Konfidenzintervalls liegt. Eine andere Variante wäre die komplette Auflistung aller möglichen Empfehlungsergebnisse.

Diese Metrik kann auch für den Nutzer dargestellt werden, wie in Figur 3 zu sehen ist. Es handelt sich um die Plattform *MovieLens*, die für empfohlene Filme mit niedriger Confidence Würfel als Kenntlichmachung verwendet. Diese Art der Visualisierung erhöht so u.a. das Vertrauen der Nutzer in das System, was im nächsten Punkt behandelt wird.

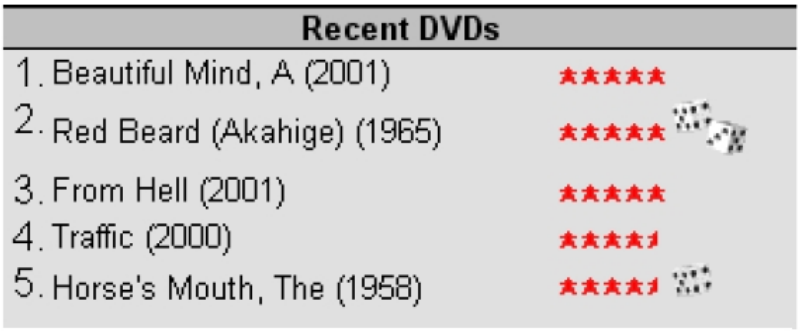

Abb. 3. Beispiel einer Confidence Visualisierung bei MovieLens.[7]

Trust Wie bereits erwähnt, handelt es sich bei Trust um das Vertrauen der Nutzer in das Empfehlungssystem. Eine richtige Berechnung zur Evaluierung kann hier nicht vorgenommen werden, sondern es müsste in Befragungen bzw. Experimenten ermittelt werden. Eine Kennzahl könnten allerdings wiederkehrende Nutzer in der Besucherstatistik sein.

Novelty Wie gelangen neue Items in die Empfehlungsliste? Gerade für Musikportale wäre das ein wichtiges Kriterium. Das könnte so umgesetzt werden, dass Items, die der Nutzer bereits bewertet oder gesehen hat, nicht mehr empfohlen werden. Doch was von den übrig gebliebenen Empfehlungen ist nun relevant? Hier setzen die Akkuranzmetriken an, die dabei helfen, dass keine irrelevanten neuen Items empfohlen werden. Weiterhin können bei der Berechnung einer Empfehlung mehr Punkte für relevante und neue Produkte vergeben werden als für populäre Items.

Serendipity Serendipity und Novelty werden oft synonym verwendet. Ich möchte sie hier bewusst trennen, obwohl beide für den Nutzer unbekannten Content erzeugen sollen. Ich würde Serendipity als Spezialform der Novelty sehen, da solche Empfehlungen über Distanzberechnungen entstehen, d.h. wie weit ist ein bestimmtes Item von einem bisher bewerteten Set an Items entfernt. Damit ist also der Zufallseffekt bzw. Überraschungseffekt einer Empfehlung gemeint. Gerade bei Buchvorschlägen wird dieses Modell gerne eingesetzt, da sich aufgrund der Metabeschreibungen eines Buches - Genre, Autor, Jahr etc. - sehr leicht eine Distanz zu anderen Genres oder ähnlichem finden lässt.

Auch hier muss wieder mit Hilfe der Akkuranzmetriken die Relevanz ermittelt werden und könnte ein höherer Score für Empfehlungen vergeben werden, die Serendipity erfüllen.

Diversity Diversity ist als Gegenteil von Ähnlichkeit definiert. Wenn beispielsweise eine Liste mit fünf Empfehlungen ausgegeben wird, kann es für ein Rei-

seportal sinnvoller sein, wenn sich die einzelnen Items mehr unterscheiden und nicht verschiedene Hotels im gleichen Ort abbilden, sondern Hotels in verschiedenen Orten. Für die Evaluierung wird die Ähnlichkeit der Items zueinander berechnet, was hauptsächlich über die Metainformationen im System erledigt wird. Außerdem kann der Unterschied zwischen Items mit Hilfe der Summe, dem Minimum, Maximum, Durchschnitt etc. gemessen werden. Das invertierte Verhältnis zwischen Diversity und Akkuranz kann auch als Graph visualisiert werden.

Weitere Metriken Es gibt noch weitere sechs Metriken, die ich hier jeweils in ein, zwei Sätzen zusammenfassen möchte, da sie bei einer Evaluierung meiner Meinung nach eher den Feinschliff ausmachen. *Utility* ist eine Aussage darüber, ob bei einer Empfehlung eher das System oder der Nutzer profitiert. Beispielsweise könnte der Sinn einer Empfehlung auf einem e-Commerce Portal sein, dass mehr gekauft wird anstatt eine hohe Akkuranz zu haben. *Risk* beschreibt die Kosten einer Empfehlung zu vertrauen, d.h. ob es den Nutzer z.b. etwas kosten würde, wenn er der Empfehlung folgt und welches finanzielle Risiko damit verbunden wäre. *Robustness* meint die Stabilität des System bei Fehl- bzw. Fake-Informationen sowie das Verhalten bei extremen Situationen, z.B. hohen Zugriffen.

Der Datenschutz von Nutzerprofilen bei Empfehlungssystemen fällt unter die Metrik *Privacy*. Dazu zählt einmal die Möglichkeit, dass Nutzer individuelle Privatsphäreeinstellungen vornehmen können und zum anderen, dass auf theoretische Lücken zum Ausspionieren des Systems hin geprüft wird. *Adaptivity* meint die Anpassungsfähigkeit des Algorithmus auf neue Nutzerinteressen. Diese Metrik beschreibt auch die Schwelle, ab der sich das System an den persönlichen Geschmack des Nutzers angepasst hat. Ein Empfehlungssystem wächst und verändert sich, Zugriffe wachsen, Datensets werden größer. *Scalability* evaluiert die Ressourcen des Empfehlungssystems.

8 Fazit

Womit kann ein System arbeiten, was kann es wie verarbeiten, was erwartet der Nutzer, wie reagiert er? Das sind die zentralen Fragen einer Evaluierung von Empfehlungssystemen. Das wichtige Verständnis zwischen System- und Nutzerlogik habe ich dargelegt, denn das skizziert auch den Trend der weiteren Forschung. Das klassische *Information Retrieval* wird immer Grundlage bleiben, da in einer binären Umgebung die Beschreibung von komplexen Zusammenhängen Grenzen hat. Die neuen Metriken betrachten dieses klassische Modell immer aus einem Nutzerinteresse heraus, was für jede Anwendung heutzutage Grundlage sein muss, da jeder Internetnutzer inzwischen Systeme wie Amazon, Zalando oder Facebook gewohnt ist. Nicht, dass diese Portale das Nonplusultra widerspiegeln und keinen Platz für eine Verbesserung lassen, ganz im Gegenteil. Doch sie bilden die Grundlage einer sozial-technischen Erwartungshaltung.

Dass die Anforderungen an ein Empfehlungssystem nicht nur ökonomischen Anforderungen genügen sollten, sondern menschliches Verhalten und Denken prägen, ist an dieser Stelle zwar nicht diskutiert worden. Ich möchte dennoch als Abschlussgedanken auf die ethisch-soziale Ebene eingehen, was bei der Monopolstellung der großen Internetfirmen in Sachen Kommunikation durchaus relevant ist. Empfehlungssysteme bilden immer Systemdomänen ab, d.h. geschlossene Systeme. Wie bleibt jedoch sichergestellt, dass Nutzer nicht nur Informationen finden, die ihrem gewohnten Schema entsprechen, sondern ihr Denken auch erweitern? Gibt es eine Verantwortung der Betreiber von großen Portalen, Informationsblasen entgegenzuwirken, auch wenn die ökonomische Leistung beeinträchtigt werden könnte? Rein technisch würde das funktionieren.

Noch ist nicht untersucht, welche sozialen Wirkungen bestimmte Internetanwendungen haben, da sie dafür zu kurz existieren, zu dynamisch sind. Dennoch hinterlassen sie Denk- und Handlungsstrukturen in Menschen. Diese mehr miteinzubeziehen und einen leicht ideellen Charakter des Internets auch im ökonomischen Umfeld weiterleben zu lassen, wäre eine wissenschaftliche Untersuchung wert.

> *„The future of new media ethics is intrinsically tied to democratic decision-making processes about the direction, risks, and impositions of new ICT. This would require informed technological citizenship where the responsibilities and the privileges – and perhaps rights – associated with living in a world suffused with technology are a matter for ethical reflection and political practice.“*[4]

Literatur

1. Agarwal, Deepak; Chen, Bee-Chung (2011): *ICML'11 Tutorial on Machine Learning for Large Scale Recommender Systems.*
   URL: `http://pages.cs.wisc.edu/~beechung/icml11-tutorial/`, zuletzt gesichtet am 05.01.2014.
2. Breese, John S.; Heckerman, David; Kadie, Carl (1998): *Empirical Analysis of Predictive Algorithms for Collaborative Filtering.*
   URL: `http://research.microsoft.com/pubs/69656/tr-98-12.pdf`, zuletzt gesichtet am 05.01.2014.
3. Brosius, Hans-Bernd; Haas, Alexander; Koschel, Friederike (2012): *Methoden der empirischen Kommunikationsforschung. Eine Einführung.* 6. Auflage, Wiesbaden: VS Verlag für Sozialwissenschaften.
4. Debatin, Bernhard (2010): *New Media Ethics.* In: Brosda, Carsten; Schicha, Christian (Hg.): Handbuch Medienethik. Wiesbaden, S. 318-330.
5. Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.;Riedl, John T. (2004): *Evaluating Collaborative Filtering Recommender Systems.*
   URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.5270&rep=rep1&type=pdf`, zuletzt gesichtet am 05.01.2014.
6. McNee, Sean M.; Riedl, John; Konstan, Joseph A. (2006): *Being Accurate is Not Enough: How Accuracy Metrics have hurt Recommender Systems.*
   URL: `http://files.grouplens.org/papers/mcnee-chi06-acc.pdf`, zuletzt gesichtet am 05.01.2014.
7. McNee, Sean M.; Lam, Shyong K.; Guetzlaff, Catherine; Konstan, Joseph A.; Riedl, John (2003): *Confidence Displays and Training in Recommender Systems.*
   URL: `http://files.grouplens.org/papers/mcnee-interact2003.pdf`, zuletzt gesichtet am 05.01.2014.
8. Manos Papagelis, Manos; Rousidis, Ioannis; Plexousakis, Dimitris; Theoharopoulos, Elias (2005): *Incremental Collaborative Filtering for Highly-Scalable Recommendation Algorithms.*
   URL: `https://www.ics.forth.gr/isl/publications/paperlink/LNCS_Formatted_ISMIS-05_34880553.pdf`, zuletzt gesichtet am 18.02.2014.
9. Ricci, Francesco: *Recommender System - Database and Information Systems.*
   URL: `http://www.inf.unibz.it/~ricci/Slides/IntroductionToRecommenders.pdf`, zuletzt gesichtet am 06.01.2014.
10. Schweiger, Wolfgang (2007): *Theorien der Mediennutzung. Eine Einführung.* VS Verlag für Sozialwissenschaften: Wiesbaden.
11. Shani, Guy; Gunawardana, Asela (2011): *Evaluating Recommendation Systems.*
    URL: `http://research.microsoft.com/pubs/115396/evaluationmetrics.tr.pdf`, zuletzt gesichtet am 05.01.2014.
12. Vermorel, Joann'es; Mohri, Mehryar (2005): *Multi-armed Bandit Algorithms and Empirical Evaluation.*
    URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.4518&rep=rep1&type=pdf`, zuletzt gesichtet am 18.02.2014.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen wörtlich oder sinngemäß übernommenen Gedanken sind als solche gekennzeichnet. Diese Hausarbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.