

GENAUIGKEIT ODER VIELFALT
EINE EMPIRISCHE ANALYSE ANHAND EINES MUSIK
EMPFEHLUNGSSYSTEMS

BACHELORARBEIT
ZUM ERWERB DES AKADEMISCHEN GRADES BACHELOR OF SCIENCE

VON

PIERRE AHRENDT
MATRIKELNUMMER: 4129610

16. JUNI 2016

FREIE UNIVERSITÄT BERLIN
INSTITUT FÜR MATHEMATIK UND INFORMATIK
ARBEITSGRUPPE CORPORATE SEMANTIC WEB

GUTACHTER:
MARKO HARASIC
PROF. DR. ADRIAN PASCHKE

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen und Hilfsmittel wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit hat keiner anderen Prüfungsbehörde vorgelegen. Mir ist bekannt: Bei Verwendung von Inhalten aus dem Internet habe ich diese zu kennzeichnen und Ausdrücke davon mit Datum sowie der Internet-Adresse (URL) als Anhang der Bachelorarbeit beizufügen.

Ort, Datum und Unterschrift

Zusammenfassung

Personalisierte Empfehlungssysteme für Musik sind in den letzten Jahren immer bedeutender geworden, da ein großer Teil von Musikstücken digital konsumiert werden und in immer größerer Stückzahl verfügbar sind. Durch diese Masse an Möglichkeiten ist es notwendig geworden, den Benutzern Software und Werkzeuge zur Seite zu stellen, um das Durchsuchen von großen Bibliotheken praktisch durchführbar zu machen. Die meisten Arbeiten zu Empfehlungssystemen haben sich dabei auf die Genauigkeit der Empfehlung konzentriert. In letzter Zeit hat sich jedoch abgezeichnet, dass in praxistauglichen Systemen die Verschiedenartigkeit (Diversity) für die Benutzer eine wichtige Fähigkeit von Empfehlungssystemen darstellt. Im Rahmen der folgenden Arbeit sollen verschiedene Verfahren skizziert und evaluiert werden, um die Vielfalt in personalisierten Musikempfehlungssystemen zu steigern, ohne hierbei die sich diametral gegensätzlich verhaltenden Eigenschaft der Genauigkeit zu stark zu reduzieren.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.2	Fragestellung	3
1.3	Gliederung	3
2	Empfehlungssysteme	4
2.1	Kollaboratives Filtern	5
2.2	Inhaltsbasierten Systeme	9
2.3	Hybride Systeme	10
2.4	Implizite und Explizite Bewertungen	12
2.5	Musikempfehlungen	13
3	Diversität	14
3.1	Bedeutung und Definitionen	15
3.2	Strategien	17
3.3	Metriken	20
4	Methodik und Material	25
4.1	Datenerhebung	25
4.2	Implementierung	28
4.3	Validierung	29
5	Ergebnisse	30
5.1	Evaluationsmetriken	30
5.2	Evaluation	32
5.3	Diskussion	41
6	Zusammenfassung und Ausblick	43
6.1	Zusammenfassung	43
6.2	Ausblick	43

1 Einleitung

Mit dem rasanten Wachstum des Internets entwickelte sich eine bedeutende Quelle an multimedialen Daten, wie etwa Videos, Büchern oder Musik. Durch die verbesserte Verfügbarkeit mobiler Abspielgeräte, neuartiger Musikangebote und Internetdienste für Musik hat sich auch das Konsumverhalten der Hörer verändert. Aufgrund der wachsenden Masse an digitalen Daten entsteht die Möglichkeit, dass Informationen zu jeder Tages- und Nachtzeit für die meisten Menschen in westlichen Industrienationen an jedem Ort zur Verfügung stehen. Menschen betrachten das Musikhören als einen wichtigen Bestandteil ihres Lebens. Für Viele ist dies eine alltägliche Tätigkeit und wird bei verschiedenen Aktivitäten durchgeführt. Während die Menge und Verfügbarkeit neuer Musik stetig gewachsen ist, haben die Möglichkeiten traditioneller Wege zum Finden von Musik abgenommen. Vor einigen Jahren war der Radio DJ mit seiner Auswahl von Musikstücken oder der Plattenladen von nebenan eine sichere Quelle, um neue oder interessante Musik zu entdecken.

Heutzutage wurde diese weitestgehend von vorgefertigten Wiedergabelisten der Musikindustrie für Radiosendungen oder von großen Einzelhändlern mit einer beschränkten Auswahl ersetzt [43]. Viele Benutzer haben eine Vielzahl von Musikstücken in digitaler Form vorliegen, die sie bisher noch nicht oder nur sehr selten gehört haben. Es besteht das Problem die vielen Millionen Titel zu sortieren und zu verwalten. Im Bereich *Music Information Retrieval* wurden verschiedene Verfahren entwickelt, die Personen dabei helfen sollen, diese Probleme zu lösen. Erwähnenswert ist hier die automatische Genre-Klassifizierung [38] oder die Identifizierung des Künstlers eines Musikstücks anhand des Audiosignals [25]. Es scheint angebracht, dem Benutzern effiziente Werkzeuge an die Hand zu geben, um eine personalisierte Ordnung ihrer Sammlung zu ermöglichen. Außerdem würden Verfahren zum Auffinden unbeachteter Liebhaberstücke des Benutzers den Wert der Musiksammlung deutlich steigern.

Diese und weitere Ziele verfolgen personalisierte Empfehlungssysteme (im weiteren auch *Recommender Systems*). Im alltäglichen Leben sind Menschen ständig auf Ratschläge anderer Menschen angewiesen. Seien es nun Empfehlungen in natürlicher Sprache bei einem Verkaufsgespräch, Empfehlungsschreiben in einem Betrieb, der Reiseführer zum Erkunden einer Stadt oder die Einordnung von Ereignissen auf der Welt mit Hilfe von Nachrichten. Empfehlungssysteme unterstützen und erweitern den natürlichen sozialen Prozess, Menschen dabei zu helfen die Fülle der verfügbaren Objekte zu durchsuchen, um die nützlichsten Informationen zu erhalten [37].

Der Bereich der *Music Recommender Systeme* beschäftigt sich damit, Personen dabei behilflich zu sein Musik oder Künstler zu entdecken, die ihren Musikgeschmack widerspiegeln. Ein gutes Musik Empfehlungssystem sollte automatisch die präferierte

Musik des Benutzers erkennen und daraus neue Vorschläge zu erstellen, wie etwa eine generierte Wiedergabeliste zum Abspielen von Musik. Da jeder Hörer einen anderen Musikgeschmack hat, eignen sich personalisierte Empfehlungssysteme gut für den Musikbereich. Jedoch leiden Empfehlungssysteme mitunter an dem Problem, dass dem Benutzer nur Musiktitel von sehr berühmten Künstlern empfohlen werden. Das liegt einerseits daran, wie die Qualität der Vorschläge von Empfehlungssystemen bewertet wird, und andererseits daran, dass Empfehlungen oft auf die Bewertungen von anderen Benutzern zur Erstellung der Vorschläge zurückgreifen. Und so Künstler mit einer hohen Bekanntheit und vielen Bewertungen häufig vorgeschlagen werden.

1.1 Motivation

Die Erkenntnis, dass nicht nur die Genauigkeit bei Empfehlungssystemen von entscheidender Bedeutung für die Zufriedenheit der Benutzer ist, entstand bereits zum Anfang des Jahrtausends in der Forschung [17, 36]. Die Bewertung eines Empfehlungssystems anhand der Vorhersagegenauigkeit ist noch immer die am häufigsten anzutreffende Methode der Evaluierung neuer Verfahren.

Die Fragen, die sich stellen, sind zum einen, wie viel Mehrwert ein System für einen Benutzer hat, das ihm genau die Musiktitel vorschlägt, die ihm bereits bekannt sind und als gut befunden wurden. Zum anderen welche weiteren Eigenschaften zur Steigerung der Zufriedenheit der Benutzer eine Rolle spielen.

Hierbei wird häufig die Vielfalt der vorgeschlagenen Objekte erwähnt [2]. Zum Beispiel sollten dem Nutzer genre-übergreifende Künstler vorgeschlagen werden und nicht nur die ähnlichsten und populärsten Künstler seines bevorzugten Genres, die dem Benutzer mitunter zu einem großen Teil schon bekannt sind. Der Aspekt der Vielfalt der Vorschläge wird ein immer wichtigerer Bestandteil bei der Bewertung von Empfehlungssystemen und ist mittlerweile neben der Vorhersagegenauigkeit, immer häufiger in der Literatur anzutreffen [26].

Neben der Vielfalt wurde auch eine Vielzahl anderer Eigenschaften von Empfehlungssystemen untersucht, um die Zufriedenheit der Benutzer mit den Vorschlägen zu steigern. Die Nützlichkeit der Vorschläge ist dabei von besonderer Bedeutung, damit Benutzer zufrieden mit einem Empfehlungssystem sind.

Die Nützlichkeit der Systeme kann sich aber von Einsatzgebiet zu Einsatzgebiet unterscheiden oder von den Intentionen des Benutzers abhängig sein [17]. Es scheint also von Vorteil zu sein, wenn die Empfehlungen für den Benutzer eine gewisse Vielfalt widerspiegeln. Seitdem wurden viele neue Ansätze erforscht und neue Verfahren entwickelt, um Empfehlungen vielfältiger zu gestalten. Dennoch benötigen auch aktuelle Empfehlungssysteme immer weitere Verbesserungen, damit die Systeme unter realen Bedingungen zum Einsatz kommen können [4].

1.2 Fragestellung

Das Hauptziel der vorliegenden Arbeit ist die Untersuchung der Auswirkung von Vielfalt der Vorschläge in einem Musikempfehlungssystem. Hierfür werden verschiedene Verfahren zur Steigerung der Vielfalt in Empfehlungssystemen vorgeschlagen und mit quantitativen Metriken analysiert. Ein traditionelles kollaboratives Empfehlungssystem soll als Referenzsystem implementiert werden, um die Vergleichbarkeit herzustellen. Dabei steht die Frage im Mittelpunkt, wie die Vielfalt in Empfehlungssystemen gesteigert werden kann ohne die Vorhersagegenauigkeit der Bewertungen zu stark zu benachteiligen. Zusätzlich werden neue Techniken und Metriken der präsentierten Eigenschaften vorgestellt, um eine Evaluation zu ermöglichen. Die Evaluation der Verfahren wird anhand eines Datensatzes durchgeführt, der Hörereignisse von Benutzern enthält. Dementsprechend müssen die impliziten Informationen in Bewertungsstruktur anhand der Häufigkeit umgesetzt werden. Der Datensatz wird dann anhand der Benutzer in fünf Teile partitioniert, so dass eine Kreuzvalidierung durchgeführt werden kann.

1.3 Gliederung

In Kapitel 2 werden die klassischen Verfahren für Empfehlungssysteme vorgestellt. Das kollaborative Filtern basiert allein auf dem Bewertungsverhalten der Benutzer für Objekte, wohingegen das inhaltsbasierte Filtern sich darüber hinaus auf die Beschreibung der Objekte in dem System bezieht. Daneben existieren Verfahren, die die klassischen Ansätze kombinieren, um im besten Fall nur die Stärken der einzelnen Systeme wieder zu verwenden und die Nachteile der einzelnen Verfahren zu eliminieren. In Kapitel 3 wird auf die drei wichtigsten Eigenschaften zur Steigerung der Vielfalt in der Liste der Empfehlungen eingegangen und Verfahren präsentiert, um diese in Empfehlungssysteme einfließen zu lassen. Vorgestellt werden die Eigenschaften Diversity, Novelty und Serendipity.

In Kapitel 4 wird ein kollaboratives System implementiert, das als Referenz für die in Kapitel 3 vorgestellten Ansätze zur Steigerung der Vielfalt verwendet wird. Des Weiteren wird der Datensatz, der zur Evaluation der Verfahren verwendet wird, vorgestellt und die Umwandlung von impliziten Informationen in explizite Bewertungen wird beschrieben. Kapitel 5 umfasst die Evaluation der zuvor beschriebenen Verfahren. Zur Bestimmung der Qualität kommen traditionellen Metriken zur Evaluation von Empfehlungssystem zum Einsatz. Abschließend werden in Kapitel 6 die Ergebnisse der Arbeit diskutiert, eine Zusammenfassung der Arbeit präsentiert und ein Ausblick auf mögliche Erweiterungen gegeben.

2 Empfehlungssysteme

In diesem Kapitel umfasst eine Übersicht der klassischen Verfahren für Empfehlungssystemen. Die Ursprünge der Empfehlungssysteme liegen in den Forschungsgebieten des Information Retrievals, der Kognitionswissenschaft und des maschinellen Lernens [4]. Mitte der neunziger Jahre entstand ein eigenständiger Forschungsbereich mit der Veröffentlichung von ersten Publikationen über das kollaborative Filtern der Empfehlungssysteme. Diese konzentrierten sich auf die Lösung des Empfehlungsproblems mit Hilfe der expliziten Bewertung von Objekte durch Benutzer [4]. Die einfachste Form des Empfehlungsproblems lässt sich beschreiben als das Abschätzen der Bewertung eines Objekts, das dem Benutzer bis dato unbekannt ist oder das er noch nicht bewertet hat. Um die Bewertung eines Objektes zu berechnen, wird das Bewertungsverhalten eines Benutzers aus der Vergangenheit herangezogen. In Empfehlungssystemen wird die Nützlichkeit eines Objekts für einen Benutzer häufig über eine Bewertung auf einem vorher festgelegten Intervall repräsentiert, die das Interesse eines Benutzers an einem Objekt darstellt [4]. Durch die Vorhersage zuvor unbekannter Objekte, kann das System dem Benutzer Empfehlungen geben, indem die Objekte mit der größten geschätzten Bewertung vorgeschlagen werden [4]. Eine allgemeine Formalisierung des Empfehlungsproblems für klassische Empfehlungssysteme lässt sich definieren als eine Funktion f_u

$$f_u : U \times I \implies R \quad (2.1)$$

die die Nützlichkeit eines Objektes $i \in I$ für einen Benutzer $u \in U$ berechnet. Dabei ist I die Menge der Objekte in dem System, die empfohlen werden können. U ist die Menge der Benutzer und R eine geordnete Menge mit nicht negative Werten zur Beschreibung der Bewertung. Um für einen Benutzer $u \in U$ ein Objekt vorzuschlagen, muss das Empfehlungssystem das Objekt $i_R \in I$ finden, das die Nützlichkeitsfunktion für den Benutzer u maximiert:

$$\forall u \in U, \quad i_R = \arg \max_{i \in I} f(u, i) \quad (2.2)$$

In produktiven Empfehlungssystemen wird für gewöhnlich eine geordnete Menge mit N Objekten (auch Empfehlungsliste) erzeugt, die dem Benutzer vorgeschlagen wird. Die Ordnung der Vorschläge basiert auf die berechnete Nützlichkeit $f_u(u, i)$. Jeder Benutzer $u \in U$ kann über Profile definiert werden, die eine Beschreibung der Benutzer enthalten. Im einfachsten Fall besteht das Profil nur aus einer ID zur Identifikation des Benutzers. Es können jedoch auch persönliche Daten, wie etwa Alter, Geschlecht oder Wohnort zur Beschreibung der Benutzer einbezogen werden.

Auf eine ähnliche Art und Weise werden die Objekte $i \in I$ anhand von Merkmalen beschrieben, z.B. bei Musikstücken der Name des Künstlers, der Titel oder das Genre. Empfehlungssysteme werden anhand ihrer Ausprägung der Nützlichkeitsfunktion klassifiziert. Die am häufigsten in produktiven Systemen zum Einsatz kommenden Verfahren sind das kollaborative Filtern und die inhaltsbasierten Techniken. Darüber hinaus werden hybride Systeme, die aus einer Kombination der traditionellen Verfahren bestehen, als eigenständige Art von Empfehlungssystemen angesehen. Dieses Kapitel stellt die grundlegenden Arten von Empfehlungssystemen vor und bespricht die Vor- und Nachteile dieser.

2.1 Kollaboratives Filtern

Der Begriff „kollaboratives Filtern“ (*collaborative filtering*) wurde als erstes im Zusammenhang mit dem kollaborative Email-Filter-System *Tapestry* erwähnt [5, 15]. Die Idee hinter dem kollaborativen Filtern beruht auf der Annahme, dass Benutzer, die unterschiedliche Objekte ähnlich bewerten, die selben Interessen haben. Anhand von vergangenen Bewertungsverhalten erstellt das System neue Empfehlungen. Das erste Empfehlungssystem, welches vollständig auf automatisierte Vorschläge auf Grundlage von Interessen der Benutzer basiert, war das *GroupLens* System, das Benutzern personalisierte Vorschläge anhand der ähnlichsten Benutzer und ihrer Interessen, erstellt [31]. Das kollaborative Filtern gilt als die am weitesten verbreitete Technik, wenn es darum geht ein Empfehlungssystem zu entwickeln [32]. Der Grund hierfür liegt darin, dass reine kollaborative Empfehlungssysteme kein Wissen über das eingesetzte Wissensgebiet benötigen, um Empfehlungen zu erstellen, denn die Erstellung der Empfehlungen basiert ausschließlich auf den Bewertungsstrukturen der Objekte oder dem Bewertungsverhalten der Benutzer. Dazu werden für einen aktiven Benutzer die k ähnlichsten Benutzer in dem System gesucht, um dann die Objekte anhand der Bewertungen dieser k Benutzer zu empfehlen. Des weiteren hat der Ansatz die Fähigkeit Objekte aus Nischen zu empfehlen, die außerhalb der unmittelbaren Präferenzen der Benutzer liegen [8]. Kollaborative Systeme werden grundsätzlich in memory-based und dem model-based Ansatz unterteilt [7]:

memory-based Die Vorhersage einer Bewertung für ein Objekt durch einen Benutzer wird anhand der Gesamtheit aller vorherigen Bewertung der anderen Benutzer vorhergesagt.

model-based Mithilfe stochastischer Verfahren und eines Trainingsdatensatzes wird versucht Muster in den Bewertungen der verschiedenen Benutzer zu finden und daraus ein Modell zu lernen. Das gelernte Modell wird dann zum Vorschlagen von neuen Objekten verwendet.

Im folgenden wird ausschließlich auf den memory-based Ansatz eingegangen, da die Nachbarschaftsverfahren, für mehr Vielfalt bei den Empfehlungen sorgen [13, 20].

Die meisten kollaborativen Systeme basieren auf der Grundlage eines Ähnlichkeitsmaßes zwischen Benutzern oder Objekten. In den folgenden Abschnitten der Arbeit werden die beiden grundlegenden Techniken für das kollaborative Filtern vorgestellt. Die *item-based Neighborhood* basiert auf der Ähnlichkeit von Objekten, wohingegen die *user-based Neighborhood* ein Ähnlichkeitsmaß Basis der Benutzer berechnet. Beide Verfahren erstellen eine Matrix M der Dimension $m \times n$ mit m Benutzern und n Objekten. Die Zelle (j, k) der Matrix M repräsentiert die Bewertung $r_{j,k}$ eines Benutzers u_j für ein Objekt u_k . Die Zeilen der Matrix M repräsentieren die Profile der Benutzer und die Spalten die Objektprofile.

2.1.1 Item-based Neighborhood

Das *item-based Neighborhood* Verfahren verwendet die Ähnlichkeit von Objekten und Bewertungsmustern der Benutzer, um Empfehlungen zu erstellen. Wenn zwei verschiedene Objekte von denselben Benutzern gut oder schlecht bewertet wurden, werden die Objekte als gleich angesehen, unter der Annahme, dass die Benutzer dieselben Präferenzen für die gleichen Objekte haben. Diese Methode betrachtet die Menge von bewerteten Objekten eines Benutzers, berechnet daraus die Nützlichkeit $f_u(u, i)$ für ein Zielobjekt und entscheidet, ob es dem Benutzer vorgeschlagen werden soll oder nicht. In Abbildung 2.1 ist eine User \times Objekt Matrix zu sehen, in der die Benutzer u_1 und u_2 zwei Objekte i_j und i_k bewertet haben.

	i_1	i_2	...	i_j	...	i_k	...	i_m
u_1				R		R		
u_2				R		R		
\vdots				\vdots		\vdots		
u_i				R		\emptyset		
\vdots				\vdots		\vdots		
u_n				R		R		

Abbildung 2.1: Modell für item-based Neighborhood mit User \times Items Matrix

Um für einen Benutzer Empfehlungen zu erstellen, muss zunächst die Ähnlichkeit zwischen den Objekte berechnet werden. Zur Berechnung der Ähnlichkeit, kommt

häufig die Kosinusähnlichkeit oder Pearson Korrelation zum Einsatz. Formel 2.3 zeigt, wie die Kosinusähnlichkeit für zwei Objekte berechnet werden kann.

$$\text{sim}(i, j) = \cos\theta = \frac{i \cdot j}{\|i\| \cdot \|j\|} = \frac{\sum_{u \in U} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U} r_{u,i}^2} \sqrt{\sum_{u \in U} r_{u,j}^2}} \quad (2.3)$$

Der Kosinus zweier Vektoren bestimmt sich aus dem Standardskalarprodukt mit $\mathbf{i} \cdot \mathbf{j} = \|\mathbf{i}\| \|\mathbf{j}\| \cos\theta$ und daraus lässt sich dann Kosinusähnlichkeit [33] ableiten, wobei $r_{u,i}$ die Bewertung des Users u für das Objekt i sei. Auf die Pearson-Korrelation wird in Abschnitt 2.1.2 näher eingegangen.

Nachdem es möglich ist, die Ähnlichkeit zwischen Objekten zu berechnen, wird anschließend die Bewertung des Objektes i für einen Benutzer u vorhergesagt, indem die k nächsten Nachbarn für die Vorhersage von Objekt i hinzugezogen werden.

Dabei wird unter einem „nahen“ Nachbarn genau die Objekte verstanden, die eine hohe Ähnlichkeit zu Objekten in dem Profil des Benutzers haben. Gleichung 2.4 wird zeigt, wie die Vorhersage der Bewertung für ein Objekt i und einem Benutzer u berechnet wird. Die Vorhersage $r_{u,i}$ ist dann definiert als:

$$r_{u,i} = \frac{\sum_{j \in N_k} \text{sim}(i, j) \cdot r_{u,j}}{\sum_{j \in N_k} \text{sim}(i, j)} \quad (2.4)$$

Hierbei enthält die Menge N_k die k ähnlichsten Objekte zu i , die der Benutzer u bewertet hat.

2.1.2 User-based Neighborhood

Die Grundidee der *User-based Neighborhood* ist die direkte Umsetzung der Idee kollaborative Empfehlungssysteme. Das Ziel ist das Auffinden von Benutzern mit ähnlichem Bewertungsverhalten die dem des aktuellen Benutzers ähneln und verwenden deren Bewertungen auf anderen Objekten zur Vorhersage der Bewertung, die der aktuelle Benutzer noch nicht bewertet hat [34]. Die Vorhersage einer Bewertung $r_{u,i}$ kann im Gegensatz zum Verfahren der item-based Neighborhood berechnet werden, indem die Nutzer berücksichtigt werden, die dem Benutzer u am ähnlichsten sind. In Abbildung 2.2 ist die (Benutzer \times Objekt) Matrix dargestellt, die verwendet wird, um die Ähnlichkeit zwischen Benutzern anhand gemeinsam bewerteter Objekte zu berechnen. Um die Ähnlichkeit zwischen zwei Benutzer u und v zu berechnen, ist eines der verbreitetsten Verfahren die Berechnung der Pearson Korrelation, die in der Gleichung 2.5 dargestellt ist.

$$\text{sim}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2.5)$$

Wenn unter Verwendung der Pearson-Korrelation die k nächsten Nachbarn gefunden wurden, lässt sich die Bewertung $r_{u,i}$ mit Hilfe der Summe des gewichteten

	i_1	i_2	...	i_j	...	i_k	...	i_m
u_1				R		R		
u_2				R		R		
\vdots				\vdots		\vdots		
u_i				R		R		
\vdots				\vdots		\vdots		
u_n				R		R		

Abbildung 2.2: User \times Objekt Matrix für User-based Neighborhood

Durchschnitts der Bewertungen in Gleichung 2.6, über die k nächsten Nachbarn, berechnen.

$$r_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_k(u)} \text{sim}(u,v)(r_{v,i} \bar{r}_v)}{\sum_{v \in N_k(u)} \text{sim}(u,v)} \quad (2.6)$$

Nachteile

New User Problem Neue Benutzer müssen erst eine Anzahl von Objekten bewertet haben müssen, damit das System anhand ihrer Präferenzen Empfehlungen erstellen kann.

New Item Problem Wenn neue Objekte in das System aufgenommen werden, können diese nicht empfohlen werden, da noch keine Bewertungen von Benutzern existieren. Es muss also erst bewertet werden, damit es bei der Empfehlung berücksichtigt werden kann.

Sparsity Mit einer großen Anzahl von Benutzern und Objekten kommt eine geringe Abdeckung von Bewertungen der Objekte von Benutzern einher. Es ist nicht ungewöhnlich, dass die Dichte in der Matrix der Bewertungen weniger als 1% beträgt [43].

Bisher wurde gezeigt, dass kollaborative Systeme nur die Bewertungen für Objekte verwenden, um Vorschläge zu erstellen. Der große Vorteil dabei ist, dass die schwierige und kostenintensive Aufgabe detaillierte Informationen zu Objekten zu sammeln, entfällt.

2.2 Inhaltsbasierten Systeme

Im Gegensatz zu kollaborativen Systemen beruht der inhaltsbasierte Ansatz zur Erstellung von Empfehlungen unter Verwendung der Beschreibung der Objekte und der Bewertung des aktiven Benutzers. Die Grundidee hierbei ist, dass dem Benutzer neue Objekte empfohlen werden, deren Merkmale mit den Merkmalen der von ihm bewerteten Objekte korrelieren.

Die inhaltsbasierten Empfehlungssysteme (content-based recommenders) haben ihren Ursprung in den Gebieten des Information-Retrievals und Filtering. Die anfänglichen Systeme konzentrierten sich hierbei auf Text und versuchten mit Techniken aus dem Gebiet des *Information Retrieval* sinnvolle Informationen aus Dokumenten zu gewinnen [4, 43]. Erstmals erwähnt wurden inhaltsbasierte Empfehlungssysteme mit den Systemen *Newsweeder* [22] und *Syskill & Webert* [29], die Nachrichten für Benutzer filtern.

Bei den inhaltsbasierten Empfehlungssystemen werden den Benutzern Objekte vorgeschlagen, die ähnlich zu den von ihnen positiv bewerteten sind. Das heißt, dass das System anhand eines Ähnlichkeitsmaßes die Vergleichbarkeit von Objekten, anhand ihrer Eigenschaften, lernen. Das können bei Filmen die Beschreibung der Handlung oder darstellenden Schauspieler sein. Bei Musikstücken hingegen kann der Liedtext, sofern vorhanden, herangezogen werden oder aber auch das Audiosignal mit den verschiedenen Spektren als Charakteristik zum Einsatz kommen. Voraussetzung für die inhaltsbasierte Analyse ist, dass die Objekte Eigenschaften besitzen, mit denen sie beschrieben werden können. Das kann unter Umständen eine Herausforderung sein, wenn solche Informationen nicht existieren oder nur sehr schwer bzw. kostenintensiv zu beschaffen sind.

Damit inhaltsbasierte Empfehlungssysteme zum Einsatz kommen können, werden die entsprechenden Techniken benötigt, um die Objekte des Systems darzustellen und die Benutzerprofile abzubilden. Zusätzlich werden Verfahren benötigt, um schlussendlich die Benutzerprofile mit der Darstellung der Objekte zu vergleichen [24]. Der Empfehlungsprozess eines inhaltsbasierten Empfehlungssystems lässt sich in drei Schritte unterteilen.

Content Analyzer Als Vorverarbeitungsschritt, werden aus unstrukturierten Informationen (z.B. Texte) relevante Merkmale extrahiert. Hauptaufgabe ist also die Information so weiter zu verarbeiten, dass die Objekte für die nächsten Schritte in einer Form vorhanden sind, dass mit diesen weitergearbeitet werden kann. Mit diesem Schritt wird eine Darstellung der Objekte erzeugt, sodass das Ergebnis als Eingabe für die nächsten beiden Schritte verwendet werden kann.

Profile Learner In diesem Schritt werden die Benutzerprofile erstellt, indem die Präferenzen eines Benutzers anhand von getätigten Bewertungen unter Zuhilfenahme von Verfahren aus dem Bereich des maschinellen Lernens zum Einsatz

kommen. Diese Verfahren können ein Modell der Benutzerinteressen erstellen, durch Rückschlüsse auf das vergangene Bewertungsverhalten.

Filtering Component Relevante Objekte werden anhand des Benutzerprofils vorgeschlagen. Hierfür wird ein Ähnlichkeitsmaß auf dem Objektraum definiert. Für Texte bzw. Bag of Words (nach der Vorverarbeitung) wäre das Kosinus-Ähnlichkeit um die Dokumenten zu vergleichen. Dadurch lässt sich eine Reihenfolge der ähnlichsten Dokumente erstellen, von denen dann die k ähnlichsten Dokumente empfohlen werden.

Nachteile

Limited Content Analysis Da Empfehlungen auf Grundlage von Merkmalen der Objekte durch das System vorgeschlagen werden, müssen diese entweder von Hand zugewiesen werden oder automatisiert durch Verfahren erfasst werden. Für Textdokumente existieren solide Verfahren um diese Objekte zu erfassen [4]. In anderen Disziplinen gestaltet sich das Erfassen von Merkmalen schwieriger, zum Beispiel bei Bildern. Und auch das manuelle Zuordnen von Merkmalen kann unmöglich sein, wenn die Anzahl der Objekte zu groß ist, was bei Empfehlungssystem häufig der Fall ist. Dadurch das Objekte durch eine begrenzte Anzahl an Merkmalen beschrieben werden, können zwei unterschiedlich Objekte die selben Merkmale aufweisen und dadurch von dem System als gleich angesehen werden, obwohl sie eigentlich verschieden sind [4, 35].

Over-Specialization Beschreibt das Problem, dass den Benutzern nur Objekte vorgeschlagen werden, die nur sehr genau mit dem Profilen der Benutzer übereinstimmen. Dadurch beschränkt sich das System selbst an Vielfalt der vorzuschlagenden Objekte. Dadurch kann es passieren, dass die Benutzer eine beschränkte Sicht auf das System haben und nur Objekte empfohlen werden, die sie schon kennen.

New User Problem Wie auch bei dem kollaborativen Filtern müssen Benutzer zunächst ihre Präferenzen dem System mitteilen, damit es Objekte empfehlen kann.

Die beiden vorgestellten Varianten von Empfehlungssystemen haben sowohl Stärken, als auch Schwächen, die sich voneinander unterscheiden. Um die Stärken beider Systemarten zu kombinieren und die Schwächen im besten Falle zu beseitigen wurden die so genannten hybriden Empfehlungssystem entwickelt.

2.3 Hybride Systeme

Bei hybriden Empfehlungssystem handelt es sich um eine Kombination der vorher genannten Verfahren. Das System kombiniert mehrere Techniken und versucht mit den Stärken eines Empfehlungssystem die Schwächen des anderen Empfehlungssystem

zu beheben. Beispielsweise leidet das Kollaborative Filtern unter dem New Item Problem und kann keine Objekte empfehlen, die noch nicht von den Benutzern bewertet wurden. Häufig liegen jedoch Beschreibungen für die zu empfehlenden Objekte vor. Das heißt, mit einem Inhaltsbasierten Empfehlungssystem könnte dieses Problem behoben werden [4]. Es existieren verschiedene Ansätze, um die klassischen Verfahren zu kombinieren und daraus ein hybrides System zu erstellen. In der Literatur wird zwischen folgenden Methoden unterschieden [8].

Weighted Die in dem hybriden System verwendeten Ansätzen arbeiten unabhängig von einander und erstellen jeweils eine Liste von Empfehlungen. Die Vorhersage der verschiedenen Empfehlungssysteme werden dann anhand einer Gewichtung zu einem Endergebnis zusammengefügt. Die Gewichte durch Versuche zu bestimmen, ist die eigentliche Herausforderung an diesem Ansatz.

Switching Das System nutzt eine Bedingung oder ein Ereignis, um zwischen den verschiedenen Ansätzen zu wechseln. Der ausgewählte Ansatz übernimmt dann die Erstellung von Empfehlungen. Einfachste Methode wäre hierbei das zufällige Auswählen eines der Systeme.

Mixed Die verwendeten Systeme erstellen unabhängig voneinander ihre Empfehlungen. Die von allen Systemen vorgeschlagenen Objekte werden dann gemeinsam in das Endergebnis übernommen und präsentiert. Diese Methode ist dann von Vorteil, wenn eine große Anzahl von vorgeschlagenen Objekten gefordert ist.

Feature Combination Bei dieser Methode werden Merkmale aus den jeweilig anderen System abgeleitet und als zusätzliche Eingabe für das anderen Empfehlungssystem verwendet. Zum Beispiel können, anhand von kollaborativen Informationen Klassen, von Benutzern erkannt werden, die dann als Merkmal für ein inhaltsbasiertes Empfehlungssystem verwendet werden.

Cascade Im Gegensatz zu den vorher angesprochenen Methoden, verwendet diese Methode einen schrittweisen Ablauf. Zunächst wird mit einem der Systeme eine grobe Vorhersage getroffen. Das Ergebnis dieser Vorhersage wird dann in einem nächsten Schritt als Eingabe für eines der anderen Verfahren übergeben und verfeinert.

Feature Augmentation Bei dieser Technik werden die Objekte mit Hilfe des ersten Empfehlungssystems in Klassen geteilt oder bewertet. Diese Informationen über die Objekte als Ausgabe des ersten Empfehlens werden dann als neue Merkmale der Objekte verwendet und damit als Eingabe den nächsten Empfehlens verwendet.

Meta Level Bei dieser Technik wird das gelernte Modell als Eingabe für das andere System verwendet. Im Gegensatz zu Feature Augmentation wird nicht die Ausgabe des ersten Empfehlens als neues Merkmal verwendet, sondern es wird das Modell übergeben, das die neuen Merkmale erzeugt hat.

Durch hybride Systeme können einige der Nachteile der einzelnen Verfahren abgeschwächt werden. Im Idealfall werden nur die positiven Eigenschaften der Grundsysteme übernommen und die Nachteile vollständig eliminiert. Häufig wird als Grundlage ein kollaboratives System verwendet und mit anderen Verfahren kombiniert, um das New Item- oder New-User Problem zu lösen. Zum Beispiel kann ein weiteres inhaltsbasiertes System eingesetzt werden, um neue Objekte zu empfehlen, die noch nicht von Benutzern bewertet wurden [8].

2.4 Implizite und Explizite Bewertungen

Damit Empfehlungssysteme Vorschläge für Objekte erzeugen können, brauchen diese Informationen über den Benutzer. Aus diesen Informationen leitet ein Empfehlungssystem die Präferenzen ab. Im Grunde wird zwischen zwei Arten von Informationen über den Benutzer unterschieden, dem expliziten und impliziten Feedback.

2.4.1 Explizites Feedback

Bei expliziten Feedback muss der Benutzer aktiv Informationen über sich preisgeben. Diese Informationen können im Rahmen von Umfragen gesammelt werden oder personenbezogene Daten, wie zum Beispiel der Wohnort oder das Geburtsdatum sein. Eine andere wichtige Form im Bereich der Empfehlungssysteme ist die aktive Bewertung von Objekten von Benutzern. Als Beispiel hier wäre die Filmdatenbank IMDb ¹ genannt. Auf der Plattform können Benutzer Filme auf eine Skala von 1 bis 10 bewerten können [43].

2.4.2 Implizite Feedback

Im Gegensatz zu dem expliziten Feedback wird beim impliziten Feedback keine aktive Mitwirkung der Benutzer gefordert. Vielmehr beziehen sich die Rückschlüsse auf die Präferenzen der Benutzer auf deren Interaktion mit dem Empfehlungssystem, zum Beispiel wie oft hört ein Benutzer einen bestimmten Künstler [19]. Eine andere Methode ist das Verhalten der Benutzer mit den Objekten aufzuzeichnen. Bekanntes Beispiel hierbei ist der Musikdienst LastFM ², der die Anzahl der Wiedergaben eines Musikstücks der Benutzer der Plattform verwendet, um die beliebtesten Stücke zu bestimmen. [19, 18]

¹IMDb: <http://www.imdb.com/> Zugriff am 13. Mai 2016

²LastFM: <http://www.last.fm/> Zugriff am 17. Mai 2016

2.5 Musikempfehlungen

In der heutigen Zeitalter gibt es eine überwältigende Anzahl an Musikstücken aus denen Benutzer wählen können. Mit Anbietern von Musik wie *iTunes*, *Spotify* oder *Google Music* sind für Jeden und Jede ein riesiger Musikkatalog fortdauernd verfügbar. Es ist also notwendig Möglichkeiten auszusortieren, um den Präferenzen des Benutzers gerecht zu werden. In den letzten Jahren wurden einige Ansätze veröffentlichte die sich speziell mit Empfehlungen im Bereich der Musik beschäftigen. Die meisten Verfahren beruhen dabei auf den traditionellen Ansätzen des kollaborativen Filterns oder der inhaltsbasierten Empfehlungssysteme. [13].

Im Gegensatz zu den anderen Bereichen der Bücher und Filme, unterscheidet sich der Musikbereich dahingegen, dass die Informationen zu den Vorlieben der Benutzer weitestgehend implizit vorliegen. Das liegt daran, dass häufig das Hörverhalten der Benutzer aufgezeichnet wird. Jedes Musikstück kann mehrmals, oder sogar ständig, gehört werden. Ein weiterer Unterschied ist, dass Musik abhängig von der Umgebung konsumiert wird. Zum Beispiel wird bei der Lohnarbeit eher langsame Musik gehört, um sich zu konzentrieren. Beim Sport, wird hingegen eher schnellere Musik gewählt, um sich selbst zu motivieren. Das heißt die Musikempfehlung ist stark vom Kontext abhängig.

Eines der ersten Empfehlungssysteme, das über der Vorhersagegenauigkeit hinaus, die Vielfalt zur Erstellung von Vorschlägen berücksichtigt hat, war *Auralist* [41]. Dieses System zur Musikempfehlung beinhaltet sowohl Verfahren zur Erzeugung von Vielfalt in den Empfehlungslisten, als auch Evaluationsmetriken zur Bewertung der erzeugten Listen. In der anschließenden Benutzerstudie konnte gezeigt werden, dass die verwendeten Verfahren die Zufriedenheit der Personen mit den empfohlenen Künstlern gesteigert werden konnte. Die drei, in der Arbeit genannte Eigenschaften *Diversity*, *Novelty*, *Serendipity* werden unter dem Begriff der Vielfalt zusammengefasst [41].

3 Diversität

Im vorherigen Kapitel wurde die drei wichtigsten Vertreter für Empfehlungssysteme vorgestellt. Zur Beurteilung der Güte, Effektivität und Nützlichkeit dieser Systeme wurde ein Hauptaugenmerk darauf gelegt, die Interessen der Benutzer möglichst genau abzubilden. Ungefähr die Hälfte der gesamten Forschung zu Empfehlungssystemen beschränkte sich auf die Vorhersagegenauigkeit der Benutzerinteressen um ihre Verfahren zu evaluieren [11]. Doch bereits Anfang des Jahrtausends kam es bei der Entwicklung von Empfehlungssystemen zu der Einsicht, dass jenseits der Vorhersagegenauigkeit noch andere Eigenschaften für diese Systeme von Interesse sein können, die die Benutzer als nützlich wahrnehmen [17, 36]. Dieses Umdenken in der Beurteilung der Nützlichkeit von Empfehlungssystemen führte dazu, dass in den letzten Jahren vermehrt auch andere Eigenschaften bei der Evaluation der Systeme berücksichtigt wurden [17].

Die Empfehlungen die auf das Interesse eines Benutzer genau abgestimmt sind, sind nicht immer die Empfehlungen, die dem Benutzer am meisten Nutzen bringen [26]. Das kann unterschiedliche Gründe haben, z.B. wenn dem Benutzer immer wieder die selbe Künstlerin vorgeschlagen wird, weil er einige ihrer Lieder sehr hoch bewertet hat. Dieses Phänomen wird auch als *Filter Bubble* bezeichnet und beschreibt eine sich selbst verstärkenden Kreislauf der Meinungsbildung einer Person [28, 41].

Um diesen Kreislauf zu durchbrechen, ist es also notwendig Vorkehrungen dafür zu treffen, dass das Empfehlungssystem ausreichend vielfältige Vorschläge erstellen kann, damit der Benutzer neue Wissensgebiete erschließen kann. Offensichtliche Empfehlungen haben für Benutzer aber auch einen Mehrwert und steigern das Vertrauen der Benutzer in ein System, denn diese möchten auch Objekte empfohlen bekommen, die ihnen bereits bekannt sind [17, 12]. Somit ist es notwendig bei der Entwicklung von Empfehlungssystemen auf ein Gleichgewicht zwischen der Vorhersagegenauigkeit und der Vielfalt von vorgeschlagenen Objekten zu achten.

Um Vielfalt in den Vorschlägen von Empfehlungssystemen einzuführen, existieren verschiedene Herangehensweisen [11]. Häufig werden hierbei die Eigenschaften Diversity (eigentlich Vielfalt), Novelty und Serendipity genannt. Unter Diversity wird die Unterschiedlichkeit der Objekte in der Liste der Empfehlungen verstanden. Mit Novelty beschreibt man vorgeschlagene Objekte, die dem Benutzer bisher unbekannt waren. Die Serendipity bezeichnet Vorschläge, die für den Benutzer sowohl nützlich, als auch überraschend sind. Wenn ein Empfehlungssystem die Eigenschaft der Serendipity erfüllt, dann geht die Eigenschaft Novelty mit einher. Im folgenden Kapitel werden die Eigenschaften für Vielfalt definiert und verschiedene Strategien vorgestellt, um diese einzuführen. Darüber hinaus werden Techniken zur Evaluation vorgestellt, um diese Eigenschaften in Empfehlungssystemen zu messen.

3.1 Bedeutung und Definitionen

3.1.1 Diversity

Unter Vielfalt(oder Diversity) versteht man im Allgemeinen eine Fülle von verschiedener Ausprägungen von Objekten in einer Klasse. Im Kontext von Empfehlungssystem sind das die vorgeschlagenen. Objekten in der Empfehlungsliste eines Benutzers und deren Unterschiedlichkeit, die mit der Eigenschaft Diversity beschrieben werden.

Für Benutzer ist die Diversity von empfohlenen Objekten von Interesse, da sie zu einem Punkt gelangen können, an dem die Empfehlung der ähnlichsten Objekte ihnen nicht weiter hilft. Es entsteht eine sogenannte *Filterblase*, in der sich die Präferenzen des Benutzers selbstverstärkend bei seiner Suche auswirken können [28]. Ein Beispiel wäre, dass dem Benutzer immer nur die selben Künstler aus dem gleichen Genre vorgeschlagen werden. Um dieses Problem zu vermeiden, ist es notwendig bei Empfehlungen neuer Objekte auch Dinge vorzuschlagen, die nicht vollständig die Vorlieben des Benutzers entsprechen [14].

Die einfachste Methode, um Vielfalt in ein Empfehlungssystem einzuführen, ist zufällige Objekte vorzuschlagen. Dadurch entsteht jedoch ein anderes Problem, dass Benutzer nur dann Vertrauen in eine Empfehlungssystem haben, wenn ihnen ein Teil der vorgeschlagenen Objekte vorher bekannt sind, so dass komplexere Verfahren zum Einsatz kommen müssen [41, 26]. Die Steigerung der *Diversity* in einem Empfehlungssystem geht auch immer mit der Abwägung der Vorhersagegenauigkeit der Vorschläge einher, da sich diese beiden Eigenschaften oftmals gegensätzlich zueinander verhalten. Für gewöhnlich nimmt die Genauigkeit der Vorhersagen mit der Vielfalt der Vorschläge ab und muss dementsprechend bei der Entwicklung von neuen Strategien berücksichtigt werden.

Die Eigenschaft der *Diversity* in einem Empfehlungssystem lässt anhand verschiedenen Betrachtungsweisen charakterisieren. Am häufigsten ist der Fall anzutreffen, dass die Verschiedenartigkeit der Empfehlungen jedes einzelnen Benutzers getrennt voneinander betrachtet werden [42], also wie verschieden die Objekte in der Empfehlungsliste eines Benutzers sind. Darüber hinaus ist es möglich eine systemweite Diversity anhand der Empfehlungslisten aller Benutzer zu definieren, also wie unterschiedlich sind die Empfehlungslisten in den System für verschiedene Benutzer. Ebenfalls wurde die Veränderung der Empfehlungen über die Zeit in Betracht gezogen, um zu erkennen wie die Empfehlungslisten der Benutzer in dem System mit Zeit variieren [23].

3.1.2 Novelty

Die Eigenschaft der Novelty ein wichtiger Aspekt des Empfehlungsproblems und es ist weitgehend anerkannt, dass zu offensichtliche Empfehlungen die Zufriedenheit der Benutzer negativ beeinträchtigen könnten [17]. Offensichtliche Empfehlungen haben zwei grundlegende Nachteile: einerseits könnten Benutzer empfohlenen Objekte bereits kennen und andererseits braucht es kein Empfehlungssystem, um Objekte vorzuschlagen, die so populär sind, dass die Mehrheit der Benutzer schon gehört wurden.

Im Gegensatz zur *Diversity* wird die *Novelty* nicht auf einer Menge von Objekten definiert, sondern beschreibt den Zustand eines einzelnen Objektes. Ein Objekt ist neuartig (novel) für einen Benutzer, wenn er mit diesem noch keine Erfahrung in der Vergangenheit gemacht hat. Deshalb ist auch verständlich, dass in verschiedenartigen Menge von Objekten, diese im Bezug zueinander als neuartig bezeichnet werden können. Darüber hinaus tendiert ein System, das den Benutzern vor allem neuartige Objekte vorschlägt dazu, die globale Vielfalt (global diversity) in der Erfahrung der Benutzer über die Zeit zu realisieren [11]. Ist jedoch der Faktor der *Novelty* zu hoch bei dem Empfehlungen, könnten Benutzer von dem System abgeschreckt werden, da die Empfehlungen unbrauchbar für den Benutzer sein könnten.

Eine einfache Methode, um *Novelty* in Systemen zu erreichen, ist das Filtern der Objekte, die von Benutzern bereits bewertet wurden. Allerdings teilen die Benutzer dem System nicht jedes Objekt mit, mit dem sie jemals interagiert haben. Es reicht also nicht aus, nur diese Objekte aus der Liste der Empfehlungen zu entfernen. Auch die Eigenschaft der *Novelty* in einem Empfehlungssystem lässt sich anhand verschiedener Gesichtspunkte im Zusammenhang der Erfahrung der Benutzer definieren.

Die einfachste Möglichkeit ist hierbei die *Novelty* über das Vorhandensein von Neuartigkeit im Bezug zu auf die vergangenen Erfahrung eines Benutzers als boolescher Wert zu definieren. Ein Benutzer hat mit dem ihm vorgeschlagenen Objekt bereits innerhalb des Systems interagiert oder es ist ihm unbekannt und somit neu. Eine andere Herangehensweise betrachtet nicht den einzelnen Benutzer, sondern alle Benutzer des Systems. Bei der *Long Tail Novelty* wird die Frage gestellt, wie vielen Benutzern des Systems ist ein bestimmtes Objekt bereits bekannt, es kann also zwischen bekannten und unbekannten Objekte unterschieden werden. Des weiteren Alternative ist die *Novelty* in Abhängigkeit der Ähnlichkeit von Objekten, die einem Benutzer bereits bekannt sind, zu definieren.

Für die Evaluation der Novelty in Empfehlungssystemen können Benutzer in Studien befragt werden, ob sie die vorgeschlagenen Objekte bereits kennen. Aber auch in Offline Experimenten ist es möglich, die Novelty zu evaluieren. Hierfür wird der Datensatz so aufgeteilt, dass das die Empfehlungen nach einem bestimmten Zeitpunkt als Testdatensatz verwendet werden und überprüft, ob die Objekte in der Zukunft empfohlen werden, die ein Benutzer noch nicht gesehen und bewertet hatte [17].

3.1.3 Serendipity

Ein Empfehlungssystem erfüllt also die Eigenschaft der Serendipity, wenn die vorgeschlagenen Objekte sowohl unerwartet als auch ansprechend für den Benutzer sind. Die Serendipity berücksichtigt dabei neben der Neuartigkeit von Empfehlungen auch die Nützlichkeit und ist eine Erweiterung der Novelty [4].

Angenommen, dem Benutzer wird eine komplett zufällige Liste von Empfehlungen vorgeschlagen, dann sind die empfohlenen Objekte zwar neu, aber nicht hilfreich. Deswegen ist es notwendig für eine Ausgewogenheit zwischen Neuartigkeit und der Nützlichkeit der empfohlenen Objekte zu sorgen. Um ein Empfehlungssystem unerwartbar für die Benutzer zu machen, werden häufig Metadaten oder inhaltsbasierende Verfahren benötigt [17].

Der Grad der Unerwartbarkeit für die Benutzer wird normalerweise definiert, als die Anzahl der Vorschläge, die nicht von einem primitiven Empfehlungssystem empfohlen werden. Die Objekte in der Empfehlungsliste des Referenzsystems können dann mit der Liste des eigentlichen Systems verglichen werden, um daraus Schlüsse zu ziehen, welche Vorschläge überraschend für den Benutzer sind.

Das Entwerfen von Metriken zur Messung der Serendipity in einem Empfehlungssystem stellt sich als schwierige Aufgabe dar, da es bei dieser Eigenschaft darum geht, dass die vorgeschlagenen Objekte sowohl nützlich als auch überraschend für die Benutzer sind. Eine gute Metrik für Serendipität würde messen, wie sich die Interessen eines Benutzers anhand der Empfehlungen über die Zeit weiterentwickeln [17].

Nachdem die wichtigsten Eigenschaften für Vielfalt in Empfehlungssystemen vorgestellt wurden, behandelt der nächste Abschnitt Verfahren, die diese drei Eigenschaften in den Systemen einführen und steigern sollen.

3.2 Strategien

Im vorherigen Abschnitt wurden die Eigenschaften in Empfehlungssystem für abwechslungsreicher Empfehlungen charakterisiert. Im vorliegenden Teil der Arbeit werden nun Verfahren und Strategien zu beschreiben, um diese drei Eigenschaften in Empfehlungssystem einzuführen und zu steigern.

Hierfür wird im ersten Teil 3.2.1 ein iteratives Verfahren vorgestellt, dass versucht ein Gleichgewicht zwischen Vorhersagegenauigkeit und Diversity der Empfehlungen herzustellen. Anschließend wird im Abschnitt Benutzerprofil Clustering 3.2.2 eine Strategie vorgestellt, die die Benutzerprofile in Cluster einteilt und anhand der Gruppe von Objekten die Empfehlungslisten erstellt. Am Ende wird im Teil Aggregierte Diversity 3.2.3 ein Verfahren beschrieben, dass die Empfehlungslisten des Benutzers neu ordnet, um so die Vielfalt zu steigern. Die ursprüngliche Empfehlungsliste wird mithilfe eines benutzerbasierten kollaborativen Empfehlungssystems anhand der Bewertungen, der k nächsten Nachbarn erstellt.

3.2.1 Greedy Selection

Algorithm 1 Greedy Reranking Algorithm

```

1:  $R \leftarrow 0$ 
2: while  $|R| < k$  do
3:    $\hat{i} \leftarrow \arg \max_{i \in C \setminus R} g(U \cup i, \lambda)$ 
4:    $R \leftarrow R \cup \hat{i}$ 
5: end while

```

Der Algorithmus der *Greedy Selection* versucht sowohl die *Diversity* in den Empfehlungslisten zu steigern, als auch die Vorhersagegenauigkeit des zugrunde liegenden Systems beizubehalten. Bei diesem Ansatz werden die Empfehlungslisten anhand einer Metrik aus Relevanz und Vielfalt für die Objekte iterative aufgebaut. Sei R die Empfehlungsliste zum Zeitpunkt j . Im ersten Schritt wird das Objekt mit der höchsten Relevanz in die Empfehlungsliste des Benutzers aufgenommen. Im j -ten Schritt wird das Objekt hinzugefügt, dass eine Funktion aus Relevanz und *Diversity* maximiert [36].

$$g(R, \lambda) = (1 - \lambda) \sum_{i \in R} f_{rel}(i) + div(R) \quad (3.1)$$

Wobei mit $\lambda \in [0, 1]$ das Verhältnis zwischen der Relevanz der Objekte und der Diveristy in der Empfehlungsliste gesteuert werden kann. Ein kleiner Wert für λ bedeutet, dass die gewählten Objekte weniger diverse sind. Ist der Wert für λ groß, dann ist die Liste der Empfehlungen sehr viel vielfältiger. Im Information Retrieval wird dieser Ansatz zum Aufbau der Empfehlungsliste als *maximum marginal relevance* (kurz *MMR*) bezeichnet [10].

3.2.2 Benutzerprofil Clustering

Ein weiteres Verfahren um einen Anstieg der Vielfalt bei Empfehlungen zu erzeugen, ist die Clusteranalyse auf den bewerteten Objekte des aktiven Benutzers. Hierbei werden ähnliche Objekte in dem Profil des Benutzers in sogenannten Clustern gruppiert und die Zentren dieser Cluster werden, anstatt des gesamten Benutzerprofils, verwendet, um Empfehlungen für den Benutzer zu erzeugen. Bei der Berechnung der Cluster wird versucht, die Summe der Abstände zwischen den Objekte in einer Gruppe zu minimieren. Zur Berechnung der Cluster kommt der k-Means-Algorithmus zum Einsatz, der sich durch seine Einfachheit und Effizienz auszeichnet [40]. Als Abstandsmaß für die Clusterberechnung bieten sich die Cosinus Ähnlichkeit 2.3 an.

k-Means Algorithmus

Der k-Means-Algorithmus ist ein Verfahren zur Berechnung von Clustern in einer Menge von Datenpunkte. Dabei wird die Menge der Punkte in k Gruppen partitioniert. Die Anzahl k muss vor der Clusterberechnung fest gelegt werden und hängt von

den zugrundeliegenden Daten ab. Der Algorithmus kommt häufig zur Berechnung von Clustern zum Einsatz, da er nach nur wenigen Iterationen die Clusterzentren findet. Folgende Schritte werden benötigt, um die Clusterzentren mit dem k-Means Algorithmus zu berechnen:

Algorithm 2 Der Kmean Algorithmus

- 1: *Initialisieren Clusterzentren* c_1, c_2, \dots, c_k zufällig
 - 2: **repeat**
 - 3: **Zuordnung** der Punkte zu dem Cluster, dessen Zentrum am nächsten liegt
 - 4: **Aktualisierung** der Clusterzentren anhand der zugewiesenen Punkte
 - 5: **until** *Konvergenzkriterium erreicht*
-

Die Schritte *Zuordnung* und *Aktualisieren* werden solange wiederholt, bis ein zuvor definiertes Abbruchkriterium erreicht wurde, beispielsweise nach einer festgelegten Zahl von Iterationen, oder wenn sich die Cluster nicht mehr ändern. Das Ziel des Algorithmus ist die Summe der quadratischen Fehler (siehe 3.2) innerhalb eines Clusters C_i zu minimieren.

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist(c_{i,x})^2 \quad (3.2)$$

3.2.3 Aggregierte Diversity

Wie schon in Abschnitt 3.1.1 angesprochen, kann die Diversity nicht nur über die Liste der Empfehlungen betrachtet werden, sondern auch als Eigenschaft des Gesamtsystems aufgefasst werden, also wie vielfältig die Objekte in den Empfehlungslisten im Bezug zu allen Benutzern des Systems sind. Um die Vielfalt in den Empfehlungen zu steigern, kommen verschiedene Methoden zum Einsatz, die auf Grundlage eines klassischen Systems die Objekte in der Empfehlungsliste neu ordnen. Sei $rank_{CF}(i) = \frac{1}{\hat{r}(u,i)}$ das Ranking eines klassischen benutzerbasierten kollaborativen Systems, das die Objekte anhand ihrer vorhergesagten Bewertung aufsteigend sortiert und sei $rank_X(i)$ ein alternatives Kriterium zur Ordnung der Objekte in der Empfehlungsliste, dann lässt sich hiermit die Methode zur Neuordnung der Objekte definieren als:

$$rank_X(i, \tau) = \begin{cases} rank_X(i) & \hat{r}(u, i) \geq \tau \\ \alpha_u + rank_{CF}(u, i) & \hat{r}(u, i) < \tau \end{cases} \quad (3.3)$$

mit $\alpha_u = \max_{i: \hat{r}(u,i) \geq \tau} rank_X(i)$

Wobei der Schwellwert τ zum Einsatz kommt, um die Neuordnung der Objekte zu steuern. Bei der richtigen Wahl von τ kann die globale Diversity gesteigert werden, bei gleichbleibender Genauigkeit der Vorhersagen. Verschiedene Ranking Funktionen

wurden vorgestellt, um die globale Diversity zu steigern [3]. Im folgenden werden einige Ranking Funktionen formal dargestellt:

$$rank_{pop}(i) = |\{u \in U | i \in u\}| \quad \text{Inverse popularity} \quad (3.4a)$$

$$rank_{rev}(i) = \hat{r}(u, i) \quad \text{reverse rating} \quad (3.4b)$$

$$rank_{avg}(i) = avg_{u':i \in u} \hat{r}(u', i) \quad \text{average rating} \quad (3.4c)$$

$$rank_{var}(i) = Var_{u':i \in u}(\hat{r}(u', i)) \quad \text{item rating variance} \quad (3.4d)$$

$$(3.4e)$$

Die Rankingfunktion 3.4a ordnet die Objekte in der Empfehlungsliste anhand ihrer Popularität in dem System. Die Funktion 3.4b ordnet die Empfehlungen anhand der vorhergesagten Bewertung von klein nach groß. Wohingegen die Funktion 3.4c die Empfehlungen anhand des Durchschnitts aller Bewertungen sortiert. Und zuletzt noch die Funktion 3.4d, die die Empfehlungen nach der Varianz der Bewertungen der Nachbarn sortiert. Die Annahme hierbei ist, dass eine höhere Varianz auch mehr Vielfalt verursacht.

Im vorliegen Abschnitt wurde drei Verfahren zur Steigerung der Vielfalt in Empfehlungssystemen vorgestellt. Im nächsten Abschnitt sollen Verfahren zur Bewertung der Vielfalt von Empfehlungssystemen vorgestellt werden.

3.3 Metriken

Im ersten Abschnitt 3 wurden die drei Zieleigenschaften für Abwechslungsreichtum in Empfehlungssystemen beschrieben und wie diese im Kontext von Empfehlungssystem zu verstehen sind. Danach wurden in dem Teil 3.2 Verfahren beschrieben, um die Vielfalt in Empfehlungssystem zu steigern. Im folgenden Abschnitt werden diese Eigenschaften weiter formalisiert und Techniken zur Evaluierung in Empfehlungssystemen vorgestellt. Damit die Evaluation der Vielfalt in Empfehlungssystem gelingen kann, muss auf die besonderen Aspekte dieser Eigenschaften eingegangen werden. Zu diesem Zweck werden die Eigenschaften anhand ihrer Sichtweisen vergewärtigt und mögliche Verfahren zur Bewertung beschrieben. Zu Beginn wird ein systemweites Maß beschrieben, das den Anteil der im System vorgeschlagenen Objekte misst. Bei dem darauffolgenden Verfahren wird eine Möglichkeit präsentiert, die die Unterschiede der Objekte innerhalb der Empfehlungsliste misst. Zum Schluss werden jeweils eine Metrik zum Messen der *Novelty* und der *Serendipity* beschrieben.

Einige der Metriken die in diesem Abschnitt beschrieben werden, verwenden ein Ähnlichkeitsmaß $sim(i, j)$, um die Objekte i und j zu vergleichen. In der Arbeit wird hierfür die Kosinus-Ähnlichkeit 2.3 aus dem Abschnitt 2.1.1 genutzt.

3.3.1 Catalog Coverage

Bei der *Catalog Coverage* oder Katalog Abdeckung wird der Anteil der Objekte in einem System gemessen, die jemals einem Benutzer vorgeschlagen werden. Hiermit

wird ermittelt, welche der in dem System vorhandenen Objekte, überhaupt den Benutzer vorgeschlagen werden können. Hat die *Catalog Coverage* den Wert 1 kann das System alle Objekte vorschlagen. Dadurch ergibt sich die globale Diversity, denn die von dem System vorgeschlagenen Empfehlungslisten sind in Beziehung zueinander divers. In der Regel wird die *Catalog Coverage* zu einem bestimmten Zeitpunkt berechnet, z.B. indem für alle Benutzer in dem System die vereinigte Menge aller Top N Empfehlungen verwendet wird 3.5 [17, 14].

$$C_{cov} = \frac{|\bigcup_{u \in U} R_u|}{|I|} \quad (3.5)$$

Die *catalog coverage* kann ein erster Hinweis darauf sein, wie divers ein Empfehlungssystem ist. Die Diversity wird also nicht anhand eines einzigen Benutzer gemessen, sondern über die Gesamtheit der Benutzer in dem System [13].

3.3.2 Intra-List Similarity und Intra-List Distance

Berechnet die paarweise Distanz aller vorgeschlagenen Objekte in einer Empfehlungsliste. Dass heißt ein höherer Wert beschreibt weniger Diversity in der Liste der Empfehlungen [42]. R_u ist die Liste der Empfehlungen für den Benutzer u und $sim(i_j, i_k)$ ist ein zu definierendes Ähnlichkeitsmaß, das die Distanz zwischen zwei Objekten i und i misst.

$$ILS_u = \frac{\sum_{i_j \in R_u} \sum_{j \in R_u, i \neq j} sim(i, j)}{N(N-1)} \quad (3.6)$$

Aus dem Ähnlichkeitsmaß ILS_u lässt sich sodann eine Metrik ILD , zur Messung der Unterschiedlichkeit der Objekte in der Empfehlungsliste R_u , ableiten.

$$ILD_u = \frac{\sum_{i_j \in R_u} \sum_{j \in R_u, i \neq j} 1 - sim(i, j)}{N(N-1)} \quad (3.7)$$

Die Definition der Ähnlichkeitsmetrik $sim(i, j)$ für zwei verschiedene Objekte i und j ist abhängig von den Objekte, die empfohlen werden sollen.

3.3.3 Long Tail Novelty

Die Neuartigkeit eines Künstlers für einen Benutzer, kann als das Gegenteil von Popularität definiert werden. Ein Künstler oder ein Objekt gilt also als neu, wenn nur wenige Benutzer damit interagiert haben und somit von dessen Existenz wissen. Das bedeutet der Künstler ist weit im *Long Tail* der Popularitätsverteilung, wie in Abbildung 3.1 dargestellt. Die Popularität eines Künstler kann als Wahrscheinlichkeit formalisiert werden, also wie Wahrscheinlich ist es, dass ein zufälliger Benutzer einen bestimmten Künstler kennt [11]. Um eine absteigende Funktion für die Popularität von Objekte zu erhalten, bietet der negative Logarithmus eine ähnlichen Ansatz, wie die inverse Dokumentenhäufigkeit (IDF) aus dem *Information Retrieval* und

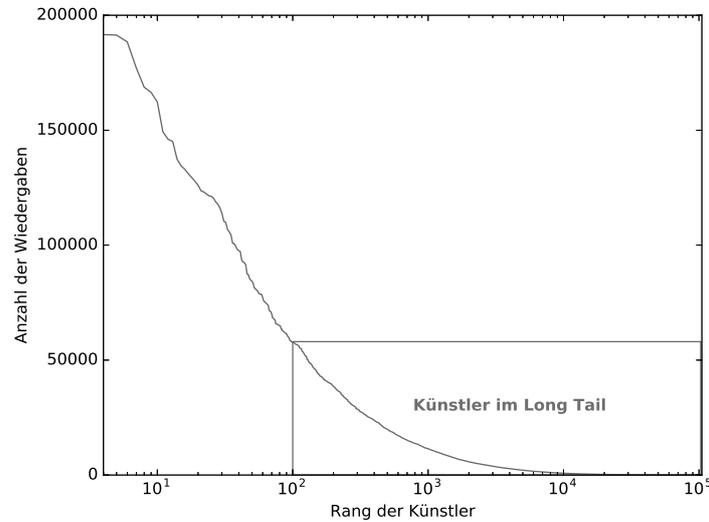


Abbildung 3.1: Longtail Popularität von Künstlern

wird als inverse Benutzerhäufigkeit $MIUF$ (siehe Gleichung 3.8) bezeichnet. Wobei $U_i = \{u \in U \mid r(u, i) \neq \emptyset\}$ die Menge der Benutzer ist, die mit einem bestimmten Objekt i interagiert haben.

$$MIUF = \frac{1}{|R_u|} \sum_{i \in R_u} \log_2 \frac{|U_i|}{|U|} \quad (3.8)$$

3.3.4 User-Specific Novelty

Die Long-tail Novelty aus der vorherigen Abschnitt ist ein Maß, dass die Novelty eines Objekts unabhängig von aktiven Benutzer misst. Es kann jedoch auch spezifischen Erfahrungen des Benutzers bei der Bewertung von Objekten, die ihm empfohlen werden sollen, zu berücksichtigen. Prinzipiell existieren zwei unterschiedliche Herangehensweisen, um die Neuartigkeit eines Objektes mit der Erfahrung eines Benutzers zu bewerten. Bei der sogenannten *Item-identity* wird das Wissen des Benutzers über ein Objekt als solches verwendet. Also hat der Benutzer das Objekt bereits in der Vergangenheit gesehen oder nicht. Bei der *Item-characteristic* wird gefragt, ob der Benutzer bereits mit Attributen des Objekt interagiert hat. Die item-identity ist also ein boolescher Wert, wohingegen das zweite Maß, als Wahrscheinlichkeit ausdrückbar ist [11]. Ein naiver Ansatz zu Berechnung der Item-identity wäre das Vorhandensein einer Interaktion von Benutzer und Objekt in dem System. Etwas weitergehend wäre die Überlegung bei wiederkehrenden Interaktionen, wie dem Abspielen von Musik, des Benutzers die Häufigkeiten als Indikator für dieses Maß zu

verwenden. Leider existieren hierfür keine Definitionen in der Literatur. Deshalb wird sich in dieser Arbeit auf die Charakteristik der Objekte beschränkt.

Für eine Definition der Novelty aus der Sicht der Benutzer, werden die Objekte in der Empfehlungsliste berücksichtigt und versucht zu ermitteln, ob der Benutzer mit den vorgeschlagenen Objekte innerhalb des Systems bereits mit diesen interagiert hat. Hierfür wird eine Distanzmaß auf den Objekten definiert. Die Formalisierung dieses Gedanken findet sich in der Gleichung 3.9, in der die durchschnittliche Distanz der empfohlenen Objekte mit den Objekten im Profil des Benutzers berechnet wird.

$$\text{UserNovelty} = \frac{1}{|R||I_u|} \sum_{i \in R} \sum_{j \in I_u} 1 - \text{sim}(i, j) \quad (3.9)$$

Die Menge $I_u = \{i \in I \mid r(u, i) \neq \emptyset\}$ ist hierbei die Menge aller Objekte i , die von einem Benutzer u bewertet wurden. Die Unterwartbarkeit bzw. Novelty eines Objektes wird auch als Grundlage zur Berechnung der Serendipity der Empfehlungen verwendet.

3.3.5 Serendipity

Wie bereits in 3.1.3 beschrieben, lässt sich die Eigenschaft der Serendipity als für den Benutzer unerwartbar und brauchbar definieren. Im vorherigen Abschnitt zu Novelty wurde eine Möglichkeit zur Definition von Unterwartbarkeit vorgestellt. Für die Eigenschaft der Serendipity kann die Unerwartbarkeit aber auch auf andere Weise definiert werden. Formal gesehen lässt sich die Unerwartbarkeit definieren, als Anteil zwischen dem Unterschied von erwarteten Vorschlägen und den tatsächlich gemachten Vorschlägen.

$$\text{Unexp} = \frac{|R \setminus EX|}{|R|} \quad (3.10)$$

Wobei die Menge EX die erwarteten Vorschläge und die Menge R der in Wirklichkeit vorgeschlagenen Objekte für den Benutzer sind. Die Menge der erwarteten Menge kann dann mithilfe eines primitiven Empfehlungssystems erzeugt werden [27].

Da nicht alle unerwarteten Vorschläge auch hilfreich für den Benutzer sind, muss auch die Nützlichkeit der Vorschläge berücksichtigt werden. Denn eine rein zufällige Liste an Vorschläge ist zwar unerwartet für den Benutzer, wird ihn aber nicht bei seiner Suche weiterhelfen.

RS_i wird definiert als ein Element in der Menge *unexpectedness*. Ist $u(RS_i) = 1$ ist der Vorschlag für den Benutzer nützlich. Ist hingegen $u(RS_i) = 0$, ist der Vorschlag für den Benutzer nicht zu gebrauchen. Ein Vorschlag wird als nützlich betrachtet, wenn die durchschnittliche Bewertung des vorgeschlagenen Objektes größer 3 ist. Dabei wird eine Skala von 0 bis 5 angenommen [2]. Wenn sowohl die Nützlichkeit, also auch Unerwartbarkeit von Vorschlägen ermittelt werden kann, dann lässt sich die Serendipity wie folgt definieren.

$$\text{serendipity} = \frac{|Unexp \cap Useful|}{|N|} \quad (3.11)$$

Wobei N die Anzahl der für den Benutzer vorgeschlagenen Elemente darstellt. In vorliegenden Kapitel wurde der Begriff der Vielfalt im Kontext zu Empfehlungssystem vorgestellt. Hierfür wurden die drei wichtigsten Eigenschaften Diversity, Novelty und Seredipity definiert und Verfahren beschrieben, damit diese innerhalb von Empfehlungssystem umgesetzt werden können. Zum Schluss wurden Möglichkeiten zur Evaluation dieser Eigenschaften vorgestellt. Im nachfolgenden Kapitel werden die vorliegenden Techniken an einem Datensatz empirisch untersucht.

4 Methodik und Material

4.1 Datenerhebung

Im Bereich der Musikempfehlungssysteme wurden in den letzten Jahren eine Reihe von Datensätzen unter verschiedenen Gesichtspunkten publiziert. Einer der ersten Datensätze für Musikempfehlungen war der Last.fm Datensatz von Òscar Celma [43]. Dieser enthält die Häufigkeit der Künstler die sich ein Benutzer auf der Plattform angehört hat. Dazu gibt es noch zu jedem Künstler einen Verweis auf den Musik Informationsdienst Music Brainz¹. Der Datensatz wurde im Mai 2009 erstellt. Ein weiterer Datensatz [9], der auf Grundlage von Last.fm erstellt wurde, enthält neben den Hörgewohnheiten der Benutzer zudem noch die Beziehungen zwischen Nutzern auf der Plattform und welche Schlagworte(Tags) die Benutzer den Künstlern gegeben haben. Der Datensatz von Bertin-Mahieux et al. [6] enthält zwar keine Benutzer, jedoch zu jedem Song eine MusicbrainzID (mbid) womit dieser mit den vorherigen Datensätze kombinieren lässt. Jedes der Musikstücke besitzt eine große Anzahl von Merkmalen, die mit Hilfe des Echo Nest Projektes² aus den Audiosignalen herausgefiltert wurden. Einen anderen Ansatz verfolgt der Million Musical Tweets Datensatz von Hauger et al. [16]. Dieser wurde anhand von Nachrichten beim Kurznachrichtendienst Twitter durch die Suche bestimmter Hashtags, wie zum Beispiel #nowplaying, erstellt. Das Profil der Benutzer wurde mit dem aktuellen Standort, sowie Längen- und Breitengrad angereichert. Erstellt wurde der Datensatz zwischen September 2011 und April 2013. Den gleichen Ansatz verfolgt der #Nowplaying Music Dataset von Zangerle et al. [39]. Dieser erhebt jedoch den Anspruch auf Aktualität und hat einen sehr viel größeren Umfang als die vorher erwähnten Datensätze und wird deswegen in dieser Arbeit zur Evaluation der Verfahren verwendet, da davon ausgegangen wird, dass in diesen sich auch die Verschiedenartigkeit der Künstler wiederfinden lässt. In der Tabelle 4.1 wird der Umfang der Datensätze gegenübergestellt.

4.1.1 Vorverarbeitung

Dadurch das der Datensatz sehr groß ist, ist die User \times Item Matrix außerordentlich dünn besetzt. Dadurch werden viele Ressourcen zur Berechnung der Empfehlungslisten zu benötigt. Um überhaupt Berechnungen zu ermöglichen wurde alle Benutzer aus dem Datensatz entfernt, die weniger als 50 verschiedene Künstler gehört haben,

¹<https://musicbrainz.org/>

²<http://echonest.com/>

Datensatz	Einträge	Benutzer	Künstler	Songs
Celma 1K	19,150,868	992	176,948	1,084,620
Celma 360K	17,559,530	359,347	294,015	—
HetRec Last.FM	92,834	1892	17,632	—
Million Song Dataset	1,000,000	—	44,745	1,000,000
Musical Tweets Dataset	1,086,808	215,375	25,060	133,968
#Nowplaying	67,699,640	5,081,320	112,190	916,545

Tabelle 4.1: Gegenüberstellung der Datensätze für Musikempfehlungen

damit wird das New-User Problem 2.1.2 umgangen. Zusätzlich wurden alle Künstler entfernt, die weniger als 2 mal gehört wurden, um so das New-Item Problem 2.1.2 zu umgehen. Der resultierende Datensatz enthält nach der Vorverarbeitung 47.665 Benutzer, 806.846 Titel und 95.135 Künstler.

4.1.2 Bewertungen

Da die Bewertungen nur als implizite Hörereignisse der Benutzer vorliegen, ist es notwendig diese in eine Schema zu überführen, das dem einer Bewertungsstruktur entspricht, damit die der klassische Ansatz des kollaborativen Filterns auf den Datensatz angewendet werden kann.

Eine einfache Methode zur Übersetzung von impliziter Informationen in explizite Bewertungen, wäre die Häufigkeit der Hörereignisse eines Benutzer im Bezug zu einem Künstler zu berechnen. Also wie oft hat der Benutzer einen bestimmten Künstler angehört. Leider hat sich gezeigt, dass sich die Verwendung von Häufigkeiten in kollaborativen Empfehlungssystem negativ auf die Vorhersagegenauigkeit auswirkt. Der Grund hierfür ist, dass die gewichtete Summe 2.6 eine eingegrenzte Bewertungsskala zur Berechnung der Bewertungen benötigt [13]. Deshalb werden im folgenden zwei Verfahren vorgestellt, die die impliziten Informationen auf ein Intervall abbilden, das den Wertebereich der expliziten Bewertungen entspricht.

Kumulierte Verteilung

Wie beschrieben, können Verfahren zum kollaborativen Filtern nur auf expliziten Bewertungen angewandt werden. Die impliziten Informationen des Datensatzes müssen also in eine Form gebracht werden, die einen Tripel ($user_id$, $artist_id$, $rating$) entspricht. Für gewöhnlich ist die Verteilung der Hörgewohnheiten stark nach rechts verzerrt und entspricht einer endlastigen Verteilung [1]. Einige wenige Künstler vereinigen also eine große Anzahl der Wiedergaben auf sich und der Rest der Künstler hat nur einige wenige Wiedergaben. In der Abbildung 4.1 ist eine Beispielverteilung der Wiedergaben eines Benutzers dargestellt. Zur Normalisierung wird die komplementäre kumulative Verteilung der Häufigkeiten der Wiedergaben der Künstler eines Benutzers berechnet. Der Hauptgedanke hierbei ist, dass die Bewertungsstruktur

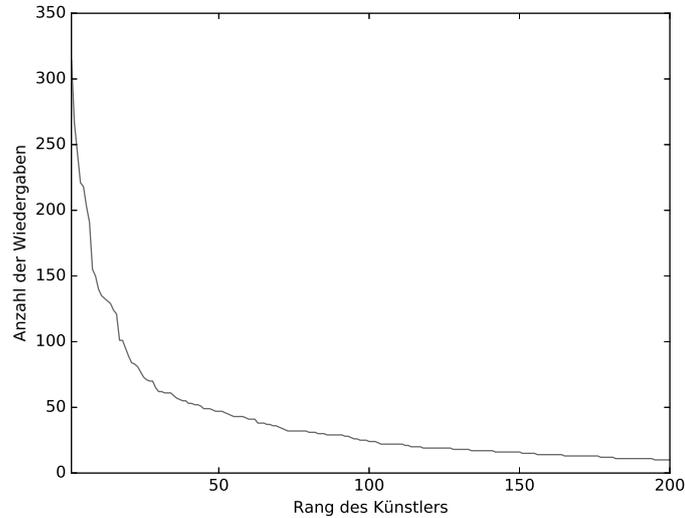


Abbildung 4.1: Verteilung der Anzahl der Wiedergaben der Künstler eines Benutzers

durch die Verteilung der Häufigkeiten widergespiegelt wird. Also nur wenige Benutzer eine hohe Bewertung erhalten und viele seltener gehörte Künstler schlechter bewertet werden [43]. Die Künstler die im 80-100% Bereich der Verteilung liegen wird der Wert 5 als Bewertung zugewiesen. Für die Künstler im Bereich 80-60% liegen, bekommen den Wert 4. Damit wird fortgefahren, bis für die letzten Künstler im Bereich 20-0% der Wert festgelegt wird. In der Abbildung 4.2 ist eine komplementäre kumulative Verteilung eines Benutzer mit den Bereichen, die für die Zuweisung der Werte verwendet werden.

Logarithmus-basierte Normalisierung

Eine weitere Methode zur Normalisierung der Häufigkeiten ist der Logarithmus basierte Ansatz und stützt sich auf die in [18] verwendeten Verfahren. In der Arbeit wird gezeigt, dass der Logarithmus von impliziten Häufigkeiten die expliziten Bewertungen in annehmbarer Art und Weise abbilden kann. Um eine Normalisierung auf das Intervall von $[0, 1]$ zu erreichen, wird der berechnete Wert noch durch den Logarithmus des am häufigsten gehörten Künstlers geteilt. In Gleichung 4.1 ist die Berechnung der Bewertung $r_{i,j}$ formalisiert und anschließend wird noch auf den Wertebereich 0 bis 5 der Bewertungen skaliert.

$$r_{i,j} = \frac{\log(1 + \text{freq}(u_i, a_j))}{\log(1 + \max_{a \in A}(\text{freq}(u_i, a)))} \quad (4.1)$$

Wobei u_i der i -te Benutzer und a_j der j -te Künstler ist und die Funktion freq die Häufigkeit angibt, wie oft der Benutzer u_i den Künstler a_j gehört hat. Die Menge A sind alle Künstler, die von dem Benutzer u_i gehört hat. Im Gegensatz zum ersten

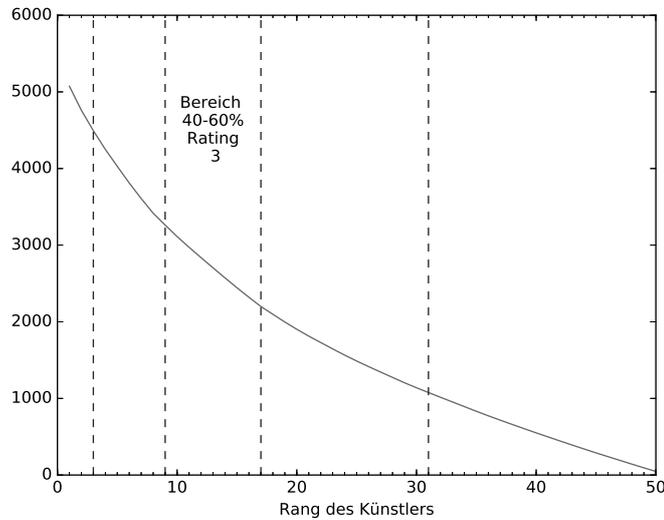


Abbildung 4.2: komplementäre kumulative Summe der Häufigkeiten von gehörten Künstlern

Verfahren der kumulierten Verteilung 4.1.2, hat der Logarithmus basierte Ansatz einen kontinuierlichen Wertebereich und keine beschränkte Menge von Bewertungen.

4.2 Implementierung

Für die Implementierung aller der in der Arbeit beschriebenen Verfahren und Evaluationen wurde die Programmiersprache Python ³ in der Version 3.4 verwendet. Die Wahl fiel auf diese Programmiersprache, weil es viele gut gepflegte Bibliotheken für die Berechnung mathematischer Probleme und zur Verarbeitung von Daten aus dem Bereich des maschinellen Lernens existieren ⁴. Als Referenzsystem, um eine Vergleichbarkeit der Ergebnisse herzustellen, wurde ein Empfehlungssystem mit dem klassischen Ansatz des kollaborativen Filterns entwickelt. Zur Vorhersage der Bewertungen wurde der Ansatz der benutzerbasierten Nachbarschaftssuche verwendet. Ein inhaltsbasiertes System wurde nicht implementiert, da hierfür die Beschreibung der Objekte in dem Datensatz nicht vorhanden waren und die Extraktion von Merkmalen aus Multimedia Daten, wie Musik oder Filme, immer noch fehleranfällig sind. Die objektbasierte Nachbarschaftssuche kam deshalb nicht zum Einsatz, da die Vorteile bei diesem Ansatz nur zum tragen kommen, wenn deutlich mehr Benutzer, als Objekte im Datensatz vorhanden sind. Darüber hinaus wurde bereits in 2.1 beschrieben, dass der Ansatz des kollaborativen Filterns die Fähigkeit hat, dass die Empfehlungen vielfältig sind [8].

³<https://www.python.org/>

⁴<https://www.scipy.org>

4.3 Validierung

Um die vorgestellten Verfahren mit den beschriebenen Metriken zu evaluieren, kommt die *k-fold Cross Validation* zum Einsatz [21]. Diese Methodik zur Evaluation ist vor allem im Bereich des maschinellen Lernens weit verbreitet, wird aber auch bei der Evaluation von Empfehlungssystemen angewandt [17]. Hierfür wird der vorhandene Datensatz in k gleich große Partitionen aufgeteilt. Jede Partition wird als eigenständiger Testdatensatz zur Evaluation der Ergebnisse verwendet. Auf den restlichen $k - 1$ Partitionen werden die Modelle der Verfahren trainiert, um mithilfe dieser die Vorhersagen für die Benutzer in der Testpartition zu erstellen. Es werden also k Durchläufe benötigt, um ein Verfahren zu evaluieren. Die Ergebnisse werden über k gemittelt. In der vorliegenden Arbeit, werden die vorgestellten Verfahren mit dem Parameter $k = 5$ evaluiert.

5 Ergebnisse

5.1 Evaluationsmetriken

5.1.1 Genauigkeitsmetriken

Der *Root Mean Squared Error* (RMSE) ist die wohl am häufigsten verwendete Metrik zur Evaluation von Empfehlungssysteme in der Literatur [4, 17]. In Offline Experimenten wird zur Berechnung des RMSE der Datensatz in einen Trainings- und Testdatensatz geteilt. Das Empfehlungssystem wird auf dem Trainingsdatensatz angelernt und dann auf die mit dem Testdatesatz verglichen. Hierfür werden die wahren Bewertungen y_i vor dem System geheim gehalten und die Fehlerquadrate mit der vorhergesagten Bewertung f_i berechnet.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2} \quad (5.1)$$

Eine weitere häufig zum Einsatz kommende Alternative ist der *Mean absolute error* (MAE), der nicht die Fehlerquadrate summiert, sondern den absoluten Fehler zwischen Vorhersage und wahrer Bewertung berechnet.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \quad (5.2)$$

Der *RMSE* bestraft, im Gegensatz zum *MAE*, größere Fehler mehr als kleinere und ist deswegen häufig die Metrik der Wahl zur Evaluierung von Empfehlungssystemen.

5.1.2 Klassifikationsmetriken

Bei Systemen die keine Bewertung von Objekten für Benutzer vorhersagen sollen, sondern dem Benutzer nur Objekte vorschlagen soll, können die oben aufgeführten Fehlermetriken nicht verwendet werden. Dadurch entwickelt sich die Evaluation dieser Systemen zu einem klassischen Klassifikationsproblem. Befinden sich die vorgeschlagenen Objekte in der Klasse der für den Benutzer nützlichen, also im Testdatensatz befindlichen, Objekte oder werden vom System Dinge empfohlen, die den Nutzer des Datensatzes nicht interessieren scheinen. In Tabelle 5.1 werden die Möglichkeiten der Klassifikationen dargestellt. Die Precision misst den Anteil der richtig klassifizierten Objekte. Hierfür wird die Anzahl der richtig empfohlenen Objekte durch die Anzahl aller empfohlenen Objekte berechnet. Eine Abwandlung ist

	Recommended	Not recommended
Used	True-positive (tp)	False-negative (fn)
Not used	False-positive (fp)	True-negative (tn)

Tabelle 5.1: Konfusionsmatrix bei Empfehlungssystemen

der Precision@N, der sich nur auf die ersten N Empfehlungen beschränkt und nicht über die Gesamtheit der Objekte im Datensatz berechnet wird.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.3)$$

Beim Recall handelt es sich um einen Kennwert, der die für einen Benutzer relevanten Objekte im Verhältnis zu der Gesamtheit der relevanten Objekte für alle Benutzer im Testdatensatz setzt.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.4)$$

Das F1-Maß oder auch F1-Measure ist eine Kombination der beiden oben genannten Metriken für das Klassifikationsproblem Precision und Recall. Es berechnet das harmonische Mittel beider Werte bei dem der Precision und Recall gleich gewichtet werden [30].

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (5.5)$$

Zur grafischen Darstellung kann die sogenannte Receiver-Operating-Characteristic-Kurve (ROC-Kurve) zum Einsatz kommen [30]. Die ROC-Kurve veranschaulicht die Abhängigkeit zwischen richtig und falsch klassifizierten Objekten des Empfehlungssystem. Wäre das dargestellte Empfehlungssystem perfekt, würde die Kurve aus nur einem Punkt in der oberen linken Ecke der Abbildung bestehen. Das ist dann der Fall, wenn die True positive Rate 1 ist und die False positive Rate hingegen 0. Mithilfe der ROC Kurve können auch, für das Empfehlungssystem benötigte, Parameter optimiert werden, indem für verschiedenen Werte der Precision und Recall gegeneinander aufgetragen werden.

Da es bei den vorgestellten Verfahren in 3.2 immer darum geht einen Ausgleich zwischen der Vielfalt und der Genauigkeit bei den Vorhersagen zu erreichen, wird zur Evaluation der Verfahren eine abgeänderte Variante des im diesem Abschnitt vorgestellten harmonischen Mittels verwendet. Hierfür wird für ein Maß $F1_{div}$ für eine der vorgestellten Metriken für Vielfalt in Abschnitt 3.3 und dem Precision definiert:

$$F1_{div} = 2 \cdot \frac{\text{precision} \cdot \text{div}}{\text{precision} + \text{div}} \quad (5.6)$$

5.2 Evaluation

Im vorliegenden Abschnitt werden das benutzerbasierte kollaborative Empfehlungssystem, das in dem Teil 2.1.2 der Arbeit vorgestellt wurde, und die Verfahren aus dem dem Kapitel 3 mit den Metriken, die im Abschnitt 3.3 beschrieben wurden, verglichen. Zu Beginn werden die Ergebnisse der Transformationen für die Bewertungen anhand der Genauigkeit betrachtet. Danach wird der Einfluss der Länge der Empfehlungslisten für das kollaborative System evaluiert. Anschließend werden die Parameter für die Verfahren zur Steigerung der Vielfalt in den Empfehlungen ermittelt. Zum Schluss werden alle Verfahren gegenübergestellt und mit den besprochenen Metriken verglichen und diskutiert.

5.2.1 Ermittlung der Parameter

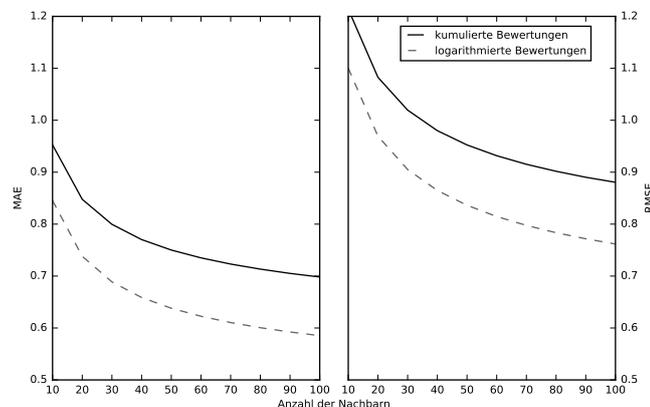


Abbildung 5.1: Genauigkeit der beiden Bewertungstransformationen

In Abbildung 5.1 wird die Genauigkeit des kollaborativen Empfehlungssystems anhand des MAE und des $RMSE$ in Abhängigkeit von der Anzahl der Nachbarn k dargestellt. Zum Vergleich wurden die beiden Ansätze zur Transformation der Bewertungen von impliziten Informationen in explizite Werte visualisiert. Sowohl für den MAE , als auch für den $RMSE$ ist zu erkennen, dass bei dem logarithmische Ansatz zur Berechnung der expliziten Bewertungen der Fehler deutlich kleiner ist, als bei dem Ansatz über die kumulierte Summe der Wiedergabehäufigkeiten. Bei der Evaluation der folgenden Verfahren werden deswegen die logarithmischen Bewertungen zur Veranschaulichung verwendet. Des weiteren ist zu erkennen, dass die Fehler für eine größere Nachbarschaft immer kleiner werden, da bei einer kleinen Nachbarschaft weniger Bewertungen zur Berechnung der Vorhersage vorliegen. Bei einer größeren Nachbarschaft hingegen, ist die Auswahl an Objekten, die vorgeschlagen werden können höher. Dadurch kann das Verfahren die Präferenzen des Benutzers besser berücksichtigen. Ab einer Nachbarschaft von $k = 50$ ist zu erkennen, dass der

Fehler der Vorhersage nur noch marginal kleiner wird. In der Literatur wird ebenfalls ein optimaler Wert für die Nachbarschaft mit $k = 50$ angegeben [17].

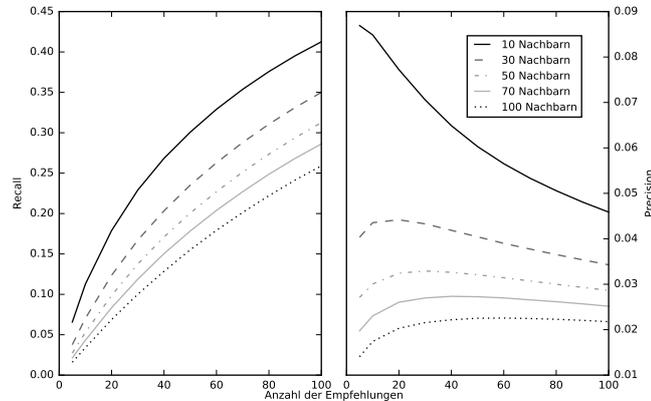


Abbildung 5.2: Klassifikationsmetriken des kollaborativen Systems mit logarithmischen Bewertungen

In Abbildung 5.2 werden Klassifikationsmetriken für verschiedene Nachbarschaften in Abhängigkeit von der Länge der Empfehlungsliste dargestellt. Der Wert des Recall ist ansteigend für längere Empfehlungslisten. Ein erwartbares Verhalten, da durch eine längere Empfehlungsliste mehr Objekte der Benutzer abgedeckt werden können. Für die Nachbarschaft $k = 10$ ist der Recall durchgängig am besten und für $k = 100$ am schlechtesten. Der Grund hierfür wird sein, dass für eine kleine Nachbarschaft nur die nächsten Benutzer zur Erstellung der Empfehlungslisten verwendet werden und diese ein ähnliches Bewertungsverhalten haben, wie der aktive Benutzer. Die Precision nimmt tendenziell ab für größere Empfehlungslisten. Bei mehr als 10 Nachbarn stabilisiert sich der Wert der Precision nach einer Größe von 50 Objekten. Bei $k = 10$ Nachbarn fällt die Precision im Verhältnis zu den anderen Kurven viel schneller bei größeren Empfehlungslisten. Der Grund hierfür wird sein, dass für längere Empfehlungsliste zu wenig Objekte für eine kleine Zahl von Nachbarn bereitstehen, um diese den Empfehlungen zu berücksichtigen.

In Abbildung 5.3 wird das harmonische Mittel aus Precision und Recall in Abhängigkeit der Länge der Empfehlungsliste für mehrere Nachbarschaften dargestellt. Die Kurven sind alle zusammen steigend und haben bei einer kleineren Nachbarschaft bessere Ergebnisse. Im Allgemeinen kann das harmonische Mittel genutzt werden, um den optimalen Wert für die Länge der Empfehlungsliste zu ermitteln. In diesem Fall überwiegt jedoch der Recall so eindeutig, dass die Kurven monoton steigend sind und keine Optimum gefunden werden kann. Da in der Arbeit eine Abwägung zwischen Genauigkeit und Vielfalt untersucht werden soll, wird die Länge der Empfehlungsliste mit $N = 50$ festgelegt, da für eine größere Nachbarschaft eine Sättigung der Precision ab diesen Wert eintritt. Die Werte für die Nachbarschaft und für die Anzahl der Empfehlungen werden verwendet, um optimalen Parameter für die vorgestellten Verfahren aus dem Kapitel 3 zu ermitteln.

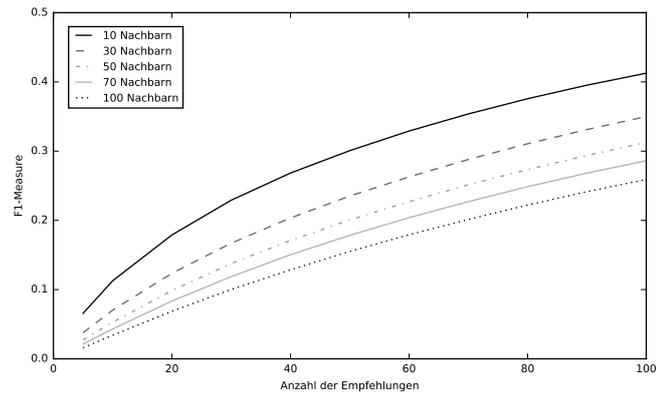


Abbildung 5.3: Klassifikationsmetriken des kollaborativen Systems mit logarithmischen Bewertungen

5.2.2 Reranking Verfahren

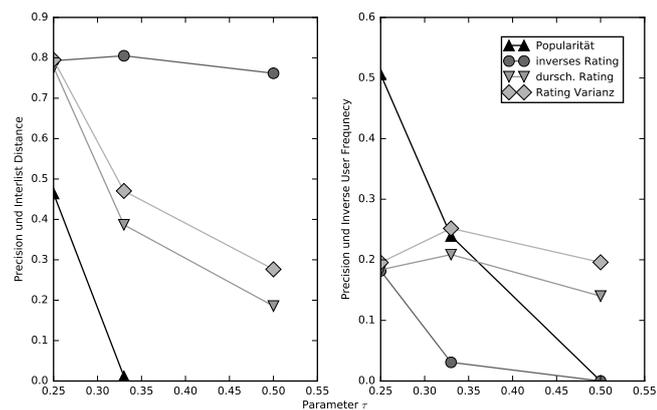


Abbildung 5.4: Reranking Verfahren in Abhängigkeit vom Parameter τ

In Abbildung 5.4 werden die Reranking Verfahren aus dem Abschnitt 3.2.3 in Abhängigkeit zu dem Parameter τ dargestellt. Der Parameter τ soll dabei ein Ausgleich zwischen der Precision und der Vielfalt der Empfehlungslisten herstellen. Zur Visualisierung und Ermittlung des optimalen Parameters wird das harmonische Mittel über die Precision und einer der in Abschnitt 3.3 vorgestellten Metriken für die Vielfalt verwendet. Bei dem Vergleich zwischen der Precision und der Intra-List Distance zeigt sich, dass das Reranking anhand der umgekehrten Popularität der Objekte in der Empfehlungsliste am schlechtesten abschneidet. Ein zu erwartendes Verhalten, da unpopuläre Künstler nicht notwendigerweise den Präferenzen der Benutzer entspricht. Das beste Ergebnis liefert die Neuordnung der Elemente anhand der inversen Bewertungen des Benutzers. Das lässt sich erklären, dass durch das Verfahren vor allem die Genauigkeit des kollaborativen Systems profitiert und einen

geringen Teil an Vielfalt in den Empfehlungslisten einführt. Bei der zweiten Kurve werden die Precision und die Globale Novelty gegenübergestellt. Wie schon im ersten Bild sind die beiden Ansätze über die durchschnittlichen Bewertungen und die Varianz der Bewertung am stabilsten bei der Änderung des Parameters τ . Diesmal schneidet der Ansatz der inversen Bewertungen am schlechtesten ab und stützt unsere Beobachtung aus dem ersten Bild. Das beste Ergebnis liefert der Ansatz über die inverse Popularität mit dem Parameter $\tau = 0.25$. Auch dieses Ergebnis war zu erwarten, da den Benutzern neue und unbekannte Künstler vorgeschlagen werden.

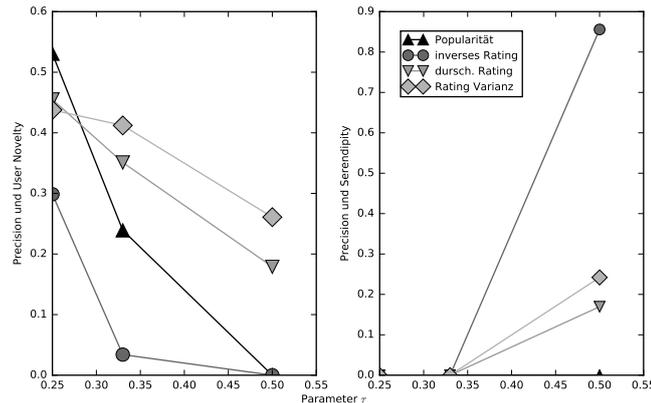


Abbildung 5.5: harmonisches Mittel aus Precision und Reranking Verfahren in Abhängigkeit vom Parameter τ

Im dem ersten Bild der Abbildung 5.5 werden die Precision und die User Novelty gegenübergestellt. Wieder sticht der Ansatz der umgekehrten Popularität für kleines τ heraus. Die Neuordnung der Empfehlungen anhand inverser Bewertungen können den Benutzern kaum neue Künstler vorschlagen. Im zweiten Graph der Abbildung 5.5 wird das harmonische Mittel aus Serendipity und Precision über den Parameter τ visualisiert. Zu sehen ist, dass die verwendeten Verfahren kaum Künstler vorschlagen, die sowohl nützlich, als auch neu für den Benutzer sind. Der Ansatz für inverse Bewertungen schafft es als einziges Verfahren für ein großes $\tau = 0.5$ die Serendipity zu steigern ohne das die Precision zu stark abnimmt.

5.2.3 Clustering der Benutzerprofile

In Abbildung 5.6 wird der Ansatz des Clustering der Benutzerprofile aus dem Abschnitt 3.2.2 in Abhängigkeit zu der Anzahl der Cluster k dargestellt. Die Kurven bilden hierbei die Ergebnisse der verschiedenen Metriken für Vielfalt im Verhältnis zu der Precision ab. Bei der Interlist Distance zeigt sich ein Optimum für $k = 5$ Cluster. Der Hintergrund ist, dass bei wenigen Clustern die Vielfalt in den Empfehlungslisten zu gering ist und bei einer großen Anzahl von Clustern, die Genauigkeit der Empfehlungen darunter leidet. Die beiden Metriken für Novelty haben ebenfalls ein Maximum für $k = 5$ Cluster. Nur die Catalog Coverage liefert das bessere Er-

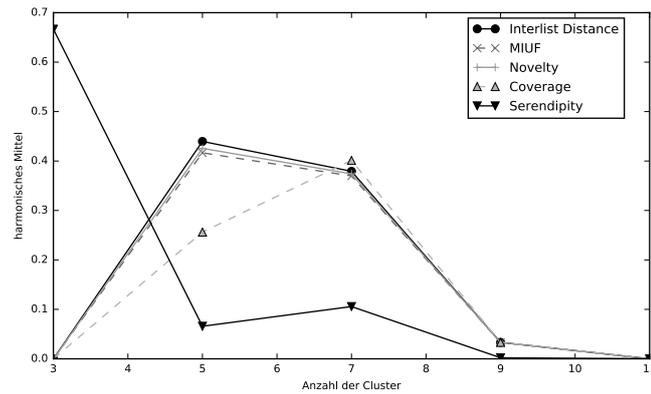


Abbildung 5.6: harmonisches Mittel aus Precision und Metriken für Vielfalt für User Profile Clustering über die Anzahl der Cluster

gebnis für $k = 7$ Cluster und die Serendipity fällt bei einer größeren Anzahl von Clustern. Das bedeutet vor allem, dass obwohl die Vielfalt in den Empfehlungslisten gesteigert werden konnte, die Precision entsprechend abnimmt.

5.2.4 Greedy Selection

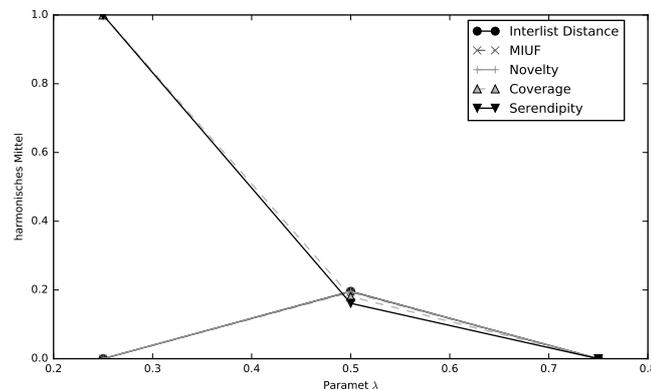


Abbildung 5.7: harmonisches Mittel aus Precision und Metriken für Vielfalt für die Greedy Selection über den Parameter λ

In Abbildung 5.7 werden die die für Vielfalt für den Greedy Selection Ansatz aus dem Abschnitt 3.2.1 über die Anzahl der Nachbarn dargestellt. Zu sehen ist, dass für kleine λ eigentlich keine Steigerung der User Novelty oder Intra-List Distance in der Empfehlungsliste zu erkennen ist. Im Gegensatz dazu, kann jedoch die Serendipity und Inverse User Frequency im Verhältnis zu der Precision deutlich verbessert werden. Bei $\lambda = 0.5$ liegen die Ergebnisse der Metriken dicht beieinander. Die Intra-List

Distance, Catalog Coverage und Inverse User Frequency können leicht gesteigert werden, wohingegen die anderen Metriken für größere λ stark abfallen. In nächsten Teil der Arbeit wird gezeigt, dass vor allem die schlechte Genauigkeit ein Grund für das abschneiden der Greedy Selection ist.

5.2.5 Vergleich der Verfahren

Nachdem im letzten Abschnitt die optimalen Parameter für die verschiedenen Verfahren zur Steigerung der Vielfalt in Empfehlungssystem bestimmt wurde, werden in vorliegenden Abschnitt die ermittelten Parameter verwendet, um eine optimale Vergleichbarkeit zwischen den Verfahren und dem kollaborativen System herzustellen. In Tabelle 5.2 sind die optimalen Verfahren für die unterschiedlichen Verfahren verzeichnet.

Verfahren	Parameter	Wert
kollaboratives Filtern	Anzahl der Empfehlungen N	50
Greedy Selection	λ	$\frac{1}{2}$
Popularität Ranking	τ	$\frac{1}{5}$
inverses Ranking	τ	$\frac{1}{3}$
Durch. Ranking	τ	$\frac{1}{2}$
Varianz Ranking	τ	$\frac{1}{3}$
Clustering	Cluster k	5

Tabelle 5.2: Optimale Parameter für die Verfahren

5.2.6 Genauigkeit

Zunächst wird die Genauigkeit der Vorhersagen anhand der Abbildung 5.8 über die Anzahl der Nachbarn betrachtet. Hierbei sind vor allem das schlechte Ergebnis der Greedy Selection und das gute Ergebnis des Clustering Verfahrens bemerkenswert. Die beiden anderen Verfahren verhalten sich ähnlich und sind im Mittelfeld der Genauigkeit angesiedelt. Wie schon beim kollaborativen System aus der Abbildung 5.2 ist zu erkennen, dass alle Verfahren bei einer größeren Nachbarschaft an Genauigkeit einbüßen.

5.2.7 Inter-List Distance

In Abbildung 5.9 wird die Diversity in den Empfehlungslisten der vorgestellten Verfahren in Abhängigkeit zu der Nachbarschaft untersucht. Zu erkennen ist, dass das Verfahren zum Clustern der Benutzerprofile am besten dazu geeignet ist, einen Ausgleich zwischen der Diversity in Empfehlungslisten und der Genauigkeit herzustellen. Die Kurve für das Clustering Verfahren ist dazu auch noch steigend, also die Vielfalt in der Empfehlungsliste steigt zudem bei einer größeren Nachbarschaft mit diesem Verfahren. Am schlechtesten schneidet der Ansatz der Greedy Selection ab, was auch

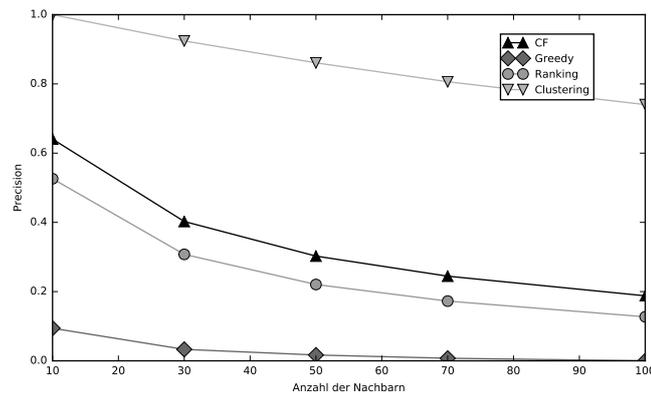


Abbildung 5.8: Precision über die Anzahl der nächsten Nachbarn

mit der schlechten Genauigkeit der Vorhersagen zusammenhängen wird. Das kollaborative Filtern und der Ranking basierte Ansatz befinden sich wieder im Mittelfeld und unterscheiden sich nur marginal.

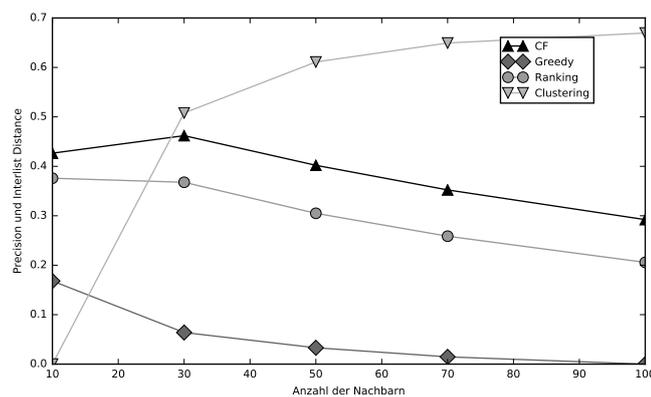


Abbildung 5.9: harmonisches Mittel aus Precision und Inter-List Distance über die Anzahl der nächsten Nachbarn

5.2.8 Catalog Coverage

In Abbildung 5.10 wird die globale Diversity anhand der Catalog Coverage im Zusammenhang mit der Precision betrachtet. Wie schon bei der Interlist Distance aus Abbildung 5.9 liefert das Clustering Verfahren die besten Ergebnisse und die Greedy Selection das schlechteste Ergebnis. Auffällig ist nur, dass für eine kleine Nachbarschaft $k \leq 30$ der Ranking Ansatz die Diversity systemweit steigern kann. Für eine größere Nachbarschaft jedoch ein schlechteres Ergebnis liefert, als der Clustering Ansatz. Außerdem ist zu erkennen, dass sich der kollaborative Ansatz und das Ranking Verfahren für $k > 30$ annähern.

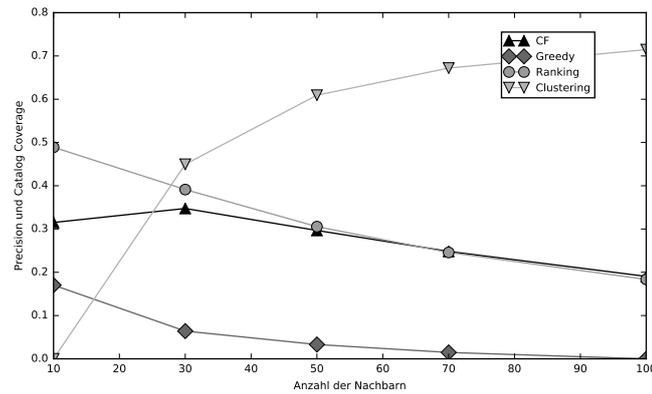


Abbildung 5.10: harmonisches Mittel aus Precision und Catalog Coverage über die Anzahl der nächsten Nachbarn

5.2.9 Inverse User Frequency und User Novelty

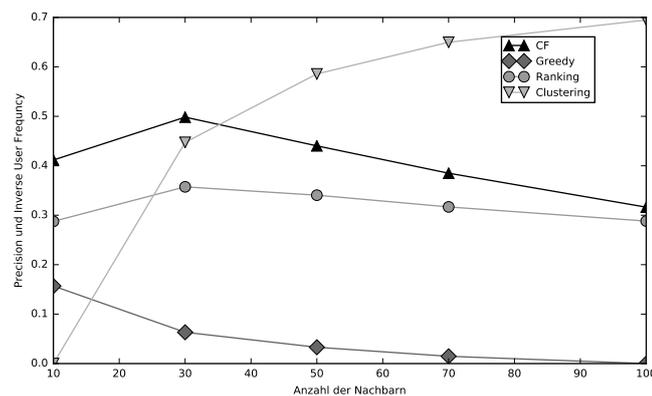


Abbildung 5.11: harmonisches Mittel aus Precision und globale Novelty (MIUF) über die Anzahl der nächsten Nachbarn

Die globale Novelty wird in Abbildung 5.11 anhand der Nachbarschaft betrachtet. Für eine kleine Nachbarschaft $k \leq 50$ scheinen das kollaborative Filtern und das Ranking Verfahren anhand der inversen Bewertungen bessere Ergebnisse zu liefern. Erst für größere k kann wieder der Ansatz des Clusterings die besseren Ergebnisse liefern. Ein Hintergrund wird sein, dass für eine kleine Anzahl an Nachbarn die Cluster zu klein werden, und entsprechend die Clusterzentren sehr nahe an den eigentlichen Objekten der Benutzer liegen, um den Benutzer neuartige Objekte vorzuschlagen. Ein ganz ähnliches Verhalten lässt sich auch in Abbildung 5.12 wiederfinden. Hier wird die User Novelty im Zusammenhang mit der Precision gemessen und ist zu erkennen, dass das Clustering für kleine $k \leq 30$ eigentlich keine neue Objekte für die Benutzer vorschlagen kann. In beiden Abbildungen 5.11, 5.12 ist zu sehen, dass das

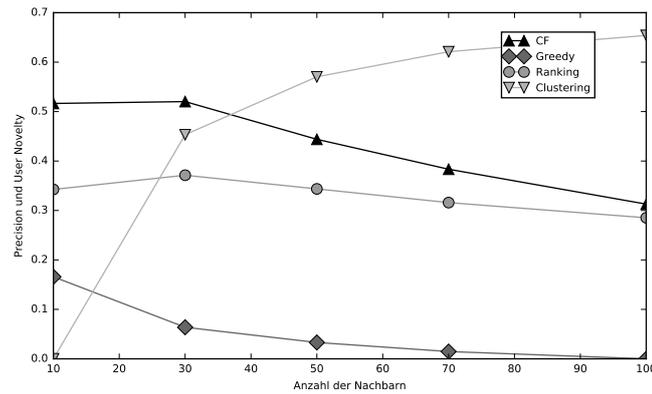


Abbildung 5.12: harmonisches Mittel aus Precision und User Novelty über die Anzahl der nächsten Nachbarn

kollaborative Filtering und der Reranking Ansatz konstant neue Objekte empfehlen können, unabhängig von der Anzahl der Nachbarn.

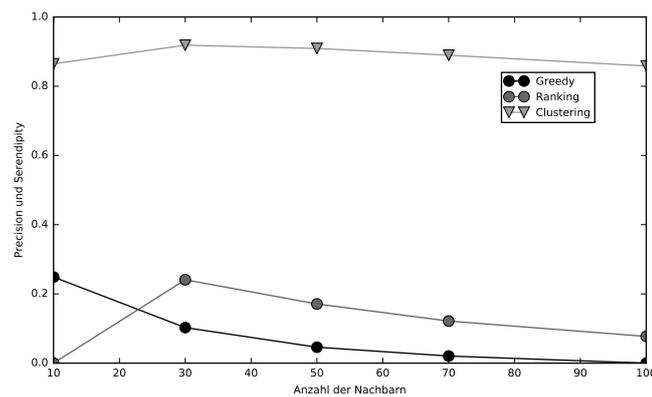


Abbildung 5.13: harmonisches Mittel aus Precision und Serendipity über die Anzahl der nächsten Nachbarn

In der letzten Abbildung 5.13 wird die Serendipity in Zusammenhang mit der Precision und in Abhängigkeit zur der Nachbarschaft dargestellt. In der Darstellung fehlt das kollaborative Filtern, da in Abschnitt 3.1.3 die erwartbaren Objekte eines Benutzers, definiert wurden als diejenigen Objekte, die von einem primitiven System vorgeschlagen werden. In dieser Arbeit wurde das kollaborative Filtern als primitives System verwendet. Nochmals ist zu erkennen, dass das Clustering deutlich besser ist, als die beiden anderen Verfahren. Für eine Nachbarschaft $k = 10$ schafft es erstmals die Greedy Selection bessere Ergebnisse zu liefern, als der Reranking Ansatz. Bei einer Vergrößerung der Nachbarschaft verschlechtert sich das Ergebnis der Greedy Selection zusehends und der Reranking Ansatz liefert bessere Ergebnisse.

5.3 Diskussion

Im Folgenden werden zunächst die Ergebnisse der verschiedenen Verfahren und der ermittelten optimalen Parameter betrachtet. Im zweiten Teil werden die Ergebnisse zu den unterschiedlichen Eigenschaften besprochen. Zum Schluss wird auf Beantwortung der Fragestellung, ob sich Genauigkeit und Vielfalt in Empfehlungssystem vereinen lassen, eingegangen. Für das kollaborative Empfehlungssystem ist die Anzahl der benachbarten Benutzer und die Länge der Empfehlungsliste von entscheidender Bedeutung. Die Ergebnisse für die klassischen Metriken MAE und $RMSE$ waren erwartbar. Mit einer höheren Nachbarschaft steigt auch die Vorhersagegenauigkeit. Bei den Metriken für die Klassifikation gestalten sich die Ergebnisse etwas überraschender, denn für eine geringe Anzahl von Nachbarn ist die Vorhersage besser. Der Grund hierfür liegt vermutlich an der verwendeten Datenbasis und an dem kollaborativen Empfehlungssystem, da bei einer größeren Nachbarschaft die Ähnlichkeit zu dem aktiven Benutzer abnimmt. Insgesamt lässt sich die niedrige Precision auf die Art der Evaluation zurückführen, da das System vor der Erstellung der Empfehlungen kein Wissen über die Benutzer hatte. Der Anstieg des Recalls bei mehr Empfehlungen, ist verständlich, da mit dieser Metrik die Abdeckung der empfohlenen Objekte in dem Benutzerprofil gemessen wird.

Bei den Reranking-Verfahren wurde auf den Ausgleich zwischen Genauigkeit und Vielfalt bei der Evaluation besonderen Wert gelegt. Abhängig von den Eigenschaften habe spezifische Verfahren Vorteile gegenüber den Anderen. Daraus lässt sich erkennen, dass Verfahren die die Vielfalt in den Empfehlungen steigern, nicht notwendigerweise auch für die Empfehlungen von neuartigen Objekten eingesetzt werden können. Der Parameter τ hat ersichtlichen Einfluss auf die Vielfalt der Empfehlungen. Der Grund liegt in der Natur der Reranking Ansätze. Wenn die Empfehlungsliste anhand der inversen Popularität neu geordnet wird, dann werden auch mehr unbekannte Objekte empfohlen, aber es werden nicht zwingend unterschiedliche Objekte vorgeschlagen. Das verwendete Verfahren hängt als stark von dem Einsatzgebiet und den Ziele des Empfehlungssystems ab.

Für das Clustering der Benutzerprofile lässt sich eine optimale Anzahl von Clustern bei $k = 5$ ausmachen, da dort die Eigenschaften für das System am ausgewogensten sind. Nur das Ergebnis der Interlist-Distance verhält sich für eine kleinere Anzahl von Empfehlungen besser. Dieses Resultat wird damit zusammenhängen, das die Objekte in dem Benutzerprofile durch die berechneten Zentren der Cluster verallgemeinert werden und bei einer höheren Anzahl an Clustern, die Zentren mit den Objekten aus dem Benutzerprofil zu sehr ähneln.

Der Parameter λ für den Ansatz der Greedy Selection hat nur wenig Auswirkung auf die Vielfalt in der Liste der Empfehlungen. Insgesamt muss für diesen Ansatz gesagt werden, dass er die Precision in einem erheblichen Maß verschlechtert, aber kaum Vielfalt in die Liste der Empfehlungen einführt. Das Ziel diese Eigenschaften zu kombinieren konnte also nicht erreicht werden und kann nur in der Auswahl der Objekte anhand ihrer Relevanz und Vielfalt begründet sein.

Das die Precision mit dem Ansatz des Clustering für Benutzerprofile so deutlich gesteigert werden konnte, ist ein unerwartetes Ergebnis. Zu erwarten war das alle Verfahren eine etwas schlechtere Vorhersagegenauigkeit, als das kollaborative Filtern haben, da das Ziel war die Vielfalt in die Liste der Empfehlungen einzuführen. Es ist anzunehmen, dass durch die Abstraktion der Objekte in dem Benutzerprofil anhand der Clusterzentren die Ergebnisse verzerrt werden und bei einer höheren Anzahl von Nachbarn die Precision ebenfalls abnimmt. Das würde sich auch mit den Ergebnissen für das kollaborative Filtern decken, bei dem die Precision höher ist für eine kleinere Anzahl von Nachbarn.

Anhand der Interlist-Distance ist zu erkennen, dass eigentlich nur der Clustering Ansatz die Diversity in den Empfehlungslisten der Benutzer steigern kann. Denn im Gegensatz zu den anderen Verfahren ist die Tendenz, bei einer größeren Nachbarschaft, steigend. Liegt nur eine kleine Datenbasis vor, scheint der kollaborative Ansatz ein guter Kompromiss zwischen Genauigkeit und Vielfalt zu sein. Ein ähnliches Verhalten zeigt sich auch bei der Catalog Coverage, also bei der systemweiten Diversity. Für eine kleine Nachbarschaft $k < 30$ können sogar bessere Ergebnisse durch das Reranking Verfahren im Verhältnis zum kollaborativen Filtern oder Clustering erreicht werden. Bei der globalen Novelty zeigen sich zumindest auch für die Reranking Ansätze eine Steigerung mit kleiner Nachbarschaft. Wenn die Novelty für einzelne Benutzer betrachtet wird, ist zu erkennen, dass sowohl das kollaborative Filtern, als auch die Reranking Ansätze, kaum Änderungen über die Anzahl der Nachbarn haben. Für eine größere Nachbarschaft ist anzunehmen, dass das Reranking Verfahren und das kollaborative Filtern ähnlich viel Novelty in die System einführt können. Wie auch für die anderen Eigenschaften, kann nur der Clustering Ansatz eine Steigerung der Novelty erreichen.

Die Serendipity kann, aufgrund der gemachten Annahmen, nur für die Ansätze zur Steigerung der Vielfalt betrachtet werden, da das kollaborative System zur Berechnung der Metrik Verwendung findet. Dadurch, dass der Clustering Ansatz eine sehr hohe Precision und die Greedy Selection eine so niedrige Precision haben, ist es schwierig die drei verbliebenen Verfahren zu vergleichen. Die Tendenz ist jedoch, dass für mehr Nachbarn die Serendipity in den Empfehlungen abnimmt.

In ihrer Gesamtheit zeigen die Ergebnisse, dass für spezifische Probleme in Empfehlungssystemen die Vielfalt, im Verhältnis zu traditionellen Verfahren, steigern lässt. Das ist zum Beispiel der Fall, wenn nur eine kleine Nachbarschaft, aufgrund einer kleinen Datenbasis, zum Einsatz kommen kann. Vor allem ist aber anzumerken, dass ein benutzerbasiertes kollaboratives System bereits vielfältige Empfehlungen erstellen kann. Diese Erkenntnis war auch die Grundlage, weshalb dieser Ansatz als Referenzsystem ausgewählt wurde. Bei den Reranking Ansätzen war zu erkennen, dass sich nicht alle Eigenschaft im gleichen Umfang verbessern ließen. Das Clustering Verfahren hat gezeigt, dass es möglich ist die Vielfalt bei einer größeren Nachbarschaft sogar zu steigern, ohne große Nachteile bei der Vorhersagegenauigkeit zu haben. Dieses Ergebnis ist keine Selbstverständlichkeit, da die Genauigkeit und die Vielfalt sich gegensätzlich zueinander verhalten.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Arbeit wurde in die aktuelle Entwicklung aus dem Bereich der Empfehlungssysteme präsentiert. Die klassischen Methoden zur Bewertung von Empfehlungssystem anhand ihrer Genauigkeit wurden dargestellt und anhand dieser argumentiert, wieso die Vielfalt der Empfehlungen von Interesse sind, um das Vertrauen der Benutzer in ein System zu stärken.

Die wichtigsten Ansätze zur Erstellung eines Empfehlungssystems wurden vorgestellt. Hierfür wurden die konkrete Umsetzung für traditionellen Ansätze in Kapitel 2 vorgestellt. Zudem wurden zwei Verfahren zur Umwandlung von impliziten Informationen in explizite Bewertungen beschrieben.

Anschließend wurden verschiedene Sichtweisen auf die Vielfalt in Empfehlungssystem diskutiert und zwischen globaler und benutzerorientierter Vielfalt unterschieden. Es wurden drei Eigenschaften von Empfehlungssystem herausgearbeitet, die die Vielfalt besonders gut abbilden. Diese Eigenschaften wurden formalisiert und Methoden zur Messung vorgestellt. Anschließend wurden Strategien entwickelt um die Vielfalt anhand dieser Eigenschaften in Empfehlungssystem zu steigern.

Um die Auswirkungen der Verfahren zur Steigerung der Vielfalt in Empfehlungssystem zu analysieren, wurde anhand eines existierenden Datensatzes die erstellten Empfehlungen evaluiert. Um eine Vergleichbarkeit zu gewährleisten wurde hierfür ein kollaboratives Empfehlungssystem, das anhand der Bewertungen anderer Benutzer die Empfehlungen erstellt, entwickelt.

Die Ergebnisse haben gezeigt, dass es anhand der definierten Metriken, möglich ist die Vielfalt in Empfehlungssystem zu steigern, ohne das die Genauigkeit dabei zu stark abnimmt. Außerdem konnte gezeigt werden, dass die Verfahren unterschiedlich starken Einfluss auf die Eigenschaften für Vielfalt haben, und das kein Goldstandard existiert, der alle Eigenschaften besonders gut abdeckt.

6.2 Ausblick

In dieser Arbeit konnte gezeigt werden, dass die Eigenschaft der Vielfalt eines Empfehlungssystem verbessert werden kann, ohne das die Vorhersagegenauigkeit darunter zu leiden hat. Da die Auffassung von Vielfalt auch ein individuelles Empfinden von Personen ist, sollten die Ergebnisse durch eine Benutzerstudie (auch Offline Evaluation) validiert werden, um so konkrete Aussagen zu den, in der Arbeit beschriebenen, Eigenschaften durch reale Personen zu erhalten. Darüber hinaus wäre

eine Untersuchung anderer Datensätzen interessant, auch aus anderen Wissensgebieten, um so zu der Erkenntnis zu kommen, ob die verwendeten Verfahren auch dort akzeptable Ergebnisse liefern. Zum Schluss wäre die einheitlichen Definition zur Messung der Vielfalt in Empfehlungssystem ein erstrebenswertes Ziel, um so die Vergleichbarkeit von Systemen zu vereinfachen.

Literaturverzeichnis

- [1] *Steady-State Properties of of GI/G/1*, pages 266–301. Springer New York, New York, NY, 2003. ISBN 978-0-387-21525-9. doi: 10.1007/0-387-21525-5_10. URL http://dx.doi.org/10.1007/0-387-21525-5_10.
- [2] Panagiotis Adamopoulos and Alexander Tuzhilin. On unexpectedness in recommender systems: Or how to better expect the unexpected. *ACM Trans. Intell. Syst. Technol.*, 5(4):54:1–54:32, December 2014. ISSN 2157-6904. doi: 10.1145/2559952. URL <http://doi.acm.org/10.1145/2559952>.
- [3] Gediminas Adomavicius and Y Kwon. Improving recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*.
- [4] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17(6):734–749, 2005.
- [5] Dhoha Almazro, Ghadeer Shahatah, Lamia Abdulkarim, Mona Kherees, Romy Martinez, and William Nzoukou. A survey paper on recommender systems. *arXiv preprint arXiv:1006.5278*, 2010.
- [6] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.
- [7] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98*, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-555-X. URL <http://dl.acm.org/citation.cfm?id=2074094.2074100>.
- [8] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.
- [9] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems, RecSys 2011*, New York, NY, USA, 2011. ACM.

-
- [10] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.
- [11] Pablo Castells, Neil J Hurley, and Saul Vargas. Novelty and diversity in recommender systems. In *Recommender Systems Handbook*, pages 881–918. Springer, 2015.
- [12] Òscar Celma and Perfecto Herrera. A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186. ACM, 2008.
- [13] Ricci Francesco, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Introduction to recommender systems handbook*. Springer, 2015.
- [14] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10*, pages 257–260, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-906-0. doi: 10.1145/1864708.1864761. URL <http://doi.acm.org/10.1145/1864708.1864761>.
- [15] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.
- [16] David Hauger, Markus Schedl, Andrej Kosir, and Marko Tkalcić. The million musical tweet dataset - what we can learn from microblogs. In *ISMIR*, 2013.
- [17] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.
- [18] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.
- [19] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2), April 2007. ISSN 1046-8188. doi: 10.1145/1229179.1229181. URL <http://doi.acm.org/10.1145/1229179.1229181>.
- [20] Marius Kaminskis and Derek Bridge. Measuring surprise in recommender systems. In *Proceedings of the Workshop on Recommender Systems Evaluation: Dimensions and Design (Workshop Programme of the 8th ACM Conference on Recommender Systems)*, 2014.

- [21] Peter A Lachenbruch and M Ray Mickey. Estimation of error rates in discriminant analysis. *Technometrics*, 10(1):1–11, 1968.
- [22] Ken Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95)*, 1995.
- [23] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10*, pages 210–217, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835486. URL <http://doi.acm.org/10.1145/1835449.1835486>.
- [24] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.
- [25] Michael I Mandel and Daniel PW Ellis. Song-level features and support vector machines for music classification. In *ISMIR 2005: 6th International Conference on Music Information Retrieval: Proceedings: Variation 2: Queen Mary, University of London & Goldsmiths College, University of London, 11-15 September, 2005*, pages 594–599. Queen Mary, University of London, 2005.
- [26] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems, CHI EA '06*, pages 1097–1101, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: 10.1145/1125451.1125659. URL <http://doi.acm.org/10.1145/1125451.1125659>.
- [27] Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. Metrics for evaluating the serendipity of recommendation lists. In *New frontiers in artificial intelligence*, pages 40–46. Springer, 2007.
- [28] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.
- [29] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Mach. Learn.*, 27(3):313–331, June 1997. ISSN 0885-6125. doi: 10.1023/A:1007369909943. URL <http://dx.doi.org/10.1023/A:1007369909943>.
- [30] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.
- [31] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In

- Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94*, pages 175–186, New York, NY, USA, 1994. ACM. ISBN 0-89791-689-1. doi: 10.1145/192844.192905. URL <http://doi.acm.org/10.1145/192844.192905>.
- [32] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.
- [33] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 0-201-12227-8.
- [34] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.
- [35] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-84705-1. doi: 10.1145/223904.223931. URL <http://dx.doi.org/10.1145/223904.223931>.
- [36] Barry Smyth and Paul McClave. Similarity vs. diversity. In *Case-Based Reasoning Research and Development*, pages 347–361. Springer, 2001.
- [37] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.
- [38] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.
- [39] Eva Zangerle, Martin Pichl, Wolfgang Gassler, and Günther Specht. #nowplaying music dataset: Extracting listening behavior from twitter. In *Proceedings of the First International Workshop on Internet-Scale Multimedia Management, WISMM '14*, pages 21–26, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3157-9. doi: 10.1145/2661714.2661719. URL <http://doi.acm.org/10.1145/2661714.2661719>.
- [40] Mi Zhang and Neil Hurley. Novel item recommendation by user profile partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '09*, pages 508–515, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3801-3. doi: 10.1109/WI-IAT.2009.85. URL <http://dx.doi.org/10.1109/WI-IAT.2009.85>.

- [41] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. Auralist: Introducing serendipity into music recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12*, pages 13–22, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-0747-5. doi: 10.1145/2124295.2124300. URL <http://doi.acm.org/10.1145/2124295.2124300>.
- [42] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM, 2005.
- [43] Òscar Celma (auth.). *Music Recommendation and Discovery: The Long Tail, Long Tail, and Long Play in the Digital Music Space*. Springer-Verlag Berlin Heidelberg, 1 edition, 2010. ISBN 3642132863,9783642132865.