UNIVERSITÄT REGENSBURG – PHILOSOPHISCHE FAKULTÄT III

Ein meta-hybrides Empfehlungssystem für Filme

Magisterarbeit
in der Philosophischen Fakultät III
(Sprach- und Literaturwissenschaften)
der Universität Regensburg

vorgelegt von:	Krasen Dimitrov
Adresse:	Boessnerstr. 9
	93049 Regensburg
Matrikelnummer:	1300640
Erstgutachter:	Prof. Dr. Christian Wolff
Zweitgutachter:	Prof. Dr. Rainer Hammwöhner
Laufendes Semester:	Wintersemester 2010/11
vorgelegt am:	8. Februar 2011

Zusammenfassung

Im Rahmen dieser Arbeit wurden die theoretischen Grundlagen und Ansätze der Empfehlungsgenerierung in Verbindung zu kleinen und mittelgroßen Filmplattformen behandelt. Durch den Entwurf eines Systemmodells für Empfehlungsgenerierung von Filmen am Beispiel von der Filmplattform *critic.de* wurde ein praktischer Bezug zu der daraus entstehenden Problematik hergestellt. Ein meta-hybrides System bestehend aus *content-based* und *collaborative filtering* wurde als Lösung für die Kaltstartproblematik, welche besonders Empfehlungssysteme mit wenigen Benutzern betrifft, vorgeschlagen. Die anschließende Implementierung des Filmähnlichkeitssystems validierte das vorgeschlagene Datenmodell zur Repräsentation von Filmen und den Ähnlichkeitsalgorithmus.

Abstract

This thesis discusses the theoretical groundwork and scientific approaches to recommender systems in the context of small and medium-sized movie portals. In the first part of this work the reader is presented with a brief overview of the said systems. In the second part through the design of a system model for movie recommendation on the basis of the movie portal *critic.de* a practical reference to the resulting problems could be established. A meta-hybrid system consisting of content-based and collaborative filtering is presented as a solution to the cold start problem, which most of all affects systems with a small number of users. The concluding implementation of the movie similarity subsystem validates the proposed data model for the representation of movies and the similarity algorithm.

Inhaltsverzeichnis

1 Einleitung

Die rasante Entwicklung des Internets hat einerseits zu einer Informationsflut und andererseits zu einer immer größer werdenden Menge an Nutzern geführt. Als Folge wurden die Information Retrieval-Systeme entwickelt, um diese Flut zu stemmen. Dennoch sind diese Systeme in vielen Bereichen nicht ausreichend. Man kann dies leicht in der Unterhaltungsbranche sehen, vor allem bei Filmportalen. Mehr als hundert Jahre Filmgeschichte haben eine nicht zu durchblickende Menge von kinematografischen Werken hervorgebracht. Die größte Filmdatenbank IMDB hat mittlerweile ca. 1,750,000 Einträge[1]. Klassische Information-Retrieval-Systeme können zwar diese Menge nach bestimmten Suchkriterien einschränken. Eine zentrale Frage bleibt bei diesen Systemen jedoch ungeklärt: sind die gefundenen Filme sehenswert? Das ist auch die Fragestellung die Empfehlungssysteme zu beantworten versuchen: Welche Filme kennt der Benutzer noch nicht und welche davon wären für Ihn interessant.

[1] s. IMDBs Datenbankstatistik. Letzter Zugriff: 04.01.2011, unter http://www.imdb.com/stats .

2 Ziel

Auf dem Gebiet der Filmempfehlungssysteme gibt es viele Lösungsansätze, vor allem haben *collaborative filtering* und *content-based filtering* viel Aufmerksamkeit bekommen, weil sie relativ weniger Aufwand im Vergleich zu *knowledge-based* und *demographiic filtering* erfordern, um an das gewünschte Ergebnis zu kommen. Viele Ansätze sind auf der Basis der großen Datasets von Netflix[2] und MovieLens[3] aufgebaut worden, weil sie strukturiert vorliegen und sehr viele Daten in Form von Bewertungen enthalten. Diese zwei Empfehlungssysteme von MovieLens und Netflix sind ein Teil ihres gesamten Erfolges[4]. Aber sie besitzen als Hauptbestandteil *collaborative filtering*, was den Einsatz von vergleichbaren Modellen auf kleinen und mittelgroßen Domänen erschweren wird. Der Grund dafür ist, dass sie eine große Anzahl an Nutzern und Nutzerbewertungen schwerer und langsamer akkumulieren können.

Ziel dieser Arbeit ist es ein Modell eines Empfehlungssystems vorzustellen, das auf kleinen und mittelgroßen Domänen erfolgreich eingesetzt werden kann und den Herausforderungen so eines Einsatzes gewachsen ist.

Als Grundlage wird die Webseite *critic.de* und deren Filmdatenbank verwendet.

---

[2] Im Jahre 2006 wurde die erste *Netflix Prize* Wettbewerb angekündigt. Ziel war ein Empfehlungsprogramm zu entwickeln, welches die Filmpräferenzen von Benutzern voraussagen und das bereits existierende System *Cinematch* um 10 % übertreffen kann. Das Dataset besteht aus 100,480,507 Bewertungen von 480,189 Benutzern über 17,770 Filmen. Jede Bewertung besteht aus dem Quadrupel <*user, movie, date of grade, grade*>. Gewinner war ein Team, das sich aus dem AT&T Forschungsteam BellKor und anderen gebildet hatte. Der Gewinner wurde mit 1 Million US-Dollar ausgezeichnet. Der Algorithmus heißt *Pragmatic Chaos*, eine Art maschinelles Lernen. Um das Endergebnis zu generieren werden die Ergebnisse von mehr als 100 Algorithmen und Modellen zusammengeführt (Siehe: The BellKor 2008 Solution to the Netflix Prize, S. 1). Sie könnten damit feststellen, dass Benutzer ältere Filme ganz anders als neue, gerade gesehene Filme bewerten und dass außerdem Bewertungen vom Wochentag oder der Laune der Benutzer abhängig sind.

[3] Es sind drei verschiedene Datasets von MovieLens verfügbar. Das größte enthält 10 Millionen Bewertungen von 71567 Benutzern und 100.000 Tags für 10681 Filme. Letzter Zugriff: 04.01.2011, unter http://www.grouplens.org/node/73.

[4] Kabelfernsehen Anbieter stehen mittlerweile unter Zugzwang, um mit Netflix zu konkurrieren. Quelle: *Tough choices for studios as Netflix surges*. Letzter Zugriff: 01.02.2011, unter http://www.marketwatch.com/story/tough-choices-for-studios-as-netflix-surges-2011-01-27.

3 Die Filmplattform *critic.de* und deren Filmdatenbank

Die Seite befindet sich seit 01.09.2004 in Betrieb und hat aktuell ca. 180.000 Besucher pro Monat bei 350.000 Seitenaufrufen. Die Datenbank enthält mehr als 2.000 komplette Filmeinträge (Kritik, Bildergalerie, Stab, Besetzung etc.). Im Moment gibt es keine Benutzerprofile und daher keine Benutzerdaten, die für ein Empfehlungssystem relevant sind. Das Content Management System ist TYPO3 mit einer Datenverwaltung über MySQL. Die Seite wurde Anfang 2010 überarbeitet. Die aktualisierte Version der Seite steht seit Mitte Mai zur Verfügung. Zurzeit gibt es keine Bewertungs- oder Empfehlungssysteme. Empfehlungen sind nur redaktionell möglich und Bewertungen erfolgen nur in Form von Kommentaren. Somit bietet die Seite alle Möglichkeiten und Herausforderungen zur Entwicklung des gewünschten Empfehlungssystems.

3.1 Die vorhandenen Filmdaten und die Ausgangslage

In diesem Kapitel wird der Istzustand der Seite zum Beginn dieser Arbeit vorgestellt, um die Möglichkeiten und Schwierigkeiten für die Einführung eines Filmempfehlungssystems zu verdeutlichen.

Die Filmdatenbank enthält mehr als 2.000 komplette Einträge. Die Filme decken eine Zeitspanne der letzten 80 Jahre der Filmgeschichte ab und repräsentieren viele Filmgenres und 84 Herstellungsländer. Zur Datenbank gehören auch ca. 15.000 Personen, welche in diesen Filmen mitgewirkt haben (z. B. als Schauspieler, Produzenten etc.). Die Filmdatenbank besitzt (Stand 2010) folgende Beschreibungsmerkmale für Filme: Originaltitel, deutscher Titel, Alternativtitel, Herstellungsland, Jahr der Fertigstellung, Spieldauer, Altersfreigabe, Regie, Drehbuchautoren, Originalautor, Produktion, Darsteller, Kamera, Musik, Schnitt, Stimme und Synchronstimme. In Tabelle 1 kann man einen kleinen Ausschnitt aus einem Datensatz der wichtigsten Merkmale von elf Filmen sehen und wie sie im MySQL dargestellt werden[5].

[5] An diesem Beispiel kann man auch einen Nachteil von TYPO3, was den Umgang mit der MySQL DB betrifft, sehen, der die tatsächliche Erstellung der Empfehlungsalgorithmen erschweren kann. Denn Verbindungen innerhalb von TYPO3 werden in Form von Zahlenwerten dargestellt, welche die jeweilige UID (Unique ID) repräsentieren, und dann werden sie in BLOB (Binary Large Object) Feldern gespeichert, deren Bearbeitung

Deutscher Titel	Land	Jahr	Dauer	Regie	Drehbuch	Produzent	Darsteller
Rapunzel - Neu verföhnt	2	2010	91	14807	14808,14809		3612,8006, …
Lügen macht erfinderisch	2	2009	100	11167	11167	13203	11167,2914, …
Die etwas anderen Cops	2	2010	98	40	40,14830	4932,3012	4151,6562, …
Mahler auf der Couch	89	2010	105	14843,14844	14843,14844		14846,3409, …
The Night Chronicles: Devil	2	2011	130	9584,9583	859		12106,13557, …
The American	2	2010	121	7532	5521	2157,2268,414	2157,14870, …
Zwischen uns das Paradies	56,18,1	2010	100	2800	2800	606,3406,9623	14884,3761, …
Mammuth	3	2010	92	688,2313	2313,688	2878,6466	2184,6660, …
The Town	2	2010	85	656	14903,656		656,14904, …
Twelve	2,3	2010	93	3063	14905		14906,10201, …
Machete	2	2010	101	5419	5419	5192,5419	14653,10795, …

Tabelle 1: Ein Ausschnitt aus der Filmdatenbank

Wie bereits erwähnt verfügt die Seite über keine Bewertungsmöglichkeiten. Es ist nur möglich, redaktionell einen Film als „Empfehlung“ zu markieren. Zudem existieren keine Benutzerprofile, welche für Empfehlungssysteme unentbehrlich sind. Die Benutzer können mit der Seite und miteinander interagieren, indem sie Beiträge zu Filmen kommentieren. Diese Kommentare können auch nicht bewertet werden. Sortierungsmöglichkeit und Navigation sind nur durch die Suchfunktion (alle Texte und Inhalte sind komplett auf der Basis von ganzen Wörtern indexiert), durch eine alphabetische Auflistung oder eine Auflistung nach Starttermin möglich.

unnötig aufwendig ist, da sie für andere Zwecke wie Transfer und Speicherung von großen Datenmengen in binärer Form, bestimmt sind.

3.2 Anforderungen an das Empfehlungssystem

In den ersten Gesprächen mit dem Betreiber der Seite wurde das Hauptziel des Systems festgelegt. Dieses soll dem Nutzer der Seite angesichts des schnell steigenden Angebots an Filmkritiken bei der Navigation helfen. Das soll gelingen, indem Empfehlungen, die für jeden Nutzer individuell gestaltet sind, generiert werden. Als Grundlage für das System wird zusätzlich eine Ähnlichkeitsanalyse von Filmen vorausgesetzt, welche erweiterte Möglichkeiten zur Auswahl und zum Überblick über die Filmkritiken schaffen soll. Damit können auch die internen Verlinkungen innerhalb der Artikel der Seite verbessert werden. Als Basis für diese Ähnlichkeitsanalyse soll ein System von Tags und Genres dienen, welches noch eingeführt werden soll.

Die persönlichen Empfehlungen und die Ähnlichkeitsanalyse benötigen weiterhin Benutzerprofile, in denen die Bewertungen über Filmmerkmale und Präferenzen gespeichert werden können. Diese müssen noch im System implementiert werden.

Dabei soll die Identität der Seite, als Portal für Fachkritiken und Interviews, nicht verletzt werden. Dieses Empfehlungssystem soll nicht die redaktionellen Empfehlungen ersetzen, sondern nur eine weitere Option für die Nutzer sein. Empfehlungen und Bewertungen der einzelnen Nutzer sollen darüber hinaus nur jeweils für sie selbst sichtbar sein und nicht mit den Kritiken als Gütekriterium in Verbindung gesetzt werden. Damit wird beabsichtigt, dass die Benutzer der Seite nicht zu „Kritikern“ werden, um den damit verbundenen Administrationsaufwand zu vermeiden und die Seite von den restlichen kollaborativen Filmempfehlungsseiten zu differenzieren. Dies bedeutet in der Praxis, dass Benutzerprofile für andere Nutzer nicht sichtbar sind.

Eine weitere Voraussetzung für eine Implementierung auf kleineren und mittelgroßen Domänen ist, dass das Empfehlungssystem nicht die Seite spürbar in ihrer Ladezeit beeinträchtigt, da diese über begrenzte Serverressourcen verfügen. Dies bedeutet, dass das System *server-based* laufen soll und nicht *client-based*, um Wartezeiten bei den Nutzeranfragen zu minimieren, da die Anzahl der Vergleichsoperationen sehr rechenintensiv ist. Dies verhindert weiterhin die Entstehung von Kompatibilitätsproblemen von *client-based* Skripten und die damit verbundenen Sorgen um Sicherheitslücken und Privatsphäre.

Außerdem müssen für die das System notwendigen Berechnungen, soweit es möglich ist, *offline* erfolgen, damit die Seite nicht verlangsamt wird. Diese Berechnungen werden durch die Ähnlichkeitsanalyse von Filmen und Benutzern erfolgen. Dies alles muss mit dem möglichst geringsten Rechenaufwand vollbracht werden, da diese Seite nur über einen Server verfügt. Das System soll außerdem leicht zu warten sein. Dabei soll es auf Veränderungen in der Datenbank reagieren und sich selbst aktualisieren. Fehlende oder falsche Eingaben im System sollen dieses nicht beeinträchtigen.

Anforderungen an das Empfehlungssystem:

- Bessere Navigation soll erreicht werden durch:
    - eine erweiterte Möglichkeit zur Auswahl und Überblick durch Metadaten
    - die Filmähnlichkeitsanalyse auf der Basis von Metadaten
    - die dazugehörige Funktion „Ähnliche Filme“
- die Generierung von möglichst genauen Filmempfehlungen auf der Basis von Benutzerpräferenzen
- Metadaten, wie Filmgenres, Tags etc. sind nur redaktionell zu ändern, um Administrationsaufwand zu vermeiden.
- die Einführung von Benutzerprofilen zur Speicherung von Benutzerpräferenzen mit folgenden Einschränkungen:
    - Profile sind privat. Bewertungen bleiben nur für den einzelnen Nutzer sichtbar, um die Privatsphäre zu schützen.
    - Benutzerbewertungen sollen nicht als Gütekriterium für Kritiken erscheinen.
    - Das System soll die Ähnlichkeit von Benutzerpräferenzen ermitteln und diese benutzen, um Empfehlungen zu generieren.
- serverseitiges System um Kompatibilitäts- und Sicherheitsprobleme zu vermeiden. Dies soll zusätzlich ermöglichen, dass keine Wartezeiten für den Benutzer entstehen.
- Das System soll möglichst autonom, ohne fremdes Einwirken und ohne die Seite zu verlangsamen arbeiten. Das erfordert O*ffline*-Berechnungen und eine automatische Aktualisierung der Benutzer- und Filmprofile.

- Die Eigenschaften der Seite, wie die ansteigende aber noch nicht große Benutzeranzahl oder die begrenzten System- und Personalressourcen, sollen berücksichtigt werden.

Die Filmdatenbank von *critic.de* hat zum Zeitpunkt der Implementierung einen großen Nachteil gehabt, dass sie nämlich keine Genres oder Tags als Filmmerkmale unterstützte. Da diese unentbehrlich für das vorgestellte Empfehlungssystem sind, müssen diese am Anfang dieser Arbeit eingeführt werden. Die spezifischen Genres und Tags sind nicht der Hauptfokus dieser Arbeit und deshalb werden sie nicht detailliert besprochen.

Der Prozess der Einführung verläuft in zwei Phasen. Erstens wird eine Liste von Genres und eine Liste von Tags erstellt, welche in der zweiten Phase den Autoren gegeben wird, um Genres und Tags für die Filme auszuwählen. Es wird großer Wert darauf gelegt, dass diese Genres und Tags sowohl dem Stand der Filmwissenschaft als auch der Online-Film-Community entsprechen. Die erste Anforderung wurde erfüllt, indem relevante Literatur und Personen mit filmwissenschaftlicher Fachausbildung zu Rat gezogen wurden (die Chefredakteure der Seite). Die zweite Anforderung wurde erfüllt, indem die Tags und Genres von großen Filmseiten im englischsprachigen und deutschsprachigen Raum (Netflix, IMDB und *moviepilot.de*) zum Vergleich genommen wurden.

3.3 Bemerkungen zu Filmgenres in der Filmwissenschaft und den Filmplattformen

Filmgenres sind nichts Festgeschriebenes und sind zudem schwer voneinander abzugrenzen. Daher besteht auch keine Einigkeit darüber, welche „die Filmgenres“ sind. In der Filmwissenschaft sind sogar ihre Existenz, Ursprünge und Brauchbarkeit umstritten:

„*Genre is a term much employed in film criticism at the moment, yet there is little agreement on what exactly it means or whether the term has any use at all. There appear to be three sorts of questions one could profitably ask: first, do genres in the cinema really exist, and if so can they be defined; secondly, what are the functions they fulfill? And third, how do specific genres originate or what causes them*?” (Buscombe 2003, S. 12)

Buscombe stellt auch den wichtigsten Aspekt von Genres in der Filmindustrie dar, nämlich dass Filmgenres Konventionen und eine Erwartungskonformität mit sich bringen. Dieses Wiedererkennen von gemeinsamen Merkmalen und Neuerungen in Filmen definiert das Filmgenre und ermöglicht den Menschen, die Filmen zu folgen und die Ideen von Regisseur und Drehbuchautor in einem Kontext zu setzen, was sie zu dem wichtigsten Beschreibungsmerkmal von Filmen macht:

"it is vital to see how icons relate to the cinema in general, but to genres in particular, and how in the popular cinema they may be reconciled to our natural desire to see films as the expression of an artistic personality. This can best be done through the notion that a genre film depends on a combination of novelty and familiarity. The conventions of the genre are known and recognized by the audience, and such recognition is in itself a pleasure." (Buscombe 2003, S. 22)

Im *Film Genre Reader 3* (Grant 2003) werden die in Tabelle 2 aufgeführten 20 Filmgenres ausgiebig besprochen und als solche im Inhaltsverzeichnis angegeben.

Action/Adventure Films	*Epic Films*	*Musical Films*	*War Films*
Biopics	*Exploitation Films*	*Road Movies*	*Western Films*
Comedy Films	*Film Noir*	*Sci Fi*	
Crime Films	*Gangster Films*	*Sports Films*	
Cult Movies	*Horror Films*	*Teenpics*	
Disaster Films	*Melodrama*	*Thrillers*	

Tabelle 2: Filmgenres in Film Genre Reader 3 (vgl. Grant 2003)

Dabei verhalten sich die meisten Filmempfehlungsseiten recht frei und begrenzen sich nicht auf die in der Filmwissenschaft vorgeschlagenen Filmgenres. Eine Ausnahme bildet IMDB, da dort die Genreauswahl mit nur 26 Filmgenres, welche in Tabelle 3 dargestellt sind, als eher konservativ zu bezeichnen ist. Dies lässt sich durch die zweitrangige Funktion des Filmempfehlungssystems (eher ein Filmähnlichkeitssystem) von IMDB erklären.

Action	*Adventure*	*Animation*	*Biography*
Comedy	*Crime*	*Documentary*	*Drama*
Family	*Fantasy*	*Film-Noir*	*Game-Show*
History	*Horror*	*Music*	*Musical*
Mystery	*News*	*Reality-TV*	*Romance*
Sci-Fi	*Sport*	*Talk-Show*	*Thriller*
War	*Western*		

Tabelle 3: IMDB Filmgenres (Quelle: IMDB Genreauflistung. Letzter Zugriff: 03.01.2011, unter http://www.imdb.com/genre)

Netflix beschränkt sich auf 19 Hauptfilmgenres, diese sind in Tabelle 4 aufgelistet, und enthält dazu Unterkategorien von Genres und Tags, deren Anzahl 300 übersteigt. Grund dafür ist erstens die große Anzahl von Filmen und zweitens die zentrale Rolle die das Empfehlungssystem bei Netflix einnimmt.

Action & Adventure	*Classics*	*Thrillers*	*Comedy*
Children & Family	*Drama*	*Faith & Spirituality*	*Foreign*
Documentary	*Gay & Lesbian*	*Horror*	*Television*
Independent	*Music & Musicals*	*Romance*	*Anime & Animation*
Sci-Fi & Fantasy	*Special Interest*	*Sports & Fitness*	

Tabelle 4: Hauptfilmgenres bei Netflix (Quelle: Netflix Genreauflistung. Letzter Zugriff: 03.01.2011, unter http://www.netflix.com/AllGenresList)

Moviepilot.de hat ein System ausgewählt, bei dem Benutzer die Filmgenres selber durch ein Votingsystem auswählen und neu einfügen können, und daher ist die Anzahl an Genres und Tags zu groß, um sie hier darzustellen.

Am Ende wurden folgende zwei Listen, welche in Tabellen 5 und 6 dargestellt sind, als endgültige Auswahl für das Genre- und Tagging-System festgelegt.

Abenteuerfilm	Erotikfilm	Liebesfilm	Satanismus
Action-Superheld	Erwachsenwerden	Mafiafilm	Satire
Actionfilm	Essay-Film	Martial-Arts-Film	Schicksalsdrama
Agentenfilm	Exorzismus	Mediensatire	Schwarze Komödie
Animationsfilm	Familiendrama	Melodram	Science Fiction-Film
Anime	Familienkomödie	Mockumentary	Screwball Comedy
Anti-Kriegsfilm	Fantasyfilm	Monsterfilm	Situationskomödie
Apokalypse & Postapokalypse	Gangsterfilm	Musikfilm	Slapstickkomödie

Biopic	Gaunerkomödie	Mystery	Slasherfilm
Blaxploitation	Gerichtsfilm	Naturdokumentation	Sozialdrama
Buddy-Film	Heimatfilm	Paranoia	Splatterfilm / Gorefilm
Coming-of-Age-Film	Heist	Parodie	Sportfilm
Computeranimation	Highschool Komödie	Piratenfilm	Superheldenfilm
Detektivfilm	Historical Fantasy	Polizeifilm	Tanzfilm
Doku-Drama	Historienfilm	Psychothriller	Thriller
Dokumentarfilm	Horrorfilm	Road Movie	Tragikomödie
Drama	Katastrophenfilm	Romantic Comedy	Utopie & Dystopie
Eastern	Komödie	Romantische Komödie	Western
Ehedrama	Kriegsfilm	Romanze	Zeichentrickfilm
Endzeitfilm	Kriminalfilm	Sandalenfilm	Zombiefilm

Tabelle 5: Die 80 Filmgenres in der Datenbank

Avantgarde	Expressionismus	Literaturverfilmung	Selbstfindungsdrama
Berliner Schule	Film Noir	Musical	Sequel
Comicadaption	Gefängnisfilm	New Hollywood	Serienmörderfilm
Debüt	Groteske	Nouvelle Vague	Stop Motion
Dogma	Gute-Laune-Film	Performance	Tragödie
Drogenfilm	IRA	Period Picture	Transzendentalismus
Episodenfilm	Kinderfilm	Prequel	Vampirfilm
Epos	Kitsch	Puppenfilm	Yakuza-Film
Experimentalfilm	Kompilationsfilm	Queer Cinema	
Exploitation	Liebesdrama	Remake	

Tabelle 6: Die 38 Tags in der Datenbank

Nachdem diese zwei Listen erstellt wurden, wurden sie an die Autoren aller Kritiken weitergeleitet und diese gebeten, Tags und Genres aus diesen Listen für die jeweiligen Filme zu vergeben. Die Ergebnisse wurden dann, wenn sie von den Listen abwichen, vereinheitlicht (z. B. Kriegsdrama, wurde in die Genres Kriegsfilm und Drama aufgeteilt), bevor sie im System eingegeben wurden.

Das Eingeben oder Entfernen von Genres und Tags ist nur redaktionell möglich. Den Nutzern der Seite wurde dies nicht erlaubt, um zusätzlichen Aufwand (Korrekturen von Vandalismus oder Fehleingaben) für die Redaktion zu vermeiden.

4 Verfahren der Empfehlungsgenerierung

Empfehlungssysteme sind personalisierte Informationssysteme, die Empfehlungen generieren. Diese Empfehlungen sind Vorschläge, die für den Nutzer relevant oder nützlich wären. Sie können sich auf Dokumente, Bücher oder andere Medien in einem Archiv oder einer Bibliothek beziehen, die für den Nutzer relevant sind. Auch im Bereich von E-Commerce finden solche Systeme Verwendung als ein erfolgreicher Teil großer Handelsplattformen wie *amazon.com* (vgl. Schafer et al. 2001, S. 5-6; Schafer et al. 2002, S. 44; Linden et al., 2003 S. 76-79). Im Grunde genommen sind diese Empfehlungssysteme Information-Retrieval-Systeme auf einer höheren semantischen Ebene. Als Ausgabe von Information-Retrieval-Systemen werden Treffer generiert, die der Benutzeranfrage entsprechen. Eine Empfehlung dagegen impliziert Nützlichkeit, Relevanz und sogar Qualität: Für den Nutzer lohnt es sich, den empfohlenen Gegenstand zu berücksichtigen. Empfehlungssysteme sind auch personalisierbar[6], indem sie sich an das Nutzerverhalten und die Bedürfnisse anpassen.

Das Empfehlungssystem erstellt ein Benutzerprofil auf Basis einer Lernmenge. Diese besteht aus allen Objekten, die der Nutzer positiv oder negativ bewertet hat. Danach soll das System bestimmen, welche noch nicht bewerteten Objekte für den Nutzer von Interesse wären. Die Objekte der Lernmenge werden formal mit Vektoren definiert, wobei die Vektorkomponenten nominal, binär oder numerisch sein können. Diese Komponenten werden entweder vom Inhalt der Objekte abgeleitet oder von Informationen über die Präferenzen des Nutzers. Auf diese Menge von Vektoren wird dann eine Funktion angewendet, dies jedes noch nicht bewertete Objekt als positiv oder negativ auswerten kann, wobei diese Funktion einen numerischen oder binären Wert zurückgibt. Dazu muss ein Schwellwert benutzt werden, um positive und negative Teilmengen zu bestimmen (vgl. Van Meteren, Van Someren 2000, S. 2-3).

[6] „Personalisierung“ ist mittlerweile auch ein Schlüsselwort für das sehr erfolgreiche Unternehmen Google. Die Treffergenerierung dort steigt über die üblichen Information-Retrieval-Methoden hinaus, indem persönliche Informationen, wie Surfverhalten und Standort berücksichtigt werden können (vgl. folgenden Artikel auf dem Googleblog. Letzter Zugriff: 10.09.2010, unter http://googleblog.blogspot.com/2009/12/personalized-search-for-everyone.html). PageRank kann auch als soziale Empfehlung interpretiert werden: Wer verlinkt zu dieser Seite? (impliziert Nützlichkeit und Relevanz). Auch die Funktion „Meinten Sie:“ ist eine Empfehlung. Die Möglichkeit für Relevance-Feedback, welches eine Art Benutzerbewertung innerhalb der Suche ermöglicht, macht Google zu einem noch rudimentären Empfehlungssystem (vgl. Burke 2007, S. 3).

Die größte Herausforderung für die Empfehlungssysteme ist das Kaltstartproblem (auch als "*ramp-up*" Problem bekannt [vgl. Konstan, et al. 1998]). Dieses besteht aus den *new user*- und *new item*-Problemen.

Das *new user*-Problem: Da Empfehlungen auf der Basis von Benutzerpräferenzen (z. B. im Form von Bewertungen) oder Benutzerverhalten (z. B. Browsergeschichte, vorherige Aktivitäten etc.) generiert werden, indem man diese mit den Daten von anderen Nutzern oder mit den Objektbeschreibungen vergleicht, kann man bei neuen Nutzern mit wenig oder gar keinen Bewertungen keine sinnvolle Empfehlung machen.
Das *new item*-Problem: Dementsprechend gibt es das *new item*-Problem, wenn ein neues Objekt im System hinzugefügt wird. Da es keine Bewertungen darüber gibt, kann es auch nicht und darf es nicht in der Empfehlungsgenerierung einbezogen werden. Eine Empfehlung wäre in diesem Fall nur dann möglich, wenn die inhaltsbeschreibenden Merkmale als Grundlage benutzt werden. Eine relativ geringe Anzahl an Bewertungen ist auch nachteilig, da sie zu Fehlempfehlungen führen kann. Deshalb versuchen alle Empfehlungssysteme das Bewerten zu stimulieren, indem sie eine Mindestanzahl an Bewertungen fordern, bevor Empfehlungen generiert werden können.

Für die Erstellung von Empfehlungssystemen sind zahlreiche Vorschläge gemacht worden. Die wichtigsten Verfahren der Empfehlungsgenerierung lassen sich nach Art der verwendeten Daten bzw. der Ähnlichkeitsberechnung unterscheiden. Burke unterscheidet folgende Arten anhand ihrer Informationsquelle (vgl. Abbildung 1): inhaltsbasierende (*content-based*), kollaborative (*collaborative*), demografische (*demographic*) und wissensbasierende (*knowledge-based*) Empfehlungssysteme (vgl. Burke 2007).

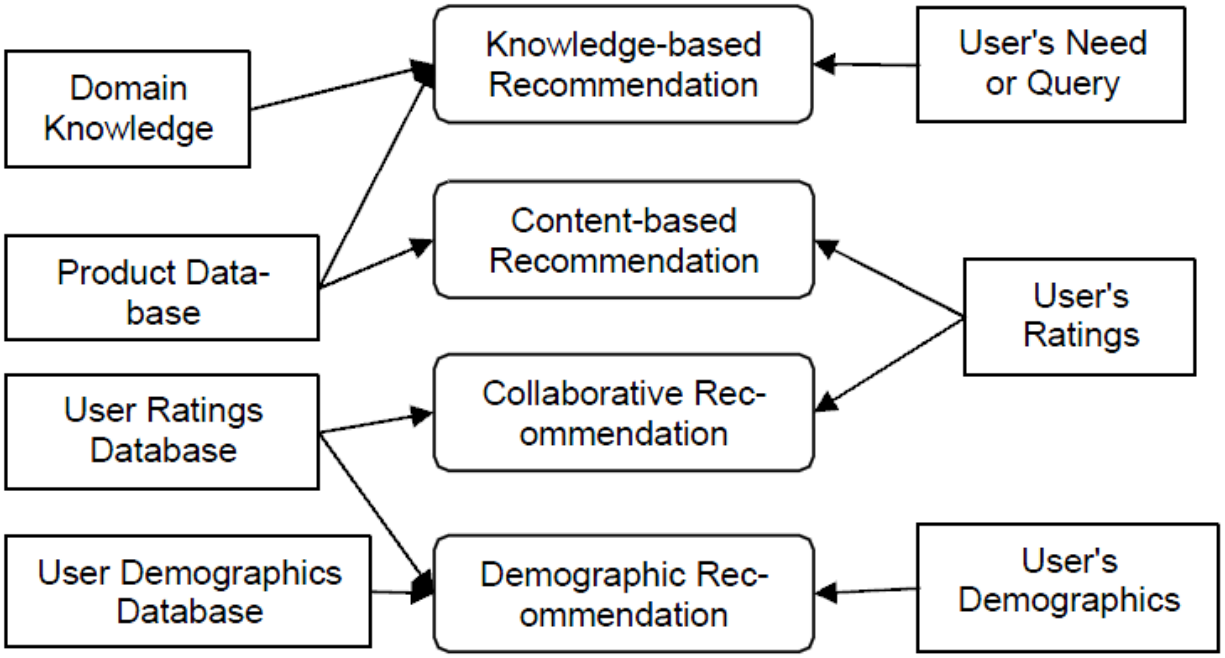


Abbildung 1: Techniken zur Empfehlungsgenerierung und deren Informationsquellen (siehe Burke 2007, S. 3)

4.1 Inhaltsbasierende Empfehlungssysteme

Beim inhaltsbasierten Filtern handelt es sich um eine Objekt-zu-Objekt-Korrelation ("*item-to-item correlation*" Schafer, Riedl 1999, S. 6). Dem Nutzer werden Objekte angeboten, die seinem Nutzerprofil zufolge passend wären. Dieses Wissen wird von den Profilen des einzelnen Nutzers abgeleitet und bezieht nicht – wie beim ko-operativen Filtern – die Profile und Bewertungen anderer Nutzer in die Berechnung ein. Die Empfehlungen werden auf der Basis von Objekten, die der Nutzer in der Vergangenheit bewertet hat und anhand von Objektmetadaten, welche die Objekte kennzeichnen (wie z. B. Genre bei Filmen, TF/IDF bei Dokumenten etc.), gegeben.
Benutzerprofile können implizit aufgebaut werden, indem man Daten über Aktionen und Verhalten sammelt oder explizit durch gezieltes Fragen (*Questionnaires*) oder durch das Bewerten.

4.1.1 Vorteile von inhaltsbasierenden Empfehlungssystemen

Die Kaltstartproblematik wird minimiert, da es kein *new item*-Problem gibt, nur das *new user*-Problem bleibt bestehen. Es existieren keine *privacy* Probleme, weil eigene Bewertungen und Vorlieben für andere Nutzer nicht sichtbar sind. Detaillierte Präferenzen von Nutzern können berücksichtigt werden.

4.1.2 Nachteile von inhaltsbasierenden Empfehlungssystemen

Content-based-Systeme sind durch die inhaltsbeschreibenden Merkmale der zu bewertenden Objekte eingeschränkt. Zum Beispiel kann ein inhaltsbasierendes Empfehlungssystem für Filme nur auf den Objektbeschreibungen beruhen, wie Filmgenre, Schauspieler, Regisseur etc., weil der Film an sich für das System nicht interpretierbar ist. Dies bedeutet, dass nur vollständig beschriebene Filme (und zwar alle Filme, die in die Berechnung einbezogen werden) zu einer erfolgreichen Empfehlung führen können. Man braucht auch genügend Deskriptoren, um Metabewertungen wie bei *collaborative filtering* zu vermeiden. Dazu kommt das Problem, dass nicht alle Aspekte eines Objektes formal beschrieben werden können. Im Falle eines Filmes wäre es fast unmöglich künstlerische Aspekte oder Emotionen als Inhaltsmerkmale wiederzugeben. Dies führt dazu, dass der „Filmgeschmack“ der Benutzer, der seine Präferenzen beeinflusst, nur schwer und nicht vollständig abgebildet werden kann. Zudem kann nicht jeder genau bestimmen, warum ihm ein Film gefallen oder missfallen hat. Das *new user*-Problem bleibt auch in seinem ganzen Umfang bestehen.

Content-based-Systeme tendieren dazu überspezifisch zu sein. Zhang behandelt die Problematik von Dokumenten, die zwar neu und relevant sind aber keine neue Information enthalten (vgl. Zhang et al. 2002). Betrachtet man Filmempfehlungen, kann man folgendes Problem feststellen: Wird ein Benutzerprofil auf der Basis von romantischen Filmen erstellt, ist es unmöglich andere Filme, die nicht dem identischen Muster entsprechen, zu empfehlen. Da die notwendigen Deskriptoren, wie Genre, Regisseur und Schauspieler innerhalb des Profils fehlen, werden keine abweichenden Filme empfohlen. Für den Benutzer bedeutet dies, dass er Filme außerhalb seiner bisherigen Vorlieben nicht vorgeschlagen bekommt, welche ihm vielleicht gefallen würden. So ein System kann auf Dauer uninteressant für den Nutzer werden, da es immer wieder die gleichen Filme vorschlagen wird.

4.2 Kollaborative Empfehlungssysteme

Beim *collaborative filtering* baut die Empfehlung auf benutzergenerierten Bewertungen auf. Man kann hier im Vergleich zu *content-based*-Filtering von einer Nutzer-zu-Nutzer-Korrelation sprechen. Über vergleichbare Interessen lassen sich Empfehlungen herleiten, da anzunehmen ist, dass Übereinstimmungen bei vor-

liegenden Bewertungen sich auf zusätzliche Objekte übertragen lassen (z. B. Amazon: „Kunden, die diesen Artikel erworben haben, haben auch jenen Artikel erworben“).

Basis sind Benutzerbewertungen auf verschiedene Gegenstände (z. B. Bücher, Filme etc.). Ein typischer Aufbau für ein solches System wäre, dass die Benutzerprofile als ein Vektor dargestellt werden, der die bewerteten Objekte und deren Bewertungen enthält. Dieser wird bei jeder weiteren Interaktion mit dem System aktualisiert. Man kann anhand dieser Vektoren Ähnlichkeiten bei den Vorlieben von Nutzern ermitteln und Nachbarschaften von Nutzern (*neighbourhoods*) berechnen. Umgekehrt kann man eine Ähnlichkeit bei den Bewertungen der Gegenstände finden und Nachbarschaften davon bilden. Die Bewertung eines Objektes kann auf der Basis der Bewertungen, die von den Benutzern aus der Nachbarschaft des aktuellen Nutzers gegeben wurden, bestimmt werden. Zudem kann sie von den Bewertungen der Objekte aus der Nachbarschaft des zu vorhersagenden Objekts bestimmt werden. Es gibt drei Methoden von *collaborative filtering*: die *user-based*, die *item-based* und die *model-based* Methode (vgl. Candillier et al. 2007, S. 548-549). Hier eine kurze Zusammenfassung:

- *User-based*

Um einen Gegenstand zu empfehlen, werden die Bewertungen der Benutzer, in deren Nachbarschaft sich der aktuelle Nutzer befindet, berücksichtigt. Dabei ist es notwendig, ein Ähnlichkeitsmaß zu definieren, bevor man die Nachbarschaften bilden kann. Auch die Objektbewertungen durch die benachbarten Nutzer müssen sinnvoll kombiniert werden.

- *Item-based*

Zuerst werden Nachbarschaften von allen Gegenständen anhand von gewissen Ähnlichkeiten (Bewertungen) ermittelt. Danach wird eine Empfehlung anhand der Bewertungen des aktuellen Nutzers bezüglich der Gegenstände aus der Nachbarschaft erstellt.

- *Model-based*

User-based und *item-based collaborative filtering* haben als Nachteil den sehr großen Rechenaufwand, der zu Zeitverzögerungen führen kann, wenn die Berechnungen nicht *offline* stattfinden. Daher versucht man diese Berechnungen mit Modellen zu generieren, um Cluster zu bilden. Jeder Cluster wird durch sein Zentrum bezeichnet. Es ist möglich Cluster von Nutzern und Objekten zu bilden. Danach kann man für die Bewertung des zu vorhersagenden Objekts die Bewertung des am nächsten gelegenen Zentrums heranziehen. Weiterhin kann man Wahrscheinlichkeitsalgorithmen, Ontologien (Supercluster), Bayes-Modelle etc. benutzen.

4.2.1 Vorteile von kollaborativen Empfehlungssystemen

Der größte Vorteil von *collaborative filtering*-Systemen ist, dass sie keinen Bezug zum Inhalt der Objekte brauchen. Dies bedeutet, dass diese von jeglicher Art von Informationen und Objektbeschreibungen komplett unabhängig sind. Man braucht nur einen Namen und die dazugehörigen Bewertungen, um das Objekt zu empfehlen. Im Vergleich dazu sind die *content-based*-Empfehlungssysteme zu hundert Prozent auf maschinell interpretierbare Informationen angewiesen. Dieser relativ kleine Aufwand im Vergleich zu anderen Empfehlungssystemen, um gute Empfehlungen zu generieren, hat dazu geführt, dass sie sehr verbreitet und erfolgreich sind (z. B. Amazon, Ebay, Netflix, Movielens, IMDB etc.).

Einer der wichtigsten Nachteile von *content-based*-Systemen existiert bei kollaborativen Systemen nicht, nämlich dass sie zu überspezifisch in ihren Empfehlungen sind. Man kann sogar vom Gegenteil sprechen. Da nur Bewertungen berücksichtigt werden und keine Inhalte, können Gegenstände empfohlen werden, die *per se* nicht zum Benutzerprofil zuordenbar wären. So ein System kann zum Beispiel einem Jazzliebhaber ein spezifisches Technolied vorschlagen, wenn genügend andere Benutzer ähnliche Vorlieben aufweisen und dieses Lied positiv bewertet haben. Dies macht den ganzen Vorgang der Empfehlungsgenerierung interessanter, weil man immer wieder neue und abwechslungsreiche Empfehlungen bekommen kann.

4.2.2 Nachteile von kollaborativen Empfehlungssystemen

Collaborative filtering-Systeme basieren im Grunde genommen auf statistischen Verfahren und brauchen eine gewisse Übereinstimmung bei den Bewertungen der Nutzer. Demzufolge verlangen sie eine sehr hohe Bewertungsdichte, um richtig zu funktionieren. Best-Case-Szenario wäre eine geringe und statische Menge von Objekten mit sehr vielen Benutzerbewertungen (z. B. sog. Blockbuster), Worst-Case-Szenario dagegen wäre eine große dynamische Menge von Objekten, bei denen eine große Übereinstimmung über viele Objekte unwahrscheinlich ist (z. B. Nachrichten).

Bei sehr dynamischen Objektmengen spielt sehr häufig das Alter der beinhaltenden Objekte eine tragende Rolle. Wenn diese Objekte nicht mehr aktuell sind, nutzen die bereits existierenden Bewertungen neuen Nutzern sehr wenig, weil diese Nutzer nicht auf der Basis von diesen alten Objekten mit den alten Nutzern verglichen werden können. Daher versuchen *collaborative filtering*-Systeme das Bewerten zu stimulieren, um eine bessere Bewertungsdichte zu erreichen. Die meisten ziehen erst Objekte zur Empfehlungsgenerierung ein, wenn eine kritische Menge von Bewertungen erreicht ist[7].

Kollaborative Empfehlungssysteme, welche auf einer *model-based*-Methode basieren, sind etwas weniger von dieser Problematik der Bewertungsdichte betroffen. Vor allem die Systeme, die mit Clusterbildung arbeiten, da man dort das Zentroid als bestimmend für den ganzen Cluster definieren kann.

Wie bereits erwähnt, gute Empfehlungen werden nur dann generiert, wenn viele Nutzer über die gleichen Objekte ähnliche Bewertungen abgegeben haben, d. h. wenn der Nutzer einer bestimmten Gruppe (z. B. Sci-Fi-Fan oder Reggaeliebhaber) gehört oder seine Vorlieben zu einer Nische passen. Wenn ein Nutzer nicht zu so einer Gruppe angehört, auch „*gray sheep*“ genannt (vgl. Claypool et al. 1999, S. 3), können keine akkuraten Empfehlungen gemacht werden, weil die Basis, nämlich ähnliche Bewertungen auf gleichen Objekten, fehlt. Nehmen wir an, er ist einer der wenigen Liebhaber von japanischen Filmen der 60er Jahren auf einer Seite, wo die

[7] IMDB braucht zum Beispiel fünf Filmbewertungen um eine Bewertung überhaupt anzuzeigen. Movielens dagegen braucht 15 Bewertungen des Nutzers um eine Empfehlung zu machen. Amazon ist eher die Ausnahme, dort werden Bewertungen sofort nach der ersten Bewertung angezeigt. Dies kann zwar nicht repräsentativ sein, hat aber auch keinen Einfluss auf die kollaborative Empfehlungsgenerierung, da Amazon mit der Häufigkeit von angesehenen und gekauften Artikeln arbeitet.

meisten Nutzer relativ jung sind und sich nur für zeitgenössische Filme interessieren. Er wird schlechte oder gar keine Empfehlungen bekommen, wenn das System nur auf kollaborativen Techniken aufgebaut ist.

Diese Menge an persönlichen Präferenzen und bisherigem Kundenverhalten, die benötigt wird, kann *privacy*-Befürchtungen[8] bei den Nutzern auslösen. Vor allem wenn diese für andere Nutzer sichtbar sind, stellt dies auch ein Problem dar, das kollaborative Systeme bewältigen müssen.

Das *cold start*-Problem bei neuen Benutzern oder Gegenständen, betrifft kollaborative Systeme am meisten, da sie nur auf Bewertungen beruhen. Deshalb gibt es bei neuen Nutzern meistens eine empfohlene Lernphase, in der Nutzer aufgefordert werden, eine Menge von Objekten, mit welchen sie bereits Erfahrung haben, zu bewerten, um diese so schnell wie möglich zu klassifizieren[9].

4.3 Demografische Empfehlungssysteme

Demografische Empfehlungssysteme generieren Empfehlungen auf der Basis von den demografischen Informationen im Profil des Nutzers. Solche Informationen wären zum Beispiel: Alter, Geschlecht, Geburtsort, Wohnort, Beruf, Bildung etc. Hier handelt es sich auch, wie bei kollaborativen Systemen, um eine Nutzer-zu-Nutzer-Korrelation. Der Unterschied besteht in den verwendeten Daten, einerseits Benutzerbewertungen, andererseits demografische Daten.

4.3.1 Vorteile von demografischen Empfehlungssystemen

Diese demografischen Daten ermöglichen den demografischen Systemen auch ohne Benutzerempfehlungen zu arbeiten, was deren größter Vorteil ist. Kollaborative und *content-based*-Systeme können ohne diese nicht funktionieren. Man versucht anhand dieser persönlichen Daten jeden Benutzer einer Gruppe mit gemeinsamen Interessen zuzuordnen, eine Art Stereotypen zu bilden (z. B. Hippies, Sci-Fi-Fans, alleinstehende Mütter etc.), und darauf basierend Empfehlungen zu generieren. Diese Gruppen können manuell oder automatisch aufgebaut werden.

[8] Siehe s. 33.

[9] Ein Beispiel wäre *moviepilot.de*, wo Nutzer aufgefordert werden mindestens sieben Filme, alle sieben von verschiedenen, aber populären Filmgenres, zu bewerten.

Ähnlich wie bei kollaborativen Systemen, können demografische Systeme Empfehlungen generieren, die nicht zu dem Nutzerprofil (im Sinne von bereits bewerteten Objekten und nicht demografischen Informationen) passend wären, weil sie sich nicht auf den Inhalt der zu empfehlenden Objekte beziehen.

4.3.2 Nachteile von demografischen Empfehlungssystemen

Auch hier bestehen die *new user*- und *new item*-Problem, obwohl das *new user*-Problem einen anderen Hintergrund hat. Zwar sind Bewertungen nicht essenziell notwendig für die Funktionalität und müssen als solche nicht eingegeben werden, aber man braucht die demografischen Daten, ehe eine Empfehlung generiert werden kann. Dies kann besonders aufwendig für den Nutzer sein und abschreckend wirken, vor allem wenn keine Bewertungen benutzt werden, braucht man eine riesige Menge an Daten, um Personen richtig zu klassifizieren. Daher ist es sinnvoll möglichst viele Daten automatisch zu beziehen, zum Beispiel anhand von der IP-Adresse den Standort zu ermitteln, oder alle notwendigen Daten aus einem bereits existierenden Profil zu sammeln.

Da hier die Daten noch persönlicher als bei kollaborativen Systemen sind, ist deren Einsatz für die Masse der Benutzer eher unwahrscheinlich. Letztendlich sind die ausschlaggebenden Daten meistens die sensibelsten, zum Beispiel Einkommen, sozialer Status, Fähigkeiten etc.
Dennoch ist deren Einsatz innerhalb von geschlossenen Kreisen, wo alle diese Daten bereits vorhanden sind, wie etwa Großunternehmen, möglich und durchaus sinnvoll (sog. *human resource management* Systeme).

4.4 Wissensbasierende Empfehlungssysteme

Ein wissensbasierendes Empfehlungssystem generiert Empfehlungen, indem das Benutzerprofil, in Form von Präferenzen und Bedürfnissen, mit vorhandenen Möglichkeiten verglichen wird. Man versucht nicht ein allgemeines Benutzerprofil zu erstellen, sondern es wird die Nützlichkeit der Gegenstände für den aktuellen Nutzer berechnet. Anhand dieser Nützlichkeit verfügt so ein System über funktionales Wissen. Es kann bestimmen, inwiefern ein Objekt die Bedürfnisse des Nutzers erfüllen bzw. nicht erfüllen kann.

4.4.1 Vorteile von wissensbasierenden Empfehlungssystemen

Wissensbasierende Empfehlungssysteme verlassen sich nicht auf statistische Daten (Bewertungen etc.) und deshalb besitzen sie viele der Probleme von den vorher beschriebenen Systemen nicht. Man braucht keine hohe Bewertungsdichte, eigentlich sind keine Bewertungen notwendig, es gibt kein „*gray sheep*“ Problem, man muss nicht zu einer repräsentativen Gruppe von Nutzern gehören, um qualitative Empfehlungen zu bekommen.

Collaborative und *content-based*-Systeme besitzen den Nachteil, da sie auf Bewertungen basieren um ein Profil zu erstellen, dass sie sich nicht schnell genug anpassen können. Das Profil ist langfristig und als solches nicht flexibel, wenn es um plötzliche Veränderungen bei den Benutzerpräferenzen geht. Bewertungen können eine Rolle spielen, auch wenn sie nicht mehr relevant sind. Ein Beispiel wäre ein Empfehlungssystem für Nachrichten.

Nehmen wir an, dass ein Nutzer vorhat, in Urlaub zu fliegen und sehr aufmerksam alle Meldungen zu dem isländischen Vulkan verfolgt. Dies wird dazu führen, dass er auch nach dem Urlaub, wenn dies für ihn vollkommen irrelevant sein wird, Empfehlungen und Weiterleitungen zu Nachrichten über den Vulkan bekommen wird. Ein anderes Beispiel wäre, ein kulinarisches Empfehlungssystem, wird man plötzlich zum Vegetarier (z. B. wegen seiner Religion, Fastenzeit etc.), ist so ein System komplett nutzlos. Systeme, die auf statistischen Verfahren basieren, brauchen sehr lange Zeit, um sich an Veränderungen anzupassen. Um dies zu beschleunigen, können Zeitaspekte bei den Bewertungen berücksichtigt werden, sodass alte Bewertungen weniger relevant oder gänzlich entfallen würden.

Das Benutzerprofil muss bei wissensbasierten Systemen in so einem Fall im Gegensatz zu den vorher aufgeführten Systemen nicht wieder neu gelernt werden. Man muss zwar sein Profil manuell aktualisieren, aber die Veränderungen der Präferenzen sind sofort danach sichtbar.

Weiterhin können nicht objektbezogene Präferenzen, sofern man diese angibt, wie Lieferzeiten und Gewährleistung von Artikeln berücksichtigt werden.

4.4.2 Nachteile wissensbasierenden Empfehlungssystemen

Der größte Nachteil von wissensbasierenden Empfehlungssystemen ist, dass solche Systeme sehr aufwendig für den Nutzer sind. Man muss alle seine Präferenzen angeben, dazu auch sinnvoll gewichten, für jedes Objekt, das für ihn interessant ist. Bei Produkten mit wenig beschreibenden Informationen kann so etwas akzeptabel sein, aber bei komplexen Gegenständen, wie Filme, Musik oder Bücher wird es zu viel sein. Wenn man sein Profil von Rockmusik auf Techno umstellen will, weil man jetzt solche Musik hören mag, muss man immer wieder bei Null anfangen.

Dieses Wissen, auf welchem diese Systeme aufbauen, besteht laut Burke, aus drei Teilen:
„*Catalog knowledge*“ – Wissen über die Objekte und deren Merkmale. Als Beispiel wird das System Entree gegeben, das zum Beispiel wissen muss, dass thailändisches Essen eine Art von asiatischem Essen ist (vgl. Burke 2002, S. 335).
„*Functional knowledge*“ – Wissen über den Zusammenhang von Objekt und Bedürfnis. Beispiel: Entree muss wissen, dass ein Restaurant mit Blick aufs Meer, das Bedürfnis einer romantischen Umgebung erfüllen kann (vgl. ebd., S. 335-336).
„*User knowledge*“ – Wissen über den Benutzer. Dieses wird in Form von demografischen Informationen oder Informationen, die direkt auf die Empfehlungssuche bezogen sind, gebildet (vgl. ebd., S. 335-336).

4.5 Inhärente Probleme von Empfehlungssystemen

Hauptproblem ist, dass die Bewertungen zu allgemein sind (sog. Metabewertungen, z. B. von sehr schlecht bis sehr gut, von 0 bis 10 etc.). Dabei wird eine einzelne allgemeine Bewertung über das Objekt gegeben, diese soll gleichzeitig Qualität, Nützlichkeit und Relevanz für den Nutzer darstellen. Dieses Problem tritt besonders beim *collaborative filtering* auf, da keine zusätzlichen Merkmale berücksichtigt werden, außer den Bewertungen. Bei einfachen Objekten, die an sich nicht mit mehr als ein paar Merkmalen beschrieben werden können, wie zum Beispiel Autoteile, wo nur Qualität und Preis eine Rolle spielen, ist dies nicht problematisch. Komplexere Gegenstände verlangen auch detaillierte Möglichkeiten zur Bewertung. Wie soll der Benutzer auch seine Eindrücke von einem Produkt über eine einzelne Note verteilen? Bei Amazon kann man Produkte über eine Fünfsterneskala bewerten. Wenn

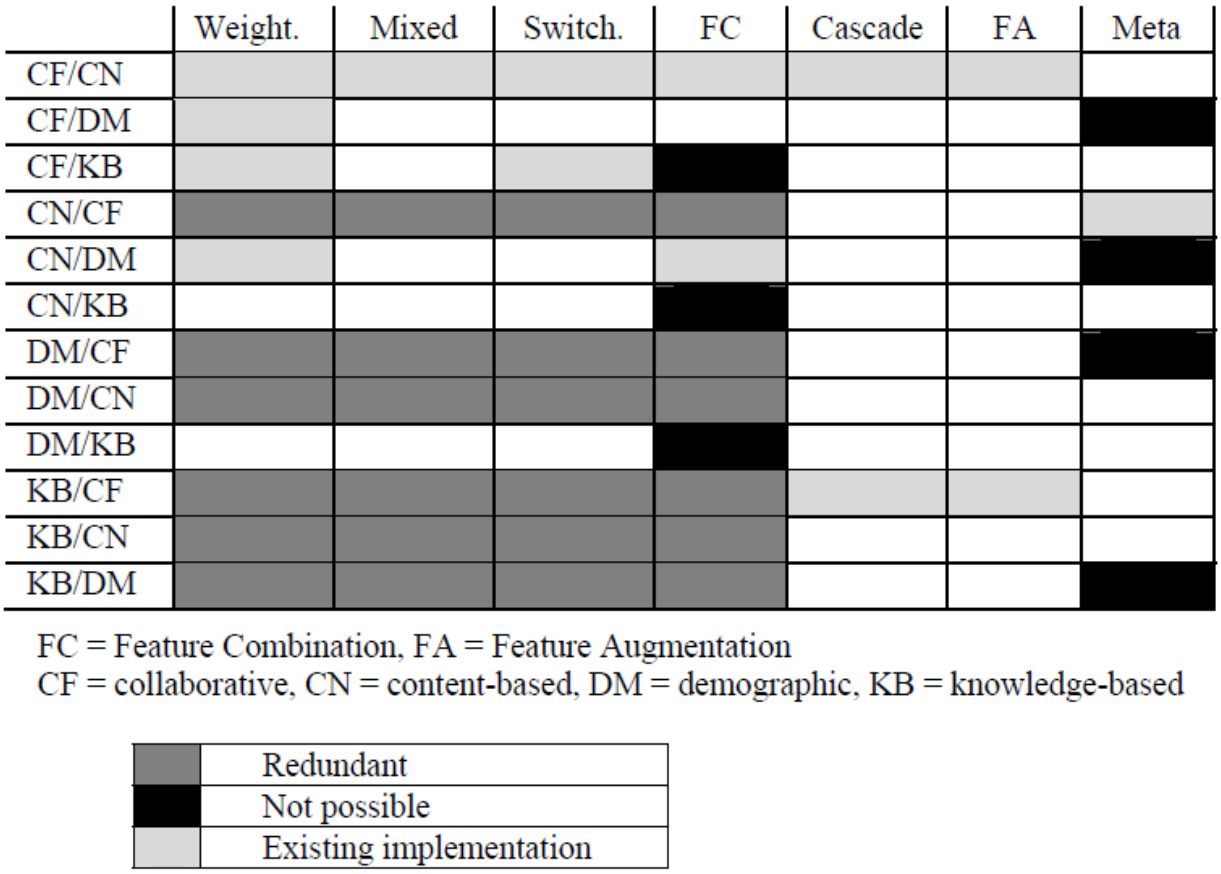
das Produkt ein komplettes PC-System von mehreren Komponenten ist, ist es sehr schwer, sinnvoll zu bewerten: Preis ist super, Rechner funktioniert, ist aber zu laut und das Gehäuse ist schlecht verarbeitet; dies mit fünf Sternen wiederzugeben ist kompliziert. Filme, Musik und Bücher sind noch schwieriger mit einer einzelnen Note abzubilden. Eine große Grundlage von Bewertungen minimiert das Problem, weil die Differenzen sich mit der steigenden Anzahl ausgleichen.

Dies führt zu Fehlinterpretationen. Zum Beispiel zwei Personen bewerten den Film Titanic mit 9, der eine hat 9 gegeben, weil er romantische Filme schätzt, der andere ist ein Fan von James Cameron und aufwendigen Effekten. Dennoch werden die beiden in der gleichen Nachbarschaft verortet, obwohl sie verschiedene Interessen haben. Die daraus folgenden Bewertungen können dementsprechend fehlerhaft sein. Dieses Problem kann durch das Einführen von mehreren Kriterien, welche das Objekt sinnvoller beschreiben und detaillierte Bewertungen zulassen, behoben werden (z. B. Bewertungen und Ähnlichkeitsanalyse erfolgen über mehrere Kriterien, wie Filmgenre, Regie, Oscarnominierungen, Hauptdarsteller etc.). So eine Implementierung geht über das normale *collaborative filtering* hinaus, weil objektbezogene Merkmale vorausgesetzt werden.

Daraus können weitere Probleme entstehen: ungewichtete Kriterien (alle Kriterien sind gleichbedeutend, was offensichtlich nicht korrekt ist). Wenn man diese gewichtet, bleibt das nächste Problem: Ist die Gewichtung richtig?

4.6 Hybride Empfehlungssysteme

Empfehlungssysteme in ihren einfachen Formen, wie im vorherigen Kapitel beschrieben, haben viele Nachteile. Das Ziel von hybriden Systemen ist es, verschiedene Modelle zusammen zu führen, um die Performanz zu steigern und die systemspezifischen und inhärenten Probleme von Empfehlungssystemen zu minimieren.



	Weight.	Mixed	Switch.	FC	Cascade	FA	Meta
CF/CN							
CF/DM							
CF/KB							
CN/CF							
CN/DM							
CN/KB							
DM/CF							
DM/CN							
DM/KB							
KB/CF							
KB/CN							
KB/DM							

FC = Feature Combination, FA = Feature Augmentation
CF = collaborative, CN = content-based, DM = demographic, KB = knowledge-based

	Redundant
	Not possible
	Existing implementation

Abbildung 2: Hybridisierungsmöglichkeiten nach Burke 2007, S. 5.

Burke unterscheidet insgesamt sieben verschiedene Möglichkeiten zur Kombinierung von Empfehlungssystemen:

- *Weighted*: Die Ergebnisse von zwei Empfehlungssystemen werden numerisch zusammengeführt.
- *Switching*: Das System wählt von verschiedenen vorhandenen Empfehlungskomponenten einen und benutzt diesen.
- *Mixed*: Empfehlungen, die von verschiedenen Systemen generiert wurden, werden gleichzeitig dargestellt.
- *Feature Combination*: Merkmale oder Kriterien von verschiedenen Wissensquellen werden kombiniert und als Input für einen einzelnen Empfehlungsalgorithmus benutzt.
- *Feature Augmentation*: Eine Empfehlungsmethode berechnet ein Merkmal oder eine Gruppe von Merkmalen und das Ergebnis wird als Eingabe für die nächste Empfehlungsmethode benutzt.

- *Cascade*: Empfehlungssysteme haben strikte Prioritäten, wobei diese mit niedrigerer Priorität die Verbindung zum Bewertungsprozess vor den Systemen mit höherer Priorität abbrechen.
- *Meta-level*: Die erste Empfehlungsmethode wird angewendet und generiert ein Modell. Dies wird danach als Input für die zweite Empfehlungsmethode benutzt.

Aus den verschiedenen Empfehlungsmethoden und den sieben Kombinationsmöglichkeiten kann man 53 verschiedene hybride Modelle von Empfehlungssystemen, welche auf Abbildung 2 zu sehen sind, zusammenstellen. Es sind so viele, weil die Modelle auch ordnungssensitiv sind (z. B. ein System, bei dem eine *content-based*-Methode auf einer kollaborativen Methode angewendet wird funktioniert anders als wenn eine kollaborative Methode auf einer *content-based* angewendet wird) (Vgl. Burke, 2007, S. 381-382.).

4.7 Zusammenfassung und Vergleich der verschiedenen Empfehlungssysteme

Alle Arten von Empfehlungssystemen haben ihre Vor- und Nachteile (siehe Vergleichstabelle, Tabelle 7). Dementsprechend sind sie nicht universal miteinander austauschbar und müssen gemäß ihren Stärken und Schwächen eingesetzt werden. In diesem Kapitel werden sie im Anbetracht von der Zieldomäne, nämlich der Filmseite *critic.de*, verglichen.

Ein Filmempfehlungssystem muss die Möglichkeit zur Bewertung und einen gewichteten Überblick (Ranking) für jeden Nutzer anbieten. *Collaborative filtering* als ein statisches Verfahren scheint auf dem ersten Blick die logische Lösung zu sein, da es nur die Bewertungen von den Nutzern braucht, um zu funktionieren. Mit steigender Anzahl von Bewertungen, steigt auch die Qualität, d. h. solche Systeme verbessern sich mit der Zeit, ohne dass fremdes Einwirken nötig ist. Die sich ständig ändernde Basis von Bewertungen trägt hinzu, dass die Empfehlungslisten dynamisch bleiben. Diese Vorteile haben zu ihrer großen Verbreitung, nicht nur auf dem Gebiet von Filmen (siehe Kap. 4,2) geführt. Ein Beispiel von *collabarative filtering* auf dem Gebiet von Filmempfehlungen im deutschsprachigen Raum ist *moviepilot.de.* Der Nutzer bekommt am Anfang eine zufallsgenerierte Liste von zehn Filmen und muss

diese bewerten, danach wird auf der Basis dieser Bewertungen eine Ähnlichkeit mit allen anderen Nutzern berechnet. Wenn man diese nicht gesehen hat und diese nicht bewerten kann, kann man weitere zehn Filme anfordern. Als Ergebnis bekommt man die Empfehlungsliste mit Filmen, die von den zehn am nächsten stehenden Benutzern abgeleitet wird und zwar von den Filmen, die sie positiv bewertet haben.

Dieser Vorgang demonstriert auch die größten Nachteile von *collaborative filtering*, nämlich das bereits erwähnte „*ramp-up*“ Problem. In diesem Beispiel versucht man das *new user*-Problem schnell und unaufdringlich für den Benutzer zu lösen, was sehr wichtig ist, immerhin erfordert nicht viel Zeit, zehn Filme zu bewerten. Die ersten Ergebnisse sind auf so kleiner Bewertungsbasis entsprechend mäßig und abhängig davon, ob ähnliche Nutzer, wie der neue Nutzer, bereits in der Datenbank existieren und ob diese zehn Bewertungen tatsächlich für die erfolgreiche Einordnung gereicht haben. Das *new item*-Problem hat als Folge, dass nicht bewertete Filme nicht für die Empfehlungsgenerierung herangezogen werden. Beide Probleme werden zusätzlich verstärkt, wenn es sich um kleine und mittlere Webseiten handelt, da sie Bewertungen langsam akkumulieren. Die kleine Anzahl von Film-Bewertungen und von aktiven Nutzern mit gut ausgefüllten Profilen senkt automatisch die Qualität der Bewertungen. Das „*gray sheep*“ Problem ist auch davon betroffen: Wenn der Nutzer einen sehr spezifischen Filmgeschmack hat, sind die Chancen, dass keine relevante Ähnlichkeit mit anderen Profilen zu finden ist, bei einer kleinen Benutzerbasis entsprechend hoch.

Content-based hat im Vergleich zu *collaborative filtering* den Vorteil, dass das *new item*-Problem minimiert wird (vgl. Lam et al. 2008, S. 208-211). Jedes Objekt kann anhand von seinen Beschreibungsmerkmalen beim Einführen in das System eingeordnet werden und bei der Empfehlungsgenerierung berücksichtigt werden. So ist das System nur auf die Bewertungen der aktuellen Nutzer über die Filme in der Datenbank angewiesen, das „*gray sheep*“ Problem existiert daher nicht.

Das *new user*-Problem bleibt zwar immer noch bestehen, dennoch wird es nicht zusätzlich durch eine kleine Anzahl von Nutzern verstärkt, d. h. sie können einen guten Einsatz bei allen Domänen finden.

Da auch hier das Nutzerprofil ständig durch zusätzliche Bewertungen bereichert wird, steigt die Qualität, wie bei *collaborative filtering*, mit der Zeit. Es bestehen auch keine *pivacy*-Bedenken, weil die Profile privat bleiben.

Der größte Vorteil bei solchen Systemen ist zugleich ihr größter Nachteil, nämlich die Objektbeschreibungen. Sie ermöglichen zwar detaillierte Präferenzen, Bewertungen, Sortierungs- und Empfehlungsmöglichkeiten, aber diese Objektbeschreibungen aufzubauen und instand zu halten erfordert deutlich mehr Aufwand als der Aufbau und Wartung von einem *collaborative filtering* System. Werden diese nämlich nicht vollständig und präzise gemacht, kann so ein System nur schlecht funktionieren. Zusätzlich ist die Auswahl der Beschreibungsmerkmale auch sehr wichtig und vor allem in dem Bereich von Filmen problematisch. Da diese Objektinformationen statisch sind, tendieren solche Systeme dazu überspezifisch zu sein, indem sie immer die gleichen Ergebnisse liefern.
Diese Nachteile haben dazugeführt, dass solche Systeme in ihrer reinsten Form nicht besonders verbreitet sind. Auf dem Bereich von Filmen gibt es relativ wenige kommerzielle Anwendungen, ein Beispiel wäre *moviefinderonline.com*, ein Filmkatalogisierungsprogramm, welches *content-based* und *collaborative filtering* getrennt voneinander anbietet. Auch IMDB bietet eine Art von Empfehlungen, auf der Basis vom Inhalt, in der Form von ähnlichen Filmen, welche keine echten nutzerbezogenen Empfehlungen sind.

Amazon.com benutzt zwar einen Teil davon, indem es Empfehlungen für Bücher und Filme auf der Basis der Beschreibungen von bereits erworbenen Artikeln beschränkt, aber zum größten Teil erfolgt die Generierung über *collaborative filtering*. Auf anderen Gebieten sind inhaltsbasierende Systeme zahlreicher vertreten: z. B. bei Nachrichten: *The DailyLearner system* (Billsus, Pazzani 2000) und *NewsWeeder* (Lang 1995).

Demographic filtering hat im Vergleich zu *content-based* und *collaborative filtering* die gleichen Vorteile wie die kollaborativen Systeme: Diese verbessern die gelieferten Ergebnisse mit der Nutzungszeit, haben auch keinen Bezug zum Inhalt der Gegenstände und sind dazu nicht überspezifisch und können Gegenstände außerhalb des tatsächlichen Benutzerprofils anbieten. Nur in zwei Punkten sind solche Systeme den

kollaborativen überlegen, nämlich dass keine Bewertungsvorgeschichte notwendig ist und dass die Anzahl der Bewertungen nicht zu einem besseren oder schlechteren Ergebnis führt. Die demografischen Daten sind ausreichend, um den Nutzer im System zuzuordnen.

Die Kaltstartproblematik bleibt dennoch bestehen, da man immer noch einen gewissen Input vom Nutzer braucht, um ihn mit den anderen Nutzern zu vergleichen. Es ist klar, dass demografische Daten einen sehr großen Vorteil mit sich bringen, vor allem für kleine und mittelgroße Domäne. Man braucht weniger Ressourcen im Vergleich zu inhaltsbasierenden Systemen um diese Daten instand zu halten, und da keine große Anzahl von Bewertungen nötig ist, kann so ein System auch auf der Basis von wenigen und nicht sehr aktiven Nutzern im Vergleich zu den kollaborativen gut funktionieren. Aber die Forderung nach diesen demografischen Daten schränkt auch den Einsatz von solchen Systemen ein.
Erstens erfordert ein vollständiges Profil von demografischen Daten einen gewissen Aufwand für die Nutzer und zweitens ist auch mit den steigenden *privacy*-Bedenken in letzter Zeit (z. B. Facebook, Netflix) die Bereitschaft, solche anzugeben gesunken[10]. Da diese Daten sehr wertvoll für Marktforschung und Kundensegmentierung sind, muss besonders viel Acht auf den Schutz von diesen Daten gelegt werden (vgl. Jannach et al. 2010, S. 211-232). Diese zwei Punkte haben auch dazu geführt, dass demografische Systeme kommerziell schwer einsetzbar sind. Pazzani zeigt im Detail, wie der Einsatz von solchen Systemen aussehen kann (vgl. Pazzani 1999).

Wissensbasierende Systeme haben gegenüber den anderen Systemen viele Vorteile, da sie keine Bewertungsgeschichte brauchen, sich schnell Veränderungen anpassen können und sie *per se* keine Kaltstartprobleme haben. Dennoch macht der große Aufwand für den Nutzer, immer die Nützlichkeit für verschiedene Optionen

[10] Es hat sogar dazu geführt, dass Netflix 2010 die zweite Auflage des Netflix Contests zur Verbesserung des bestehenden Filmempfehlungssystems wegen Bedenken im Bereich von *privacy* aufgeben musste. Man gewährte den Kandidaten einen eingeschränkten Zugriff auf die Netflix-Datenbank, da alle Datensätze anonymisiert wurden. Dennoch stellten Wissenschaftler von der Universität von Texas fest, dass diese Daten, auch anonymisiert, eine Identifizierung der echten Personen erlauben. Netflix musste den Wettbewerb nach einem gerichtlichen Verfahren von KamberLaw L.L.C. abbrechen. (Quelle: New York Times. Letzter Zugriff: 13.09.2010, unter http://www.nytimes.com/2010/03/13/technology/13netflix.html?_r=1).

einzugeben, den Einsatz auf dem Gebiet von Filmempfehlungen unplausibel, da Filme eine sehr große Anzahl von unterschiedlichen und schwer zu definierenden Aspekten, bezüglich „Nützlichkeit“, mitbringen. Außerdem werden die Nützlichkeitsfunktionen sehr viele Ressourcen in Form von Programmierzeit erfordern, was sie für kleine Seiten nicht rentabel macht. Towle und Quinn geben ein Beispiel für den Einsatz von solchen Systemen und für die damit verbundenen Schwierigkeiten (vgl. Towle, Quinn 2000).

Zusammenfassend kann man für dieses Kapitel sagen, dass keine der Empfehlungsgenerierungsmethoden allein die Anforderungen erfühlen kann. Daher ist die Entwicklung und Einführung von einem hybriden Empfehlungssystem notwendig, um die Nachteile der einzelnen Methoden zu reduzieren. Dieses wird im nächsten Kapitel vorgestellt.

Empfehlungssysteme	**Vorteile**	**Nachteile**
Inhaltsbasierende	A. Qualität wird besser mit der Zeit. B. Keine *Privacy*-Probleme. C. Inhalt des Objektes kann berücksichtigt werden.	I. *New user*-Problem. J. Abhängig von Qualität der Beschreibung von den Objekten. K. Qualität abhängig von Anzahl der Bewertungen. L. Überspezifisch
Kollaborative	A. D. Kein Bezug zum Inhalt der Objekte. E. Nicht überspezifisch. Empfehlungen sind dynamischer.	I, K. M. *New item* Problem. N. *“Gray sheep”* Problem. O. *Privacy*-Bedenken
Demografische	A, E, D.	I, M, O. P. Demografische Daten notwendig.
Wissensbasierende	F. Keine Bewertungsgeschichte notwendig. G. Kann sich schnelle Veränderungen anpassen. H. Kann nicht objektbezogene Details berücksichtigen.	Q. Relation zwischen Bedürfnissen und vorhandenen Optionen muss integriert werden. R. Nicht lernfähig. S. Sehr großer Eingabeaufwand für den Nutzer.

Tabelle 7: Vergleichende Tabelle von den Systemtypen (vgl. Burke 2002, S. 6)

5 Ein Empfehlungssystem für *critic.de*

Als Lösungsansatz wird auf der Basis obiger Vorüberlegungen von bei Chapphannarungsri vorgestellten Ansatz ausgegangen (Chapphannarungsri et al. 2009) und es wird ein hybrides System vorgeschlagen, das inhaltsbasierte Bewertungen mit Benutzerprofilen koppelt und welches um CF-Elemente erweitert werden kann[11].

Das hybride System soll durch die Zusammensetzung von *content-based* und *collaborative filtering* Methoden am besten die Voraussetzungen am System erfüllen und die Nachteile von beiden auf ein Minimum reduzieren. Dazu sollen die Eigenschaften der Seite, wie eine steigende aber noch nicht große Benutzeranzahl oder die begrenzten System- und Personalressourcen berücksichtigt werden.

Ein Hybrid von *content-based* und *collaborative filtering* wird *per se* folgende Vor- und Nachteile, wie in Tabelle 8 aufgeführt, aufweisen (abgeleitet von der vergleichenden Tabelle 7):

Empfehlungssystem	Vorteile	Nachteile
Ein Hybridsystem von *content-based* und *collaborative filtering*	A. Qualität wird besser mit der Zeit. C. Inhalt des Objektes kann berücksichtigt werden. E. Nicht überspezifisch. Empfehlungen sind dynamischer.	I. *New user*-Problem. J. Abhängig von Qualität der Beschreibung von den Objekten. K. Qualität abhängig von Anzahl der Bewertungen. O. *Privacy*-Bedenken

Tabelle 8: Vor- und Nachteile des Hybridsystems

Nur zwei der Nachteile sind bei diesem Hybridsystem in ihrem vollen Umfang vorhanden, nämlich das *new user*-Problem und dass das System sehr abhängig von der Qualität der Metadaten ist. Die zwei anderen Nachteile sind nachrangig. Die Qualität der Empfehlungen ist zwar von der Anzahl der Bewertungen abhängig, aber nicht in dem Umfang, wie beim *collaborative filtering* allein, da auf der Basis von Metadaten und deren Bewertungen von einem einzelnen Nutzer schon Empfehlungen zu

[11] Dieser Lösungsansatz ist eine Weiterentwicklung des bei WAM2010 vorgestellten Modells (vgl. Dimitrov, Wolff 2010).

diesem generiert werden können. Der andere Nachteil, Sicherheit von Privatdaten, ist aufgrund von der Anforderung, dass persönliche Präferenzen und Daten für andere Nutzer nicht sichtbar sein sollen, irrelevant.

Das vorgeschlagene Hybridsystem korreliert mit den Systemanforderungen (siehe Kap. 3.2) auf folgende Weise.

Die erste Anforderung, dass eine bessere Navigation gewährleistet wird, kann durch den *content-based* Teil des Systems erfüllt werden. Die Metadaten werden als Basis dafür benutzt um eine Filmähnlichkeit von Filmen zu berechnen, und ermöglicht somit das Einführen von einer Funktion „Ähnliche Filme“. Ferner erlauben sie mehr Möglichkeiten für den Nutzer zur Auswahl und Überblick von Filmen.

Die zweite Anforderung, dass möglichst genaue Filmempfehlungen generiert werden, kann am bestem durch eine Zusammenarbeit von beiden Grundsystemen gewährleistet werden. Das inhaltsbasierende System kann bei fehlenden oder wenigen *collaborative filtering*-Daten (Bewertungen von anderen Nutzern) immerhin benutzerbezogene Empfehlungen generieren. Diese können durch CF-Bewertungen verbessert werden, da *collaborative filtering* ermöglicht, dass eine Ähnlichkeit von Benutzerpräferenzen ermittelt werden kann.

Die restlichen Anforderungen beziehen sich nicht auf die Vor- und Nachteile des Systems, diese sind vielmehr durch das Datenmodell und die Implementierung zu realisieren und werden zu einem späteren Zeitpunkt angesprochen.

Der *content-based* Teil des Systems bietet ferner eine Lösung zu den inhärenten Problemen von Filmempfehlungssystemen welche im Kapitel 4.5 besprochen wurden. Dabei wird der Vorschlag von Chapphannarungsri weiterentwickelt (Chapphannarungsri et al. 2009). Der Grund für diese Problematik sind die allgemeinen Bewertungen (sog. Metabewertungen). Bei diesen generischen Bewertungen (gut, schlecht etc.) ist der tatsächliche Begründungszusammenhang der Bewertung nicht mehr zu erkennen. Solche Systeme, die unterschiedliche Kontextfaktoren berücksichtigen, werden u. a. bei Adomavicius diskutiert (Adomavicius et al. 2005).

Die offensichtliche Lösung, nämlich das Einführen von mehreren Bewertungskriterien (in diesem Fall Bewertungen von verschiedenen Filmmerkmalen), um multipara-

metrische Verfahren, wie sie gerade im Multimedia-Retrieval üblich sind, zu nutzen, bringt uns zum nächsten Problem, das der ungewichteten Kriterien. Dies wird dazu führen, dass die Bewertungen von Filmmerkmalen als gleich wichtig interpretiert werden. Zum Beispiel die Bewertung über den Regisseur wäre genauso ausschlaggebend, wie die Bewertung über das Entstehungsland. Ein solcher Zusammenhang ist eindeutig nicht korrekt, da die Präferenzen von Nutzern selten universal übertragbar sind, und wird ein Fehlverhalten bei der Empfehlungsgenerierung auslösen.

Der nächste offensichtliche Schritt wäre diese Kriterien zu gewichten, um eine Differenzierung der Vorlieben der Nutzer zu ermöglichen. Dennoch bringt diese Lösung das nächste Problem mit sich: Welche Gewichtung ist für welches Filmmerkmal korrekt?

Es gibt viele Möglichkeiten die Gewichtungen zu bestimmen, welche für verschiedene Situationen geeignet sind und welche jeweils Vor- und Nachteile haben: der eher spekulative *educated guess*, die Expertenbefragung, die Benutzerbefragung und andere.
Dennoch haben alle diese als Nachteil, dass sie nicht für jeden Nutzer persönlich zugeschnitten sind. Und im Grunde genommen muss eine Empfehlung persönlich generiert werden und nicht auf der Basis von Daten über alle Nutzer, geschweige denn von der Meinung von einzelnen Personen gebildet werden.

Content-based filtering bietet einen Teil der Lösung, indem es mehrere Filmmerkmale zur Bewertung, im Gegensatz zu *collaborative filtering* bereitstellen kann. Diese kann man natürlich gewichten. Als Lösung für das Problem der sinnvollen Gewichtung von den Kriterien wird ein sowohl impliziter (automatischer) als auch expliziter Aufbau von Benutzerprofilen vorgeschlagen, damit die Profile einerseits direkt nutzerbezogen sind und andererseits eine automatische objektive Aktualisierung gewährleistet ist.
Hinsichtlich einer automatischen bzw. impliziten Ermittlung von Nutzerprofilen besteht ebenfalls eine Kaltstartproblematik, weshalb zunächst mit einem expliziten Aufbau des Nutzerprofils gearbeitet wird. Das tatsächliche Surfverhalten des Nutzers führt ergänzend zum kompletten Aufbau seines Profils.

5.1 Das Datenmodell

Die Repräsentation der wesentlichen Filmmerkmale wird durch ein einfaches Datenmodell ermöglicht. Dabei werden die Ausprägungen der Merkmale Genre, Regisseur (*Director*), Schauspieler (*Actor*) und Land (*Country*) jeweils als Vektoren repräsentiert.
Dies bedeutet, dass jeder Film, wenn vier Merkmale zur Filmkennzeichnung genommen werden, durch ein Quadrupel von vier Vektoren (die Vektoren: *Genre*, *Director*, *Actor*, *Country*) dargestellt wird. Diese werden vorerst benutzt, um die Ähnlichkeit von Filmen untereinander zu bestimmen.

Eine Berücksichtigung der Entstehungsepoche der Filme nach Dekaden, wie von Chapphannarungsri vorgeschlagen (vgl. Chapphannarungsri et al. 2009), hat sich als wenig praktikabel erwiesen, da Jahre an sich schon strukturierte Daten sind und deren Abstraktion durch Vektoren keine Vorteile mit sich bringt (z. B.: Ein Film, der in der Zeitspanne von 1995-2000 entstanden ist, wird die gleiche Ähnlichkeit zu einem Film aus den Jahren 1965-1970 aufweisen, wie zu einem aus den Jahren 2005-2010, nämlich gar keine. Dieses Ergebnis wäre offensichtlich nicht erwünscht, wenn Vektoren dafür eingesetzt würden).

Bei Bedarf kann das Datenmodell durch die bereits vorhandenen Daten in der Filmdatenbank erweitert werden (Produzent, Drehbuchautor etc.). In dieser Arbeit werden die Beispiele durch das Quadrupel (Genre, Land, Regisseur, Schauspieler) abgebildet, um eine leichtere Darstellung zu ermöglichen.

Die folgenden Überlegungen begründen die Repräsentation der relevanten Inhalte einer Filmdatenbank durch Vektoren:

- Vektoren können aufgrund ihrer einfachen Struktur leichter als strukturiertere Datenmodelle in gängigen mathematischen Modellen genutzt werden.
- Bei Einschränkung auf binäre Vektoren (diese enthalten nur die Werteausprägungen 0 und 1), können Bitoperationen zum Einsatz kommen, was die Performanz dramatisch steigern kann. Dies ist hier nicht vollständig realisierbar, da die Bewertungen nicht-binäre Ausprägungen haben.
- Vektoren sind zur Darstellung von Genres und vergleichbaren Filmmerkmalen sehr gut geeignet, da diese i. d. R. eine überschaubare Anzahl an Werteaus-

prägungen haben. Dadurch wird eine Abstraktion von Metadaten ermöglicht, damit diese in mathematischen Modellen benutzt werden können.

Die Filmmerkmale müssen allerdings in einem angemessenen Verhältnis zum Umfang der Filmdatenbank stehen, dies wird am besten an den Filmgenres ersichtlich: In einer Datenbank mit 2000 Filmen, die nur 10 Filmgenres ausweist, werden zwangsläufig, da die Variationen von Beschreibungen anhand von Filmgenres sehr wenig wären, sehr viele Filme als vollkommen ähnlich dargestellt d. h. das Modell wäre zu unspezifisch für Empfehlungen. Umgekehrt führt ein Datenmodell mit 200 Genres bei 2000 Filmen zu einer Überspezifizierung der Ergebnisse bzw. sogar zu leeren Empfehlungslisten, wenn man einen zu hohen *cut off*-Wert für die Filmähnlichkeit einsetzt. Das nachfolgend vorgestellte Modell soll den Rahmenbedingungen in angemessener Weise Rechnung tragen und zu einer adäquaten Anzahl hinreichend spezifischer Empfehlungen führen.

Für die Repräsentation der Vektoren in der Datenbank ist ein geeignetes Format zu wählen. Eine einfache Lösung wäre die Zuweisung eines Feldes in einer Tabelle zu jeder Dimension des Vektors. Bei sehr langen Vektoren kann dies aber zu Problemen führen, da in MySQL die maximale Anzahl von Feldern auf 64 MB oder 4096 Felder vom Typ *Integer* eingeschränkt ist. Diese Darstellungsform ist auch sehr speicher- und prozessorintensiv. Dazu empfiehlt es sich bei längeren Vektoren (ab ca. 50 Elementen), was bei Filmmerkmalen der Fall ist, diese als Ganzes in Text der Binärfelder zu speichern, um die Anzahl von *write*-Operationen zu verringern.

Die nachfolgende Tabelle 9 illustriert Merkmale (*Feature*) und ihre Ausprägungen[12]; das Datenmodell entwickelt den Vorschlag von Chapphannarungsri weiter (vgl. Chapphannarungsri et al. 2009, S. 700, Tab. 1).

[12] Die Anzahl von Filmgenres und ihre Benennung wurde angepasst, um eine leichtere Darstellung zu ermöglichen, siehe Tabellen 5 und 6 für die komplette Genreauflistung.

Feature	***Werteausprägungen***
Feature(1): **Genre**	*Action(1), Adventure(2), Animation(3), Children(4), Comedy(5), Crime(6), Documentary(7), Drama(8), Fantasy(9), Film-Noir(10), Horror(11), Musical(12), Mystery(13), Romance(14), Sci-Fi(15), Thriller(16), War(17) und Western(18)*
Feature(2): **Regisseur**	Aakeson, Kim Fupz(1), …, Zvi Howard(13600); nur positive Werte werden gespeichert.
Feature(3): **Schauspieler**	Aakeson, Kim Fupz(1), …, Zvi Howard(13600); nur positive Werte werden gespeichert.
Feature(4): **Land**	Afghanistan(1), Argentinien(2), Armenien(3),…, USA(79), Zypern(80)

Tabelle 9: Übersicht über das Datenmodell

Abbildung 3 zeigt zunächst den Einsatz des Datenmodells und die Befüllung der Beschreibungsvektoren.

Für „The Matrix“ gelten folgende Beschreibungsmerkmale: Genre: *Action, Adventure, Sci-Fi, Thriller*; Land: *USA*; Regisseur: *Andy* und *Lana Wachowski*; Schauspieler: *Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss u.v.a..* Dies wird wie folgt als vier Vektoren im Datenmodell repräsentiert:

Feature(1): Genre	1	1	0	0	.....	0	0	1	1	0	0
Feature(2): Regisseur	0	0	0	0	.....	0	0	1	1	0	0
Feature(3): Schauspieler	0	...	0	...	1	..	1	0	...	1	0
Feature(4): Country	0	0	0	0	0	0	...	0	0	1	0

Abbildung 3: Repräsentation von vier Filmmerkmalen als Vektoren

5.2 Beschreibung des Verfahrens

Die Filmmetadaten, welche mittels Vektoren im Datenmodell repräsentiert sind, fließen dabei in einen *movie feature vector* (MFV) ein, der die oben genannten Merkmale umfasst. Zum Abgleich als Grundlage von Empfehlungen dient ein *user preference vector* (UPV), der das Interessenprofil des Nutzers reflektiert. Hinsichtlich einer automatischen bzw. impliziten Ermittlung von Nutzerprofilen besteht ebenfalls eine Kaltstartproblematik, weshalb zunächst mit einem expliziten Aufbau des Nutzerprofils gearbeitet wird. Das tatsächliche Surfverhalten des Nutzers führt ergänzend zum Aufbau eines *feature dependency vector* (FDV). In diesen werden Suchanfragen des Nutzers ebenso wie explizite Ergebnisbewertungen – im Sinne des *relevance feedback* – oder die Auswahl bestimmter Sortierkriterien in die Ergebnisdarstellung durch Benutzer aufgenommen. UPV und FPV werden als benutzerbezogene Daten in einem gemeinsamen Vektor zusammengeführt. Diese Vektoren werden im nachfolgenden Abschnitt definiert.

5.2.1 Movie Feature Vector (MFV)

Der *movie feature vector* ermöglicht als mathematische Abstraktion von Filmmetadaten die Berechnung von Filmähnlichkeiten und die Bewertung von den verschiedenen Filmmerkmalen und bildet somit die Basis des Empfehlungssystems.

Der MFV wird nach dem Einführen eines neuen Films im System erstellt oder wird nach dem Einführen von neuen Einträgen in den Filmmerkmallisten aktualisiert. Der MFV wird wie folgt definiert:

$$MFV = ((f11, f12, \ldots, f1m), (f21, \ldots, f2m), (fN1, \ldots, fNm))$$

Wobei bei f(ij): i ist das Feature/Kriterium, j ist die Komponente davon, m ist die Anzahl der Komponenten und N die Anzahl der Features/Kriterien. Als Werte für f(ij) werden anfangs 1 und 0 (wahr oder falsch) festgelegt, um darzustellen, ob dieses Filmmerkmal bei diesem Film präsent ist oder nicht. Dennoch werden diese nicht als strikt binär festgelegt, sodass eine Gewichtung zu einem späteren Zeitpunkt möglich ist.

Der MFV, für den in Abbildung 3 dargestellten Film „The Matrix“, wird demzufolge so aussehen:

$MFV(The\ Matrix) = ((1,1,\ldots,1,1,0,0),(0,0,\ldots,1,1,0,0),(0,..,1,\ldots,1,\ldots,1,0\ ),(0,\ldots,0,1,0))$

5.2.2 User Preference Vector (UPV)

Der UPV stellt die Vorlieben der Benutzer für die verschiedenen Filmmerkmale dar und enthält seine persönlichen Gewichtungen der Merkmale. Dieser wird explizit durch Fragen ermittelt. Dieser Vektor wird immer neu generiert, nachdem der Nutzer eine neue Bewertung bezüglich eines Films oder eines Filmmerkmals abgegeben hat.

Dem Nutzer werden bei der Bewertung von Filmen zwei Möglichkeiten angeboten: Er kann den Film allgemein durch eine sog. Metabewertung bewerten, wobei diese als Gewichtung für alle präsenten Merkmale benutzt wird, oder er kann jedes einzelne Filmmerkmal separat bewerten, um sein Benutzerprofil genauer zu gestalten, was bessere Empfehlungen generieren kann.
Der UPV stellt das langfristige Benutzerprofil dar, indem er das arithmetische Mittel seiner Bewertungen über die Filmmerkmale enthält.

Die Bewertungen gewichten den MFV für die bewerteten Filme durch Multiplikation so, dass folgender multiplizierter MFV (mMFV) gebildet wird:

$mMFV = ((f11,f12,\ldots,f1m),(f21,\ldots,f2m\ ),(fN1,\ldots,fNm\ ))$, $f(ij)$ kann Werte von 0 bis 1 einnehmen.

Wobei $mMFV = f(ij) \times a(ij)$. $a(ij)$ ist seine Bewertung bezüglich Kriterium i und Komponente j davon. $a(ij)$ kann Werte von 0 bis 1 einnehmen.

Ein Beispiel: Der Benutzer gibt an, dass der Film „The Matrix“ ihm sehr gut wegen der Genres *Action, Adventure, Sci-Fi* gefallen hat, der Rest der Kriterien war ambivalent für ihn, d. h. neutrale Bewertung für *Thriller, Actor, Director* und *Land*. Wenn wir festlegen, dass sehr gut = 1, neutral = 0.5 und sehr schlecht = 0 ist. Kommen wir auf mMFV (*The Matrix*) ((1, 1, ..., 1, 0.5, 0, 0), (0,0, ..., 0.5, 0.5, 0, 0), (0, .., 0.5, …, 0.5, ..., 0.5, 0), (0, ..., 0, 0.5, 0)).

Der UPV wird von der Summe allen mMFV wie folgt abgeleitet:

$UPV = \frac{\sum \mathrm{mMFV}}{\mathrm{N}}$, wobei N die Anzahl der bewerteten Filme ist.

5.2.3 Feature Dependency Vector (FDV)

Der FDV wird implizit durch das Beobachten von Benutzerverhalten ermittelt (vgl. Maneeroj 2005). Er stellt dar, wie sehr der Benutzer von bestimmten Kriterien (Filmmerkmale) bei der Auswahl von Filmen abhängig ist.

Jedes Mal wenn der Benutzer nach bestimmten Filmmerkmalen (spezifische Genres oder ein bestimmter Schauspieler) sucht oder seine Ergebnisse durch die vorhandenen Filmmetadaten sortiert und dann auf einem Film klickt oder durch *relevance feedback* bestätigt, dass die Ergebnisse nach seinem Geschmack sind, wird der FDV aktualisiert.

$FDV = FDV + k(ij)$, wobei $k(ij)$ ein Vektor ist, der eine Statistik über das Suchverhalten nach der letzten Interaktion des Benutzers mit dem System enthält, $k(ij) = 1$ falls der Benutzer dieses Kriterium, um Filme zu sortieren, zu filtern oder auszuwählen benutzt hat, ansonsten $k(ij) = 0$.

Der UPV und der FDV müssen kombiniert werden, um die ganze gesammelte Information zu nutzen. Am einfachsten wäre es, sie zu multiplizieren, um einen mUPV (*modified user preference vector*) zu bekommen. Dieser wird das tatsächliche Benutzerprofil darstellen, welches durch implizites und explizites Sammeln von Benutzerpräferenzen gebildet worden ist.

$mUPV = ((f11, f12, \ldots, f1m), (f21, \ldots, f2m), (fN1, \ldots, fNm))$, $f(ij)$ kann Werte von 0 bis 1 einnehmen.

Ehe beide Vektoren zusammengeführt werden können, muss auch der $FDV(ij)$ auf dem Bereich von 0 bis 1 normalisiert werden:

$$FDV'(ij) = (FDV(ij) - min) \times \frac{MAXnorm - MINnorm}{max - min} + MINnorm$$

Dabei setzen wir MAXnorm auf 1 und MINnorm auf 0 fest; min und max stellen den kleinsten und den größten Wert innerhalb von dem Vektor $FDV(ij)$ dar.

Der mUPV wird wie folgt gebildet:

$$mUPV(ij) = UPV(ij)xFDV(ij)$$

Für die so gebildeten Vektoren ist aus den bekannten Vergleichsmetriken (u. a. Euklidische Distanz, Dice-, Jaccard-, Cosinus-Maß, vgl. dazu schon [Noreault et al. 1981; Jones, Furnas 1987; Santini, Jain 1999]) ein geeignetes Ähnlichkeitsmaß auszuwählen.

5.3 Die Wahl des Ähnlichkeitsmaßes

Ähnlichkeitsmaße geben für ein Vektorpaar einen numerischen Wert, der die Ähnlichkeit zwischen den Vektoren darstellt. Man kann sie in zwei Arten unterscheiden: Korrelations- und Distanzmaße. Im Grunde genommen werden sie in der Statistik als Ähnlichkeitsmaße und Distanzmaße definiert, da diese Maße in dieser Arbeit im Hinblick auf Filmähnlichkeit betrachtet werden, werden diejenigen, die eine Korrelation darstellen, Korrelationsmaße (auch eine gängige Benennung in der deskriptiven Statistik) genannt. Alle Maße haben ihre Vor- und Nachteile und müssen für das jeweils spezifische Problem ausgewählt werden.

Ziel ist es, ein Ähnlichkeitsmaß zu finden, das mit den folgenden Beschaffenheiten der Vektoren in der Datenbank das konstanteste Verhalten aufweist:

- Sehr lange Vektoren (Personenvektoren mit über 14.000 Stellen).
- Vektoren mit sehr vielen Nullwerten (von den 14.000 Stellen der Personenvektoren sind pro Film ca. 30 nicht Nullwerte).
- Gewichtete Vektoren, mit einer Gewichtung zwischen 0 und 1. D. h. sie sind auf die Werte von 0 bis 1 normalisiert, es handelt sich somit um normalisierte Vektoren.
- Außerdem sollen Termgewichtung und Position der Terme innerhalb des Vektors gleichermaßen bewertet werden (z. B. wenn die gleichen Genres bei zwei Filmen vorhanden sind, ist genau so determinierend für die Filmähnlichkeit, wie die Gewichtung der einzelnen Genres).

In dieser Arbeit wird kurz auf die gängigsten Ähnlichkeitsmaße eingegangen und ihre Schwächen und Stärken werden veranschaulicht (vgl. dazu Haenelt 2007).
Die gängigsten Maßen sind:
Korrelationsmaße:

- Skalarprodukt
- Cosinus (Skalarprodukt nicht normierter Vektoren)
- Dice
- Jaccard

Distanzmaße:

- Euklidischer Abstand

Es muss vorerst erwähnt werden, dass gewichtete Vektoren nicht nur eine mathematische Darstellung von Daten sind. Sie tragen vielmehr auch einen semantischen Wert, der ganz genau interpretiert werden kann (vgl. Jones, Furnas 1987).
Nehmen wir als Beispiel fünf *feature dependancy*-Vektoren (V1-V5), welche in Tabelle 10 definiert sind, als Abbildungen von Benutzerprofilen und beschränken diese nur auf zwei Filmmerkmale (Genre und Regie), um eine grafische Darstellung zu ermöglichen.

	V 1	**V 2**	**V 3**	**V 4**	**V 5**
Genre	0.1	0.1	0.2	0.1	0.1
Regie	0.1	0.1	0.2	0.5	0.5

Tabelle 10: Fünf Beispielvektoren

Ihre grafische Darstellung kann auf Abbildung 4 betrachtet werden.

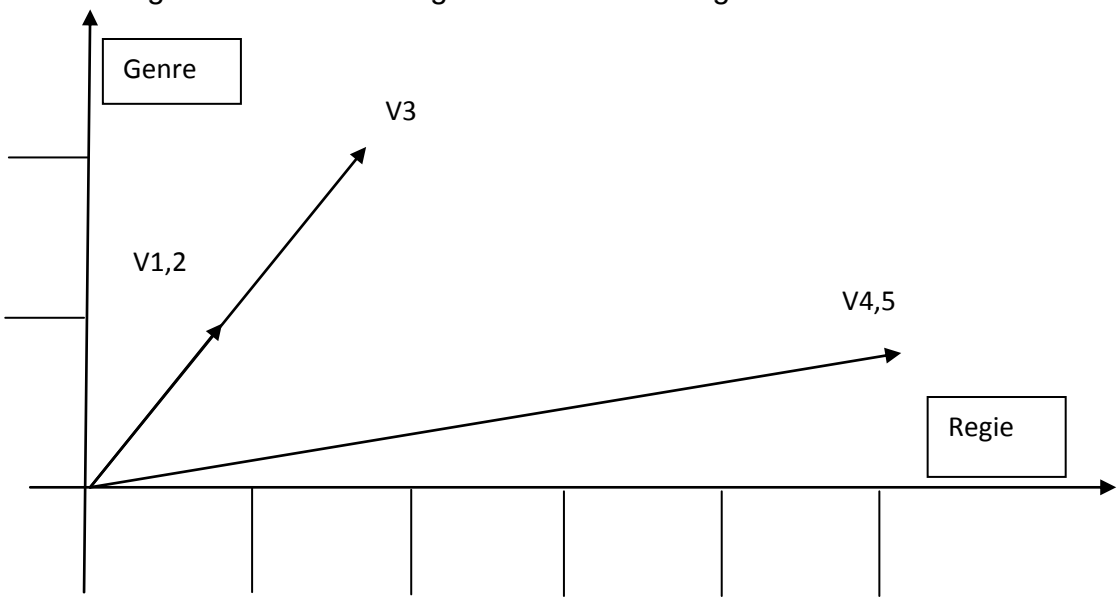


Abbildung 4: Grafische Darstellung der fünf Beispielvektoren (vgl. Jones, Furnas 1987, S. 442, Abb. 1)

Die Vektoren haben zwei Merkmale: Richtung und Länge. Diese lassen sich folgendermaßen interpretieren (vgl. Jones, Furnas 1987, S. 420-423).

- Richtung
    - wird durch das objektinterne Verhältnis der Terme zueinander definiert, in dem Fall von den vorgestellten UPV, wie die Terme (Regie und Genre) gewichtet sind. V1 und V3 weisen die gleiche Richtung auf, da Regie und Genre in gleichem Verhältnis zueinanderstehen.
    - Jones und Furnas nennen diese Interpretation „*topic*“ oder das Thema des Vektors. Auf UPV angewandt wird es die Abhängigkeit der Benutzer von den Merkmalen, Genre und Regie, bei der Auswahl von Filmen darstellen.
- Länge
    - Die Länge wird, wenn sie im Verhältnis zu anderen Vektoren betrachtet wird, durch das objektübergreifende Verhältnis der Terme definiert. Bei dem Beispiel kann man sehen, dass V3 zweimal länger als V1 ist, da diese eine doppelt stärkere Gewichtung aufweist.
    - Jones und Furnas nennen sie *„intensity“* oder Intensität des Themas. Als UPV wird damit nicht die Abhängigkeit der Benutzer von den Merkmalen dargestellt, sondern wie stark sie davon abhängig sind.

Bei den Korrelationsmaßen entspricht der größte Wert dem ähnlichsten Paar. Je größer der Wert ist, desto ähnlicher sind die zwei Objekte zueinander.

5.3.1 Skalarprodukt

Der Punkt oder das Skalarprodukt ist das einfachste Ähnlichkeitsmaß (auch einfache Übereinstimmung genannt). Seine Einfachheit hat dazu beigetragen, dass es als Teil von komplexeren Maßen benutzt wird. Seine algebraische Darstellung lautet:

$\sum_{i=1}^{n} A(i) \times B(i)$, wenn A und B zwei n-stellige Vektoren sind.

Die Eigenschaften des Skalarprodukts lassen sich am besten durch ein Beispiel veranschaulichen. Nehmen wir an, dass wir sechs Vektoren (MFV1-6), wie in Tabelle 11 definiert, mit jeweils sechs Möglichkeiten für Filmgenres haben.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
Action	0.1	0.1	0.1	0.2	0.1	0.1
Adventure	0.1	0.0	0.2	0.2	0.0	0.1
Drama	0.1	0.1	0.3	0.2	0.5	0.1
Fantasy	0.1	0.0	0.4	0.2	0.0	0.1
Sci-Fi	0.1	0.1	0.5	0.2	0.6	0.1
Horror	0.1	0.0	0.6	0.2	0.0	0.1

Tabelle 11: Sechs Beispielvektoren für Genres

Deren Filmähnlichkeit wäre auf der Basis des Skalarprodukts, wie in Tabelle 12.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
MFV1	-	0.03	0.21	0.12	0.12	0.06
MFV2	0.03	-	0.09	0.06	0.12	3.0
MFV3	0.21	0.09	-	0.42	0.46	0.21
MFV4	0.12	0.06	0.42	-	0.24	0.12
MFV5	0.12	0.12	0.46	0.24	-	0.12
MFV6	0.06	0.03	0.21	12.0	0.12	-

Tabelle 12: Filmähnlichkeit mittels des Skalarprodukts

In Tabelle 12 lassen sich vier negative Eigenschaften vom Skalarprodukt erkennen[13]:

- Die Ähnlichkeit des Objekts ist proportional von der Gewichtung von beliebigen Termen abhängig. Eine Erhöhung des Terms steigert die Ähnlichkeit, dies gilt vor allem für Vektoren mit dem gleichen Thema (Richtung). Jones und Furnas nennen es *radial monotonicity* und *component-wise monotonicity* (ebd. S. 425).

  Als Beispiel kann man $sim(MFV1, MFV2) < sim(MFV1, MFV2)$ betrachten.
- Die Gewichtung der verschiedenen Terme ist unabhängig. Eine sehr hohe Gewichtung von beliebigen Termen im Vergleich zu anderen Vektoren führt zu

[13] Jones und Furnas geben eine detailliertere mathematische Beschreibung der Probleme (vgl. Jones, Furnas 1987, S. 423-426).

sehr hohen Ähnlichkeitswerten. Jones und Furnace nennen es *unbounded single-component influence* (ebd. S. 425).

Als Beispiel dafür kann man MFV3 nehmen.

- Absurde Ähnlichkeitswerte bei nicht normalisierten Vektoren aufgrund der vorherigen Probleme.

  Als Beispiel sehen wir, dass $sim(MFV1, MFV4) > sim(MFV1, MFV6)$ obwohl MFV1 und MFV6 identisch sind.

Alle diese Probleme machen das Skalarprodukt zu einem ungeeigneten Kandidaten für das Ähnlichkeitsmaß. Vor allem, dass die Gewichtung mehr zu der Ähnlichkeit beiträgt als die Richtung macht es unplausibel für Vektoren mit vielen Nullwerten:
z. B. $a = (1,1,1,1), b = (1,1,1,1)\ und\ c = (0,4,0,0) \rightarrow\ sim(a,b) = sim(a,c) = 4$

5.3.2 Das Cosinus-Maß

Die Formel für die Cosinusähnlichkeit lautet:

$$\frac{\sum_{i=1}^{n} A(i) \times B(i)}{\sqrt{\sum_{k=1}^{n} A(k)^2} \times \sqrt{\sum_{j=1}^{n} A(j)^2}}$$

An der Formel kann man schon einige der Eigenschaften des Cosinus-Maßes erkennen. Der Zähler ist das oben vorgestellte Skalarprodukt. Der Nenner ist der Hauptteil der Formel und gibt die wichtigste Eigenschaft des Maßes an, das Skalarprodukt wird nämlich durch das Produkt der Längen (euklidische Länge) des Anfragevektors und des zu vergleichenden Vektors dividiert. Diese Division stellt einfach eine Normalisierung der Vektorlängen dar. Daraus lassen sich folgende Eigenschaften erkennen, wenn wir die oben dargestellten Filmvektoren (MFV1-6), wie auch in Tabelle 13 definiert, als Beispiel nehmen.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
Action	0.1	0.1	0.1	0.2	0.1	0.1
Adventure	0.1	0.0	0.2	0.2	0.0	0.1
Drama	0.1	0.1	0.3	0.2	0.5	0.1
Fantasy	0.1	0.0	0.4	0.2	0.0	0.1
Sci-Fi	0.1	0.1	0.5	0.2	0.6	0.1
Horror	0.1	0.0	0.6	0.2	0.0	0.1

Tabelle 13: Sechs Beispielvektoren für Genres

Die daraus folgende Ähnlichkeitswerte werden in Tabelle 14 aufgeführt.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
MFV1	-	0.707	0.898	1.000	0.622	1.000
MFV2	0.707	-	0.544	0.707	0.879	0.707
MFV3	0.898	0.544	-	0.898	0.612	0.898
MFV4	1.000	0.707	0.898	-	0.622	1.000
MFV5	0.622	0.879	0.612	0.622	-	0.622
MFV6	1.000	0.707	0.898	1.000	0.622	-

Tabelle 14: Filmähnlichkeit mittels des Cosinus-maßes

Die Eigenschaften des Cosinus-Maßes lassen sich so zusammenfassen:

- Die Ähnlichkeit zwischen zwei Vektoren wird nur durch die Richtung (Thema) des Vektors bestimmt. Nur das Verhältnis der Terme und nicht deren Gewichtung hat Einfluss auf die Ähnlichkeit, da die vorher erwähnte Normalisierung das Cosinus-Maß zu einer radialen Konstante macht. Furnas und Jones bezeichnen es als *radial monotonicity* (ebd. S. 429).
- Diese Normierung macht eine Manipulation durch beliebige Erhöhung einzelner Terme im Vergleich zum Skalarprodukt unmöglich (*unbounded single-component influence*, ebd. S. 429).
  Als Beispiel kann man $sim(MFV4, MFV6) = sim(MFV1, MFV6)$ betrachten.
- Für nicht negative Vektoren ist die Ähnlichkeit immer zwischen 0 und 1. Dies bedeutet, dass es einen Idealwert gibt, nämlich das Maximum 1 (*boundedness of similarity values*, ebd. S. 429).
- Die Ähnlichkeit wird sehr stark durch das Vorhandensein von Nullwerten beeinflusst, sogar bestraft, da dies die Richtungen von dem Anfrage- und Zielvektor sehr auseinander treibt ("*representational penalty*", ebd. S. 429).
  Als Beispiel sehen wir, dass $sim(MFV3, MFV1) > sim(MFV3, MFV5)$ ist.

Die zwei Eigenschaften, nämlich dass die Termgewichtung zwischen den einzelnen Vektoren keine Rolle spielt und dass die Nullwerte sehr großen Einfluss auf die Ähnlichkeit haben, machen das Cosinus-Maß ungeeignet für den Einsatz auf das vorgestellte Modell, das eben diese zwei Eigenschaften vorweist. Das Cosinus-Maß

lässt sich demzufolge nur auf den binären MFV anwenden, was nur eine Teillösung wäre.

5.3.3 Dice-Koeffizient

Die Formel für den Dice-Koeffizient lautet:

$$\frac{2 \times \sum_{i=1}^{n}(A(i) \times B(i))}{\sum_{k=1}^{n} A(k) + \sum_{j=1}^{n} B(j)}$$

Der Dice-Koeffizient wird gebildet, indem die Summe der Produkte von dem Anfragevektor und dem zu vergleichenden Vektor mal zwei multipliziert wird und durch die Summe von den Vektorlängen geteilt wird. Dieses Maß stellt eine große Verbesserung im Vergleich zu dem Skalarprodukt und dem Cosinus-Maß dar. Diese lassen sich wieder am besten durch eine Tabelle illustrieren.

Wir haben die gleichen Filme mit den MFV1-6, wie in Tabelle 15 definiert.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
Action	0.1	0.1	0.1	0.2	0.1	0.1
Adventure	0.1	0.0	0.2	0.2	0.0	0.1
Drama	0.1	0.1	0.3	0.2	0.5	0.1
Fantasy	0.1	0.0	0.4	0.2	0.0	0.1
Sci-Fi	0.1	0.1	0.5	0.2	0.6	0.1
Horror	0.1	0.0	0.6	0.2	0.0	0.1

Tabelle 15: Sechs Beispielvektoren für Genres

Die daraus folgenden Ähnlichkeitswerte sind in Tabelle 16 zu sehen.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
MFV1	-	0.0666	0.1555	0.1333	0.1333	0,1
MFV2	0.0666	-	0.1500	0.0800	0.16	0.0666
MFV3	0.1555	0.1500	-	0.2545	0.2787	0.1555
MFV4	0.1333	0.0800	0.2545	-	0.2	0.1333
MFV5	0.1333	0.16	0.2787	0.2	-	0.1333
MFV6	0.1	0.0666	0.1555	0.1333	0.1333	-

Tabelle 16 : Filmähnlichkeit mittels des Dice-Koeffizienten

Man kann sehen, dass der Dice-Koeffizient sowohl die Gewichtung (Vektorlänge) als auch das Thema (Vektorrichtung) berücksichtigt. Dennoch spielt die Gewichtung eine viel größere Rolle, was dazu führt, dass die Qualität der Ergebnisse sehr von der Vektorlänge abhängig ist.

Dies kann zu folgenden Ergebnissen führen:
Als Beispiel: $sim(MFV1, MFV4) = sim(MFV1, MFV4)$.
Beide Ähnlichkeiten sind gleich, obwohl die Vektoren nicht die gleiche Richtung haben. Oder noch schlimmer $sim(MFV1, MFV4) = sim(MFV1, (0{,}0{,}0{,}0{,}0.6{,}0.6))$. Wenn man die Ergebnisse interpretiert, wird es heißen, dass eine Person, die diese sechs Genres nicht besonders mag, die gleiche Ähnlichkeit aufweisen wird wie eine, die *Sci-Fi* und *Horror* mag.

Ein weiteres Beispiel dafür, dass die Gewichtung eine zu leichte Manipulation der Ähnlichkeit erlaubt, ist $sim(MFV3, MFV4) < sim(MFV3, MFV5)$. Höhere Werte führen zwangsläufig zu einer höheren Ähnlichkeit. Auch identische Vektoren haben eine kleinere Ähnlichkeit untereinander, wenn sie mit hochgewichteten Vektoren verglichen werden:

$$sim(MFV1, MFV6) < sim(MFV1, MFV5).$$

Da im Datenmodell die Vektoren sehr viele Nullwerte haben, welche hauptsächlich die Richtung beeinflussen, und die Bewertungen der Benutzer sehr unterschiedlich zueinander sein können, ist der Dice-Koeffizient auch nicht geeignet.

5.3.4 Jaccard-Koeffizient

Die Formel für den Jaccard-Koeffizient lautet:

$$\frac{\sum_{i=1}^{n}(A(i) \times B(i))}{\sum_{j=1}^{n} A(j) + \sum_{k=1}^{n} B(k) - \sum_{i=1}^{n}(A(i) \times B(i))}$$

Der Jaccard-Koeffizient ist dem Dice-Koeffizient sehr ähnlich und weist fast die gleichen Eigenschaften auf. Betrachten wir die MFV1-MFV6, diese werden noch einmal zu besserem Überblick in Tabelle 17 aufgelistet, und ihre Ähnlichkeitswerte, welche mittels des Jaccard-Koeffizienten ermittelt worden sind.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
Action	0.1	0.1	0.1	0.2	0.1	0.1
Adventure	0.1	0.0	0.2	0.2	0.0	0.1
Drama	0.1	0.1	0.3	0.2	0.5	0.1
Fantasy	0.1	0.0	0.4	0.2	0.0	0.1
Sci-Fi	0.1	0.1	0.5	0.2	0.6	0.1
Horror	0.1	0.0	0.6	0.2	0.0	0.1

Tabelle 17: Sechs Beispielvektoren für Genres

Die Ähnlichkeitswerte mittels der Jaccard-Koeffizienten sind in die Tabelle 18 zu sehen.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
MFV1	-	0.0345	0.0843	0.0714	0.0714	0.0526
MFV2	0.0345	-	0.0390	0.0417	0.0870	0.0345
MFV3	0.0843	0.0390	-	0.1458	0.1620	0.0843
MFV4	0.0714	0.0417	0.1458	-	0.1111	0.0714
MFV5	0.0714	0.0870	0.1620	0.1111	-	0.0714
MFV6	0.0526	0.0345	0.0843	0.0714	0.0714	-

Tabelle 18: Filmähnlichkeit mittels des Jaccard-Koeffizienten

Genau wie bei dem Dice-Koeffizienten kann man erkennen, dass die Gewichtung oder die Vektorlänge maßgebend ist. Dieses Maß verhält sich asymmetrisch, was bei Vektoren mit sehr unterschiedlichen Längen/Gewichtungen, was bei Filmpräferenzen anzunehmen ist, zu fehlerhaftem Clustering führen kann:
Als Beispiel die Vektoren MFV3 und MFV5, sie haben einen viel größeren Ähnlichkeitswert dank ihrer höheren Gewichtung im Vergleich zu anderen Vektoren.

Die Manipulation der Ähnlichkeit durch sehr hohe einzelne Werte bleibt bestehen, als Beispiel:

$$sim(MFV1, MFV4) = sim(MFV1, MFV5).$$

Außerdem wird das Vorhandensein von wenigen gemeinsamen Einträgen sehr stark bestraft. Je weniger gemeinsame Beiträge existieren, desto größer ist der Nenner,

also eine geringere Ähnlichkeit. Ein Maximum an Ähnlichkeit ist auch nicht gegeben. Alle diese Punkte machen auch dieses Maß für den Einsatz in diesem Datenmodell ungeeignet.

5.3.5 Der euklidische Abstand

Der euklidische Abstand ist im Gegensatz zu den vorherigen Korrelationsmaßen ein Distanzmaß. Er wird in diesem Fall, wie die vorherigen Maße, auf normalisierte Vektoren angewandt, um Skalierungsprobleme zu meiden (vgl. Santini, Jain 1999, S. 880-881). Bei diesem Maß wird eine hohe Ähnlichkeit nicht durch einen möglichst hohen Wert abgebildet, sondern durch eine möglichst geringe Distanz. Dies bedeutet, dass es einen Idealwert gibt, der 0 beträgt, nämlich keine Distanz. Die Formel lautet:

$$\sqrt{\sum_{i=1}^{n}(A(i)-B(i))^2}$$

Der euklidische Abstand lässt sich sehr einfach interpretieren. Er stellt die kürzeste Luftlinie zwischen zwei Punkten dar. Da er ein Distanzmaß ist, kann man auch eine maximale Distanz berechnen. Oder anders formuliert: Es sind immer die geringste und höchste Ähnlichkeit bekannt, was eine Umwandlung in einem Ranking (z. B. in Prozenten) unproblematisch macht. Betrachtet man die MFV-Tabelle 19 und die Ähnlichkeitstabelle 20, kann man diese und andere Eigenschaften erkennen.

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
Action	0.1	0.1	0.1	0.2	0.1	0.1
Adventure	0.1	0.0	0.2	0.2	0.0	0.1
Drama	0.1	0.1	0.3	0.2	0.5	0.1
Fantasy	0.1	0.0	0.4	0.2	0.0	0.1
Sci-Fi	0.1	0.1	0.5	0.2	0.6	0.1
Horror	0.1	0.0	0.6	0.2	0.0	0.1

Tabelle 19: Sechs Beispielvektoren für Genres

	MFV1	MFV2	MFV3	MFV4	MFV5	MFV6
MFV1	-	0.1732	0.7416	0.2449	0.6633	0.0
MFV2	0.1732	-	0.8717	0.3872	0.6403	0.1732
MFV3	0.7416	0.8717	-	0.5567	0.7810	0.7416
MFV4	0.2449	0.3872	0.5567	-	0.6164	0.2449
MFV5	0.6633	0.6403	0.7810	0.6164	-	0.6633
MFV6	0.0	0.1732	0.7416	0.2449	0.6633	-

Tabelle 20: Filmähnlichkeit mittels des euklidischen Abstands

Man kann in Tabelle 20 erkennen, dass das Maß sowohl Richtung als auch Länge berücksichtigt. Besonders wichtig ist, dass sich dies symmetrisch verhält. Beide Vektoreigenschaften haben nämlich den gleichen Einfluss. Dies ist sehr wichtig für die Interpretation des Datenmodells. Da die Vektorrichtung das Thema des Filmes oder die Benutzerpräferenz und die Länge die Intensität davon bestimmt, ist es durchaus sinnvoll, dass beide die gleiche Wichtigkeit haben.

Als Beispiel kann man $sim(MFV1, MFV6) = 0$ nehmen: zwei Filme haben die höchste Ähnlichkeit bei gleichen Merkmalen und der gleichen Bewertung der Merkmale.

Dagegen wird die niedrigste Ähnlichkeit durch $sim(MFV2, MFV3) = 0.8717$ dargestellt: zwei Filme, welche weder ein ähnliches Thema, noch eine ähnliche Gewichtung aufweisen. An diesem Beispiel lässt sich auch die maximale Distanz zeigen: Bei einem sechsstelligen Vektor mit Werten von 0 bis 1 beträgt sie $\mathrm{Dmax} = \sqrt{6 * 1}$=2.4494. Außerdem lässt sich die Ähnlichkeit durch die Erhöhung einzelner Termen nicht positiv beeinflussen, zum Beispiel MFV3 und MFV5: ihre sehr hohen Gewichtungen haben eine profunde negative Wirkung auf ihre Ähnlichkeit zu den anderen Vektoren, da diese eine kleinere Gewichtung oder andere Richtung aufweisen. Eben diese Eigenschaften machen den euklidischen Abstand zu einem geeigneten Ähnlichkeitsmaß für Vektoren, welche Filmmerkmale oder dazugehörige Benutzerpräferenzen repräsentieren.

Der euklidische Abstand hat dennoch auch einen Nachteil bei dem Umgang mit binären Vektoren, welche unter anderem in der Form von MFV für die Ähnlichkeit zwischen den Filmen zuständig sind. Das Problem ist an sich eher vom dem

semantischen Wert von 0, als von dem euklidischen Abstand, abhängig. Dies lässt durch das folgende Beispiel am besten verdeutlichen.

Nehmen wir an, dass wir drei binäre MFV, welche in Tabelle 21 definiert sind.

	MFV1	MFV2	MFV3
Action	1	0	1
Adventure	0	1	1
Drama	0	1	1
Fantasy	0	0	1
Sci-Fi	1	0	0
Horror	1	0	0

Tabelle 21: Drei binäre Beispielvektoren für Genres

Gemäß des euklidischen Abstandes werden wir die Distanzwerte in Tabelle 22 erhalten.

	MFV1	MFV2	MFV3
MFV1	-	$\sqrt{5}$	$\sqrt{5}$
MFV2	$\sqrt{5}$	-	$\sqrt{2}$
MFV3	$\sqrt{5}$	$\sqrt{2}$	-

Tabelle 22: Filmähnlichkeit mittels des euklidischen Abstands für binäre Vektoren

Man kann nämlich feststellen, dass $sim(MFV1, MFV2) = sim(MFV1, MFV3)$, obwohl beide Filme, welche durch MFV1 und MFV3 repräsentiert werden, ein Actionfilm sind. Das liegt daran, dass die gemeinsamen Werte 0 und 1 die gleiche Gewichtung haben. Dass zwei Filme nicht den Genre *Fantasy* angehören, ist auch eindeutig ein Ähnlichkeitsmerkmal. Die Frage, ob dieses aber die gleiche Wichtigkeit hat, lässt sich nicht so einfach beantworten.

Würde man verschiedene Maße für die verschiedenen Filmmerkmale benutzen, würde man sie unvergleichbar untereinander machen, daher muss man sich auf ein einzelnes Maß beschränken. Bei den gewichteten Vektoren, wie zum Beispiel bei den Benutzerprofilen, ist dies nicht problematisch, da die 0 eine zusätzliche semantische Bedeutung trägt. Sie kann auch ausdrücken, dass der Benutzer jedes

Merkmal bei Filmen überhaupt nicht mag oder er es noch nicht bewertet hat, was eindeutig die Filmempfehlung beeinflussen soll.

Dieser Zusammenhang, dass Nullwerte auch eine Rolle bei der Ähnlichkeit spielen, lässt sich aber bei binären Vektoren, welche Länder oder Beteiligung von Personen (Regisseur, Schauspieler und Produzent) repräsentieren, nicht mehr rechtfertigen. Dass die restlichen 16.000 Leute und hundert Länder nicht an einem Film beteiligt sind, sagt sehr wenig über die Ähnlichkeit von zwei Filmen aus. Die Distanz wird an sich über das obere Beispiel hinaus nicht mehr dadurch beeinflusst, da nur sich unterscheidende Werte diese vergrößern können. Das Vorhandensein von sehr vielen Nullwerten kann aber sehr gravierenden Folgen bei dem Zusammenführen von den verschiedenen Ähnlichkeitwerten zu einer gemeinsamen Filmähnlichkeit haben, was im nächsten Punkt besprochen wird.

5.4 Bemerkungen zu Vektoren als Repräsentation von Filmmerkmalen

Vektoren haben trotz ihrer bereits erwähnten guten Eigenschaften einen schwerwiegenden Nachteil. Sie lassen sich zwar ohne Weiteres untereinander mittels aller möglichen Maße vergleichen, aber nur solange sie zum gleichen Vektorraum gehören.

Dies bedeutet, dass Vektoren, welche die Genres beschreiben, sich *per se* nicht mit den Vektoren, welche Schauspieler abbilden, vergleichen lassen. Als Folge davon ist auch ihre Distanz nicht vergleichbar, was dazu führt, dass die daraus gebildete Ähnlichkeiten sich nicht zusammenführen lassen. Dies lässt sich am besten erkennen, wenn man die Distanzen auf Prozentwerte normalisiert, um die Ähnlichkeit darzustellen.

Betrachten wir den Vektorraum der Darsteller und nehmen wir an, dass es insgesamt 16.000 Darsteller gibt, d. h. 16.000 Dimensionen. Dabei haben wir zwei Filme, welche durch folgende MFV die Schauspielerbesetzung repräsentieren:

$$MFV1 = (1, \dots, 1, \dots, 1, 1, \dots, 1, 1, 1, \dots, 1, 0)$$

$$MFV2 = (1, \dots, 0, \dots, 0, 1, \dots, 1, 1, 1, \dots, 1, 0)$$

Sie haben sechs gemeinsame Schauspieler von insgesamt acht Schauspielern, die in beiden Filmen spielen. Ihre Distanz wäre also $\sqrt{2}$. Bilden wir nun die Prozentwerte, um die Ähnlichkeit verständlicher zu machen, wobei D die Distanz und D‘ die Ähnlichkeit in Prozenten ist:

$$D' = (D - min) \times \frac{MAXnorm - MINnorm}{max - min} + MINnorm$$

Da wir Prozente brauchen, müssen wir *MAXnorm* und *MINnorm* auf 1 und 0 festlegen. Der Wert für *min* ist die kleinste euklidische Distanz 0. Da wir uns in einem Vektorraum befinden, ist auch die maximale Distanz bekannt. Es ist die Distanz zwischen dem Nullvektor und dem möglichst längsten Vektor, in diesem Fall ein Vektor mit 16.000 1er Stellen. *Max* wäre dann 400. Bilden wir nun D‘:

$D'(MFV1, MFV2) = \left(\sqrt{2} - 0\right) \times \frac{1-0}{400-0} + 0 = 0{,}003535$ oder 99,65% Ähnlichkeit!

Solange wir nur diesen Vektorraum betrachten, ist dies unproblematisch. Diese Ähnlichkeit kann im Vergleich zu den anderen auch die höchste sein, auch wenn sie mit 99,65 % unsinnig erscheint. Das richtige Problem kommt beim Vergleich der Ähnlichkeiten von Filmgenres.

Zu diesen zwei Filmen definieren wir zusätzlich auch zwei MFV, welche die Filmgenres abbilden. Dabei nehmen wir als maximale Anzahl von Filmgenres zum Beispiel 20 an.

$$MFV1 = (0, \dots, 1, \dots, 1, 1, \dots, 0)$$

$$MFV2 = (1, \dots, 1, \dots, 0, 1, \dots, 0)$$

Deren Distanz wäre auch $G = \sqrt{2}$, da sie sich nur in 2 Filmgenres untereinander unterscheiden. Die Ähnlichkeit wäre nun:

$G'(MFV1, MFV2) = \left(\sqrt{2} - 0\right) \times \frac{1-0}{60-0} + 0 = 0{,}0707$ oder 92,93 % Ähnlichkeit!

Man sieht, die Distanzen sind zwar gleich, aber nicht vergleichbar und daher nicht zusammenführbar. Eine mathematisch korrekte Lösung, um Vektoren verschiedener Dimensionen vergleichbar zu machen, kann in der multilinearen Algebra, welche sich u.a. mit zusammengehörigen Vektorpaaren oder sog. Multivektoren beschäftigt, ge-

funden werden. Deren Komplexität wird aber den Rahmen dieser Arbeit sprengen. Andere Lösungen, welche beide Vektorräume zu einem gemeinsamen Vektorraum zusammenführen (z. B. wie die Abbildung eines dreidimensionalen Raumes auf einen zweidimensionalen) und somit alle Vektoren untereinander vergleichbar machen, bringen eine große Verzerrung der Werte mit sich, was die Ergebnisse unbrauchbar machen würde. Es muss also eine andere Lösung gefunden werden.

Bei einer näheren Betrachtung der binären Vektoren und vor allem was sie, als mathematische Abstraktion, darstellen, kann die Ursache dafür und auch den Ansatz für eine Lösung gefunden werden.

Die binären Vektoren stellen dar, welche Merkmale bei einem Film präsent sind – ob er zu diesem oder jenem Genre gehört, ob dieser Schauspieler spielt oder ein anderer. Mathematisch gesehen wäre der längste Vektor, einer, wo alle Merkmale präsent sind, aber in der Realität gibt es keinen Film, in dem alle Personen in der Datenbank mitwirken oder welcher zu allen Filmgenres gehört. Beachtet man ferner, dass man generell bei Schauspielern sehr selten mehr als 10-15 Schauspieler von den Filmverleihen angegeben werden, erweist sich die Bildung der maximalen Distanz, zumindest was binäre Vektoren angeht, durch die Wurzel des längstmöglichen Vektors als Unsinn. Wie bereits erwähnt, kommt eine mathematische Lösung wegen der Komplexität des Problems nicht in Frage, dennoch kann die Statistik einen Ausweg bieten, solange wir die maximale Distanz auf anderem Weg definieren.

Denn was ist die tatsächliche maximale Distanz? Man kann sie für jedes Vektorpaar als die Summe der zwei binären Vektoren definieren oder als Interpretation, wie viele Schauspieler in beiden Filmen spielen. Nehmen wir an, dass wir wieder die oben erwähnten Schauspielervektoren haben:

$$MFV1 = (1, \dots, 1, \dots, 1, 1, \dots, 1, 1, 1, \dots, 1, 0)$$

$$MFV2 = (1, \dots, 0, \dots, 0, 1, \dots, 1, 1, 1, \dots, 1, 0)$$

Die Summe der binären Vektoren wäre: $MFV1 + MFV2 = (1, \dots, 1, \dots, 1, 1, \dots, 1, 1, 1, 1, \dots, 1, 0)$, die maximale Distanz $\sqrt{8}$ und deren Ähnlichkeit:

$D'(MFV1, MFV2) = \left(\sqrt{2} - 0\right) \times \frac{1-0}{\sqrt{8}-0} + 0 = \frac{\sqrt{2}}{\sqrt{8}} = \sqrt{\frac{1}{4}} = 0{,}5$ oder 50 % Ähnlichkeit.

Bei den Genrevektoren hätten wir folgende Ergebnisse:

$$MFV1 = (0, \dots, 1, \dots, 1, 1, \dots, 0)$$

$$MFV2 = (1, \dots, 1, \dots, 0, 1, \dots, 0)$$

Die Summe wäre: $MFV1 + MFV2 = (1, \dots, 1, \dots, 1, 1, \dots, 0)$ und die maximale Distanz $\sqrt{4} = 2$ und deren Ähnlichkeit:

$G'(MFV1, MFV2) = \left(\sqrt{2} - 0\right) \times \frac{1-0}{2-0} + 0 = 0{,}707$ oder 29,3 % Ähnlichkeit!

50 % Ähnlichkeit bei zwei Filmen mit sechs gleichen und zwei unterschiedlichen Schauspielern und 29,3 % Ähnlichkeit bei zwei Filmen mit zwei gleichen und zwei unterschiedlichen Genres sind durchaus Werte, die man zusammenführen kann. Als positiver Nebeneffekt, da gemeinsame Nullwerte nicht die Ähnlichkeit beeinflussen, wird die Interpretation der Ergebnisse bei binären Vektoren im Gegensatz zu reiner euklidischen Distanz als Rankingmaß verbessert.

Nehmen wir an, dass wir drei binären MFV haben, wie in Tabelle 23 definiert.

	MFV1	**MFV2**	**MFV3**
Action	1	0	1
Adventure	0	1	1
Drama	0	1	1
Fantasy	0	0	1
Sci-Fi	1	0	0
Horror	1	0	0

Tabelle 23: Drei binäre Beispielvektoren für Genres

Gemäß des euklidischen Abstandes werden wir die Distanzwerte in Tabelle 24 erhalten.

	MFV1	MFV2	MFV3
MFV1	-	$\sqrt{5}$	$\sqrt{5}$
MFV2	$\sqrt{5}$	-	$\sqrt{2}$
MFV3	$\sqrt{5}$	$\sqrt{2}$	-

Tabelle 24: Filmähnlichkeit mittels des euklidischen Abstands für binäre Vektoren

Obwohl die Distanz zwischen MFV1 und MFV2 gleich der Distanz zwischen MFV1 und MFV3 ist, wird durch die Anwendung des oben erwähnten Verfahrens ihre Ähnlichkeit logisch richtig interpretiert.

$G'(MFV1, MFV2) = \left(\sqrt{5} - 0\right) \times \frac{1-0}{\sqrt{5}-0} + 0 = 1$ oder 0 % Ähnlichkeit, bei null gemeinsamen und fünf unterschiedlichen Genres.

$G'(MFV1, MFV3) = \left(\sqrt{5} - 0\right) \times \frac{1-0}{\sqrt{6}-0} + 0 = 0{,}91$ oder 9 % Ähnlichkeit, bei einem gemeinsamen und fünf unterschiedlichen Genres.

Das Distanzmaß wird an sich nicht verändert, es wird lediglich die Interpretation der Ergebnisse nur auf Vektorpaare beschränkt, um eine nicht mathematische Lösung zu bieten. Dies ist eine zulässige Vereinfachung, da im Grunde genommen die Vektoren hier nur eine Abstraktion von Metadaten sind und einer Interpretation bedürfen, um benutzt zu werden.

Diese Probleme, welche bei der Darstellung von subjektiven Merkmalen, wie Genre oder Ähnlichkeit, durch mathematische Modelle entstehen, werden näher besprochen von Santini und Jain (vgl. Santini, Jain 1999). Sie kommen zu dem Ergebnis, dass mathematische Modelle bei der Betrachtung der Ähnlichkeit von Merkmalen, welche im Normalfall eine subjektive oder psychologiebasierende Betrachtung erfordern (in ihrem Paper handelt es sich um die Ähnlichkeit von Bildern), oft nicht anwendbar sind. In solchen Fällen erwähnen sie als mögliche Lösungen nichtlineare Ähnlichkeitsmaße und Ähnlichkeitsmaße, welche die Dreiecksgleichung nicht erfüllen müssen.

Nachdem die Distanz anhand von prozentualer Ähnlichkeit (auf dem Bereich von 0 bis 1) von Filmpaaren normalisiert ist, können wir die verschiedenen Ähnlichkeiten zu einer gemeinsamen Ähnlichkeit zusammenführen. Diese wird für zwei Filme A und B

auf folgende Weise berechnet, wobei G die Ähnlichkeit von Genres, L von Land, R von Regie, P von Produzenten und D von Darstellern darstellt. Ferner seien x_i die Gewichtungen von den verschiedenen Ähnlichkeiten, mit $\sum_i^n x_i = 1$:

$$sim(A, B) = x_1 \times G + x_2 \times L + x_3 \times R + x_4 \times P + x_5 \times D$$

Diese Gewichtungen können anfangs nicht durch die implizite Datengewinnung bestimmt werden, daher müssen beim Start des Systems schon feste Werte angegeben werden oder die Gewichtungen sind ganz auszulassen bis genügend Daten gesammelt worden sind. Dennoch ist eine subjektive Gewichtung in diesem Fall besser als gar keine, da offensichtlich die verschiedenen Merkmale nicht die gleiche Wichtigkeit bei der Filmauswahl haben. Deshalb wurden nach einem Gespräch mit dem Chefredakteur von *critic.de* folgende Startwerte festgelegt:

Genres: 70 %
Land: 5 %
Regie: 5 %
Produktion: 10 %
Darsteller: 10 %

Nachdem genügend Benutzerbewertungen gesammelt worden sind, werden diese Gewichtungen diesen Daten entsprechend aktualisiert und in gewissen Zeitintervallen automatisch angepasst.

6 Das hybride Empfehlungssystem

Auf der Basis vom oben vorgestellten Datenmodell wird ein zweistufiges Empfehlungssystem vorgeschlagen. Das System ist ein Hybrid aus *content-based* und *collaborative filtering*. Dabei baut das kollaborative Filtern auf dem inhaltsorientierten Abgleich zwischen Inhaltsbeschreibung und Benutzerprofilen auf. Das nachfolgende Abbildung 5 zeigt den hybriden Aufbau des Systems.

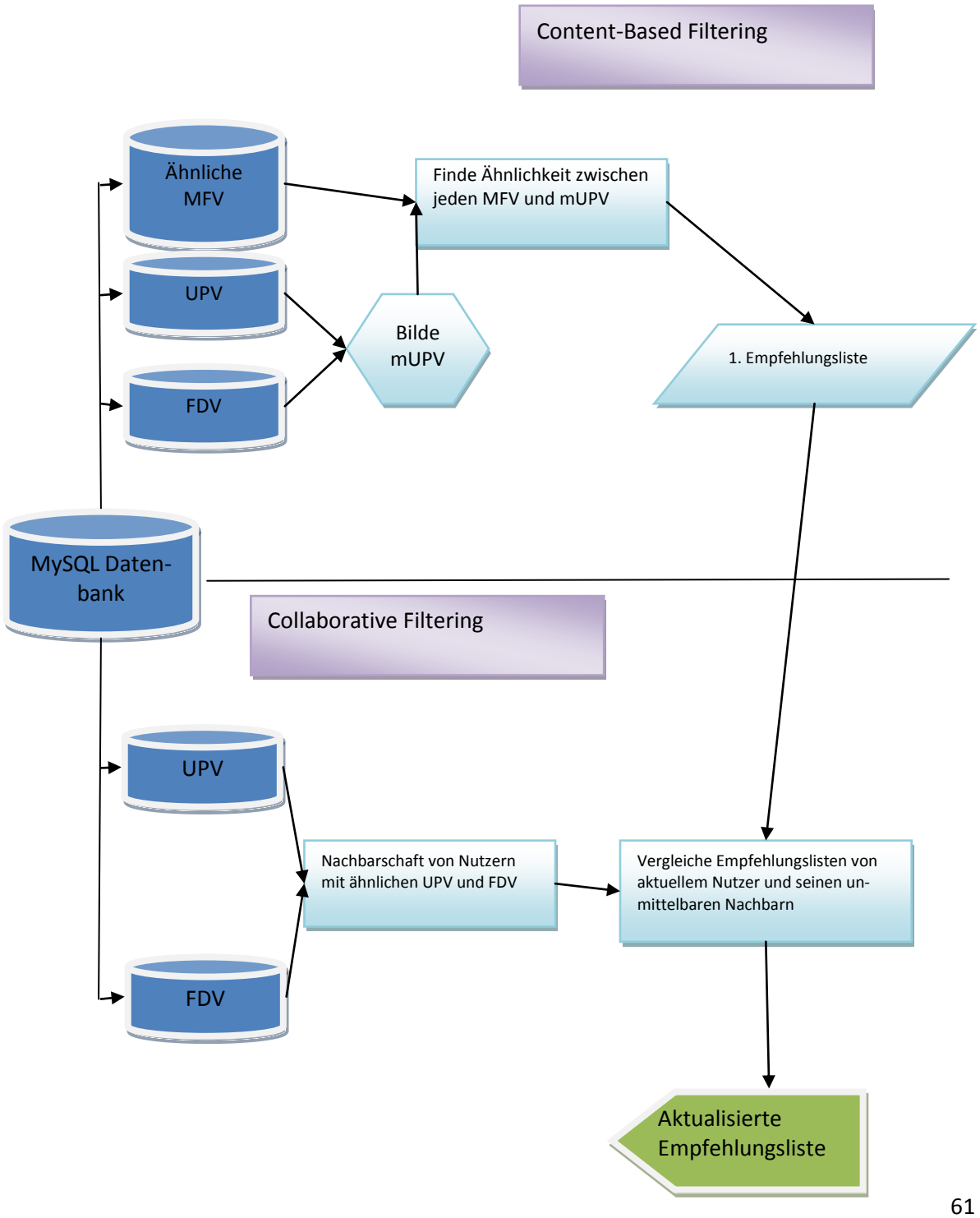


Abbildung 5: Schematische Darstellung des Systemaufbaus (vgl. Maneeroj et al. 2005, S. 185, Abb. 2)

Im oberen Bereich des Diagramms ist der inhaltsbasierte Teil des Verfahrens zu sehen, der zunächst ohne Abgleich mit Benutzerbewertungen auskommt und daher teilweise das Kaltstartproblem vermeidet, da nur die Bewertungen der aktuellen Nutzer notwendig sind. Der Kern wird durch die Korrelationen zwischen den Filmen, welche durch die MFV repräsentiert werden, gebildet. Dieser wird zusätzlich für die Funktion „ähnliche Filme“ benutzt. Wenn der Nutzer Bewertungen abgibt und mit der Seite interagiert, wird sein Benutzerprofil (mUPV) aufgebaut. Danach können beide Elemente verglichen werden: nämlich das Benutzerprofil mit den Filmen, um zu ermitteln, welche Filme am meisten zu diesem passen würden.

Der untere Teil der Grafik zeigt, wie *collaborative filtering* durch Ermittlung ähnlicher Nutzerprofile zu einer modifizierten endgültigen Empfehlungsliste führen kann. Dabei wird der Output von der *content-based filtering* Phase, nämlich die erste Empfehlungsliste, welche durch die Übereinstimmung von Benutzerprofil und Filmähnlichkeiten ermittelt wird, als Input für die zweite Phase *collaborative-filtering* benutzt.

Zuerst werden die bereits in der ersten Phase gebildeten Profile untereinander verglichen, um Nachbarschaften von Nutzern zu bilden. Jetzt können die Empfehlungslisten vom aktuellen Nutzer und seinen am nächsten platzierten Nachbarn untereinander verglichen und dementsprechend bereichert werden.

Nach der Systematik von Burke (vgl. Burke 2002) handelt es sich somit um ein *meta-level hydridization*-Modell (vgl. auch Adomavicius et al. 2007; Balabanovic 1997). Burdek spricht von folgenden Vorteilen:

“*The benefit of the meta-level method, especially for the content/collaborative hybrid is that the learned model is a compressed representation of a user’s interest, and a collaborative mechanism that follows can operate on this information-dense representation more easily than on raw rating data.*” (Burke 2002, S. 340)

Es werden nämlich für das *collaborative filtering* keine zusätzlichen Daten benötigt. Alle relevanten Daten, wie Benutzerprofile und die dazugehörigen Empfehlungslisten, werden schon von dem *content-based filtering* bereitgestellt. Lediglich die Profile werden nur nach Ähnlichkeit verglichen. Da aber diese Profile auf der Basis von Metadaten aufgebaut worden sind (Bewertungen über Filmgenres, Schauspieler

etc.), ist es nun möglich den „Filmgeschmack“ von Nutzern zu erkennen, was über den üblichen kollaborativen Vergleich von gemochten und nicht gemochten Filmen hinausgeht, weil der Ähnlichkeitsvergleich nicht eindimensional ist.

Was enthalten aber die sogenannten ersten Empfehlungslisten? Sie enthalten die bereits hoch bewerteten Filme von dem jeweiligen Nutzer und die noch nicht gesehenen Filme, welche durch das Empfehlungssystem als sehenswert für den Nutzer markiert sind. Dabei ist in der ersten Phase für den Nutzer nur der noch nicht gesehene Teil der empfohlenen Filme sichtbar.

Die aktualisierte Empfehlungsliste wird durch die eigenen noch nicht gesehenen und vom System empfohlenen Filme zusammen mit den am höchsten bewerteten Filmen von den Nutzern, die die ähnlichsten Benutzerprofile aufweisen, zusammengestellt. Außerdem werden auch Filme berücksichtigt, welche sogar von beiden Nutzern noch nicht bewertet worden sind, aber dennoch durch ihre Filmmerkmale als sehr sehenswert eingestuft worden sind. Dies dient allein dem Zweck die Empfehlungen divers und interessant zu halten. Deshalb soll dieses Verfahren gezielt gegen die Tendenz zum Überspezifizieren von Empfehlungen, welche dem *content-based filtering* eigen ist, entgegenwirken.

So ein hybrides System, welches auf der Basis von inhaltsbasiertem Filtern aufgebaut wird, hat ferner für die kleinen und mittelgroßen Seiten den Vorteil, dass es dennoch Empfehlungen generieren kann, während genügend Bewertungen für das kollaborative Filtern akkumuliert werden.

6.1 Ein Anwendungsbeispiel

An dem Beispiel von zwei Filmen soll nun die Funktionsweise des Systems verdeutlicht werden. Zunächst werden die Beschreibungsvektoren durch den Einsatz des Datenmodells befüllt (vgl. Chapphannarungsri 2009, S. 700, Tab. 1).

Für „The Matrix“ gelten folgende Beschreibungsmerkmale: Genre: *Action, Adventure, Sci-Fi, Thriller*; Land: *USA*; Regisseur: *Andy* und *Lana Wachowski*; Schauspieler:

Keanu Reeves, *Laurence Fishburne*, *Carrie-Anne Moss* und viele andere. Dies wird wie folgt in Tabelle 25 als *movie feature vector* (MFV) repräsentiert.

Feature(1): Genre	1	1	0	0	....	0	0	1	1	0	0
Feature(2): Regisseur	0	0	0	0	....	0	0	1	1	0	0
Feature(3): Schauspieler	0	...	0	...	1	..	1	0.	...	1	0
Feature(4): Land	0	0	0	0	0	0	...	0	0	1	0

Tabelle 25: MFV für „The Matrix“

Im zweiten Beispiel gelten für „Constantine“ folgende Werte: Genre: *Fantasy, Horror, Thriller*; Land: *USA*; Regisseur: *Francis Lawrence*. Schauspieler sind: *Keanu Reeves*, *Rachel Weisz*, *Shia LaBeouf u.v.a.* Es ergibt sich in Tabelle 26 folgender MFV.

Feature(1): Genre	0	0	0	...	1	0	1	...	1	0	0
Feature(2): Regisseur	0	0	0	...	0	...	1	...	0	0	0
Feature(3): Schauspieler	0	...	1	...	0	..	0	1	...	1.	0
Feature(4): Land	0	0	0	0	0	0	...	0.	0	1	0

Tabelle 26: MFV für „Constantine“

Die nachfolgende Abbildung 6 zeigt exemplarisch die Berechnung des Benutzerprofils auf der Basis der hybriden Architektur des Systems (als Beispiel werden die oben dargestellten Genre-MFV von den Filmen „The Matrix“ und „Constantine“ benutzt):

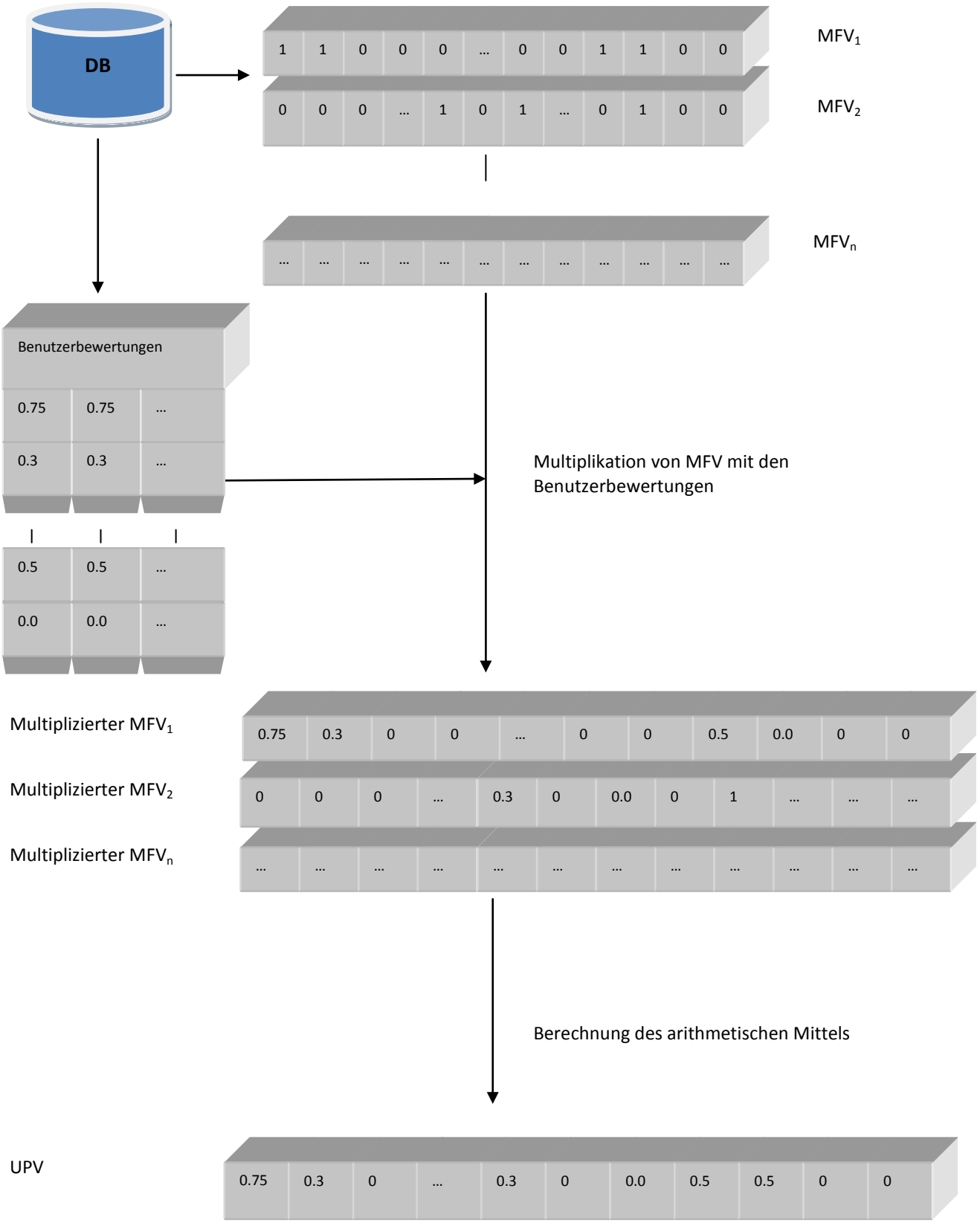


Abbildung 6: Berechnung der Benutzerprofile (vgl. Chapphannarungsri 2009, S. 700, Abb. 2)

Durch die Zusammensetzung von den Filmmerkmalen und deren Bewertungen seitens des Benutzers wird der erste Teil des Benutzerprofils (UPV) explizit gebildet. Nun kann dieses mit dem impliziten Benutzerprofil (FDV) vereint werden, um das komplette Benutzerprofil zu erstellen. Dies wird in Abbildung 7 grafisch dargestellt.

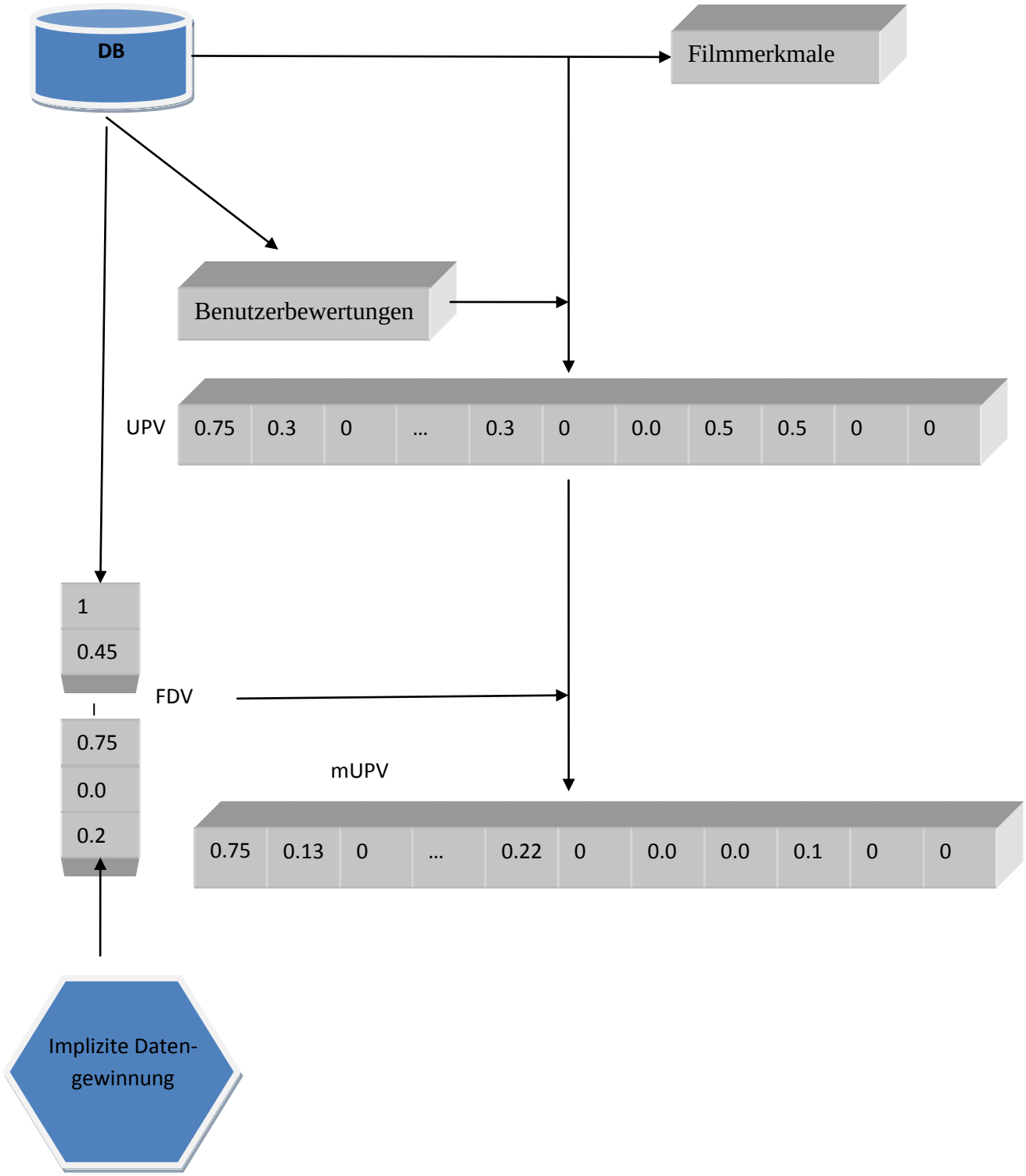


Abbildung 7: Berechnung vom Benutzerprofil anhand von impliziter und expliziter Datengewinnung

Nun stehen beide Datensätze, die für die Generierung von Filmempfehlungen notwendig sind bereit nämlich die Beschreibungsvektoren von Filmen und Nutzern. Die ersten Empfehlungen werden dann auf der Basis eines Ähnlichkeitsvergleichs, nach dem Verfahren des *content-based filtering* zwischen Benutzerprofil und den Beschreibungsvektoren von Filmen, generiert.

Wenn später im System genügend Benutzerprofile existieren, können die Nachbarn des jeweiligen Nutzers ermittelt werden, und mittels *collaborative filtering* die ersten Empfehlungen verbessert werden. Die Generierung der ersten Empfehlungsliste wird in Abbildung 8 grafisch verdeutlicht.

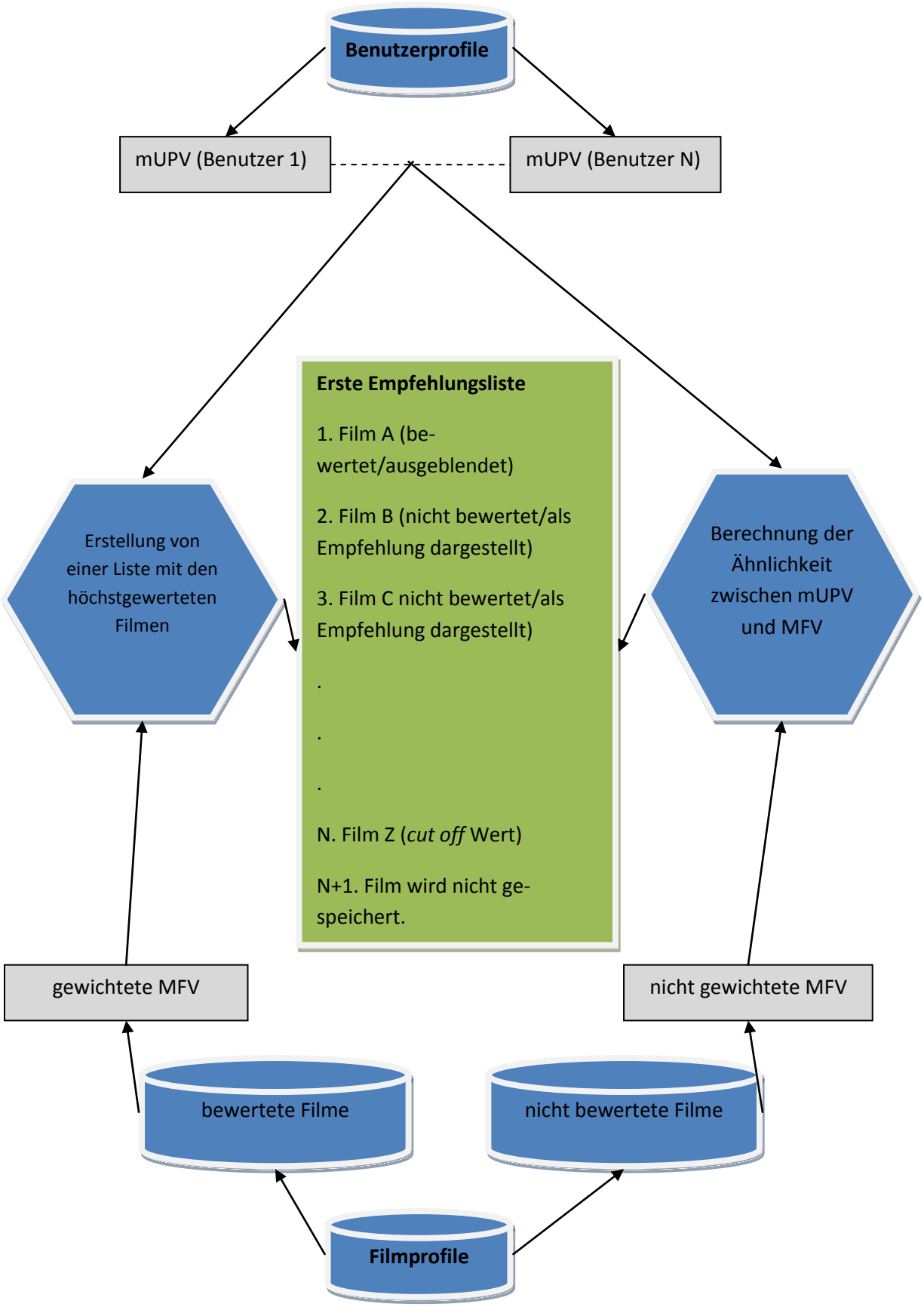


Abbildung 8: Generierung der ersten Empfehlungsliste (*content-based filtering*)

Nachdem die ersten Empfehlungslisten für alle Nutzer ermittelt worden sind, kann man diese mit *collaborative filtering* verbessern. Grundvoraussetzung ist, dass bereits genügend Profile gesammelt worden sind. Bis zu diesem entscheidenden Punkt werden die Empfehlungen, um eine Grundfunktionalität für die Nutzer zu gewährleisten, durch diese ersten Listen vertreten. Die endgültigen Empfehlungslisten werden, wie in Abbildung 9 schematisiert, erstellt.

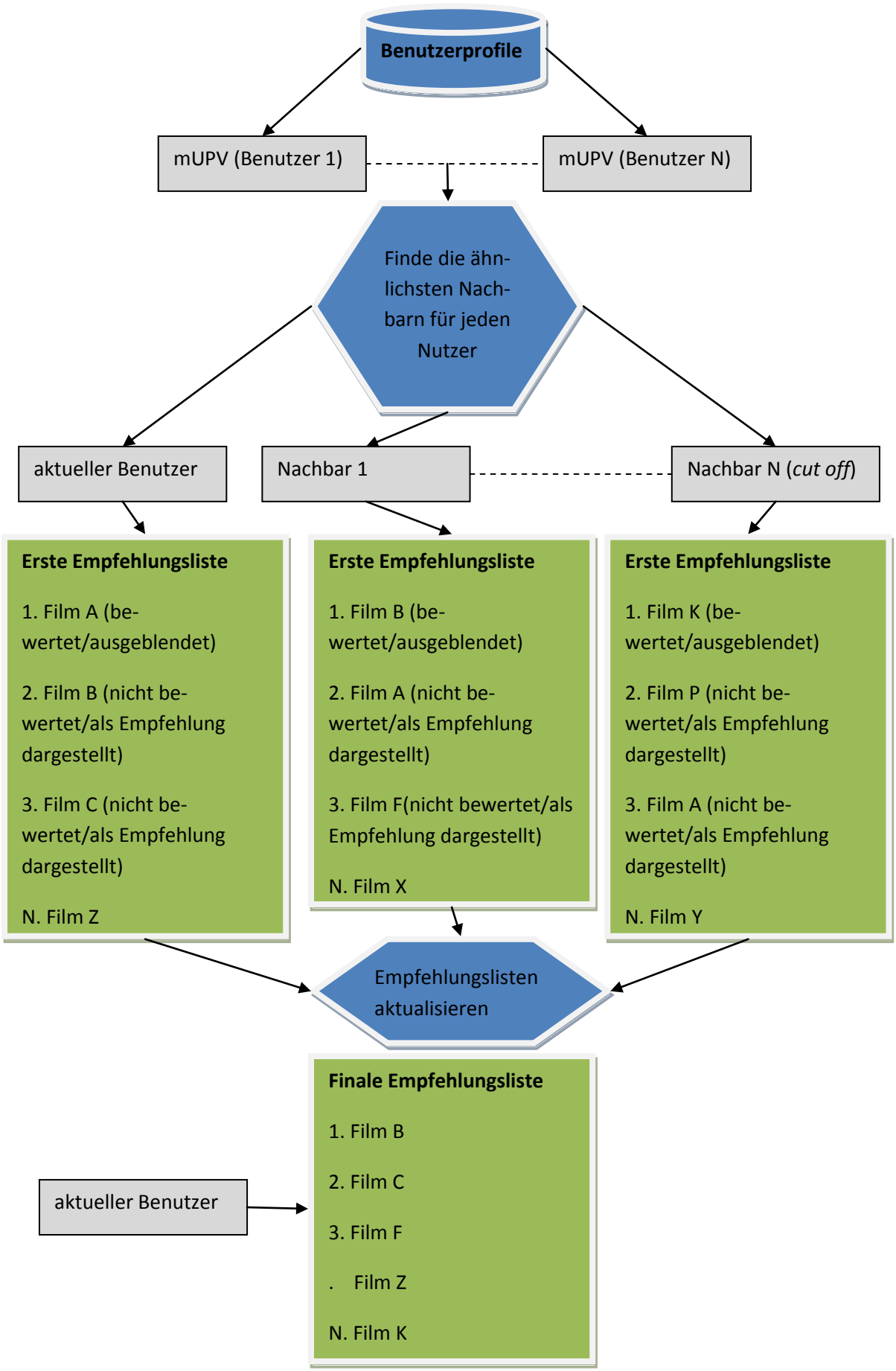


Abbildung 9: Generierung der ersten Empfehlungsliste (*collaborative filtering*)

6.2 Implementierung

Zwar wurde eine komplette praktische Realisierung dieses Verfahrens geplant, konnte aber wegen des verspäteten Relaunchs der Seite nicht während dieser Arbeit als Ganzes in der Praxis umgesetzt werden. Programmiert und getestet wurden das vorgeschlagene Datenmodell in Form von Vektoren als Repräsentationsmerkmale, deren Umsetzung in MySQL und die Berechnung, ferner die Bildung von Filmähnlichkeiten um das Ähnlichkeitsmaß zu testen.

Da die Vergleichsoperationen über die Vektoren speicher- und rechenintensiv sind, wurde zunächst zu Testzwecken eine *offline*-Berechnung auf einer separaten Maschine realisiert. Diese benötigte ca. 5 bis 8 Minuten für die Berechnung aller Datensätze bei ca. 1 bis 2 GB Arbeitsspeicher, indem die vollständigen Matrizen in den Arbeitsspeicher geladen wurden. Diese vorläufigen Tests wurden mittels der objektorientierten Datenbanksprache FoxPro[14] realisiert.

Diese Sprache hat als Vorteil, dass es keine der aus anderen Programmiersprachen bekannten Restriktionen (z. B. einheitliche Daten im *array* oder maximale Länge) bezüglich des Datentyps *array* hat, was bei der Arbeit mit Vektoren besonders vorteilhaft ist.

Da sich die Datenbank aber auf einem gemieteten Server befindet, der nur lokalen Zugriff erlaubt, war eine Lösung erforderlich, bei der die Berechnungsskripte auf dem Server laufen. Der Server wurde in der Relaunchphase auf Linux umgestellt, daher war der Einsatz des Testprogramms unmöglich, da Anwendungen, welche mit FoxPro, eine Microsoft-Entwicklung, erstellt worden sind, nur auf Windows laufen können. Daher musste das Programm in Form von Skripten neu gestaltet werden. Als Restriktionen gelten dabei ein maximal verfügbarer Arbeitsspeicher von 256 MB und eine maximale Ausführungszeit von 4200 sec pro Script.

Für die oben konzeptuell erläuterten Verfahren wurde anschließend eine Realisierung als PHP-Routinen geplant. Für die Speicherung der Vektoren kommt eine MySQL-Datenbank zum Einsatz. Der Systemaufbau der Filmähnlichkeit wird in Abbildung 10 gezeigt.

[14] Die offizielle Seite von FoxPro. Letzter Zugriff: 20.11.2010, unter http://msdn.microsoft.com/de-de/vfoxpro/bb190291.aspx.

PHP wurde für die Implementierung gewählt, da hier umfangreiche Funktionen zur Arbeit mit MySQL bereitstehen und die Skripte auf dem Server unter Unix sich leicht durch Cronjobs steuern lassen. Außerdem war PHP erforderlich, da die bisherige Programmierung der Seite (Routinen, Plug-ins etc.) auch auf dieser Sprache beruhte. PHP hat aber den Nachteil der speicherintensiven Repräsentation von strukturierten Datentypen[15]. Vollständige Abbildung umfangreicher Datenmatrizen im Arbeitsspeicher (wie in der ersten Testphase) sind deswegen zu vermeiden.

Für diesen Ansatz konnte dies dadurch gelöst werden, dass nur die Matrizen, welche die Genrevektoren, Ländervektoren etc. enthalten, in MySQL gespeichert wurden. Die Speicherung der kompletten Relationsmatrix, die alle Film-Film-Ähnlichkeiten enthält, wurde nach ersten Tests verworfen, da sie zu viel Prozessorzeit und Arbeitsspeicher beansprucht (mehr als $6*10^6$ Ähnlichkeiten plus deren Indexierung). Da davon auszugehen ist, dass der Endnutzer nie *alle* Relationen brauchen wird, werden nur die 25 besten Treffer mit der höchsten Ähnlichkeit gespeichert. Deren Erzeugung erfolgt stufenweise, da nur einzelne Paare von Vektoren in den Arbeitsspeicher geladen werden, um die Auslastung des Arbeitsspeichers zu reduzieren. Nach Berechnung der euklidischen Distanz als Ähnlichkeitsmaß zwischen den Vektoren wird das Resultat mittels einer Variante von *Insertion Sort* mit binärer Suche in die Ergebnisliste eingefügt.

Befürchtungen, dass die euklidische Distanz zu viel Zeit für die Berechnungen in Anspruch nehmen könnte, da sie Quadrieren als sehr rechenaufwendige Operation enthält, haben sich bei den vorliegenden Größenordnungen[16] nicht bestätigt.

Dennoch sind bei viel größeren Datenmengen auch andere Distanzmaße einsatzfähig, wie etwa die *City Block* Distanz (vgl. Krause 1986). Als Engpass haben sich vielmehr die *read*- und *write*-Zugriffe auf dem Server erwiesen, was noch ein weiterer Grund für den Verzicht auf die komplette Relationsmatrix war.

[15] In einem PHP-Array umfasst ein einzelnes *integer*-Element 68 Byte.

[16] 2000 Genrevektoren mit jeweils 6 Elementen im Durchschnitt und insgesamt $4*10^6$ Distanzberechnungen.

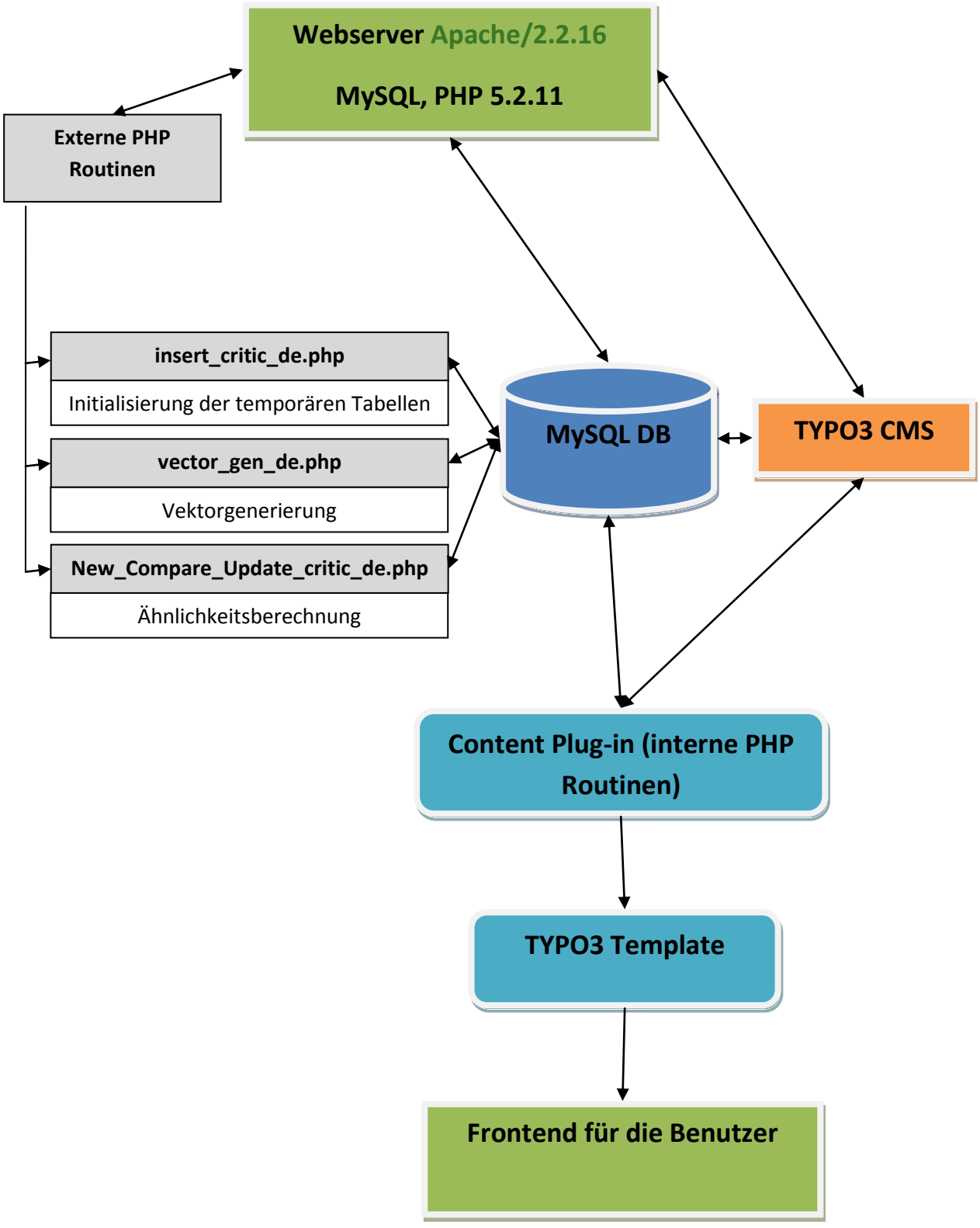


Abbildung 10: Systemaufbau für die Filmähnlichkeit

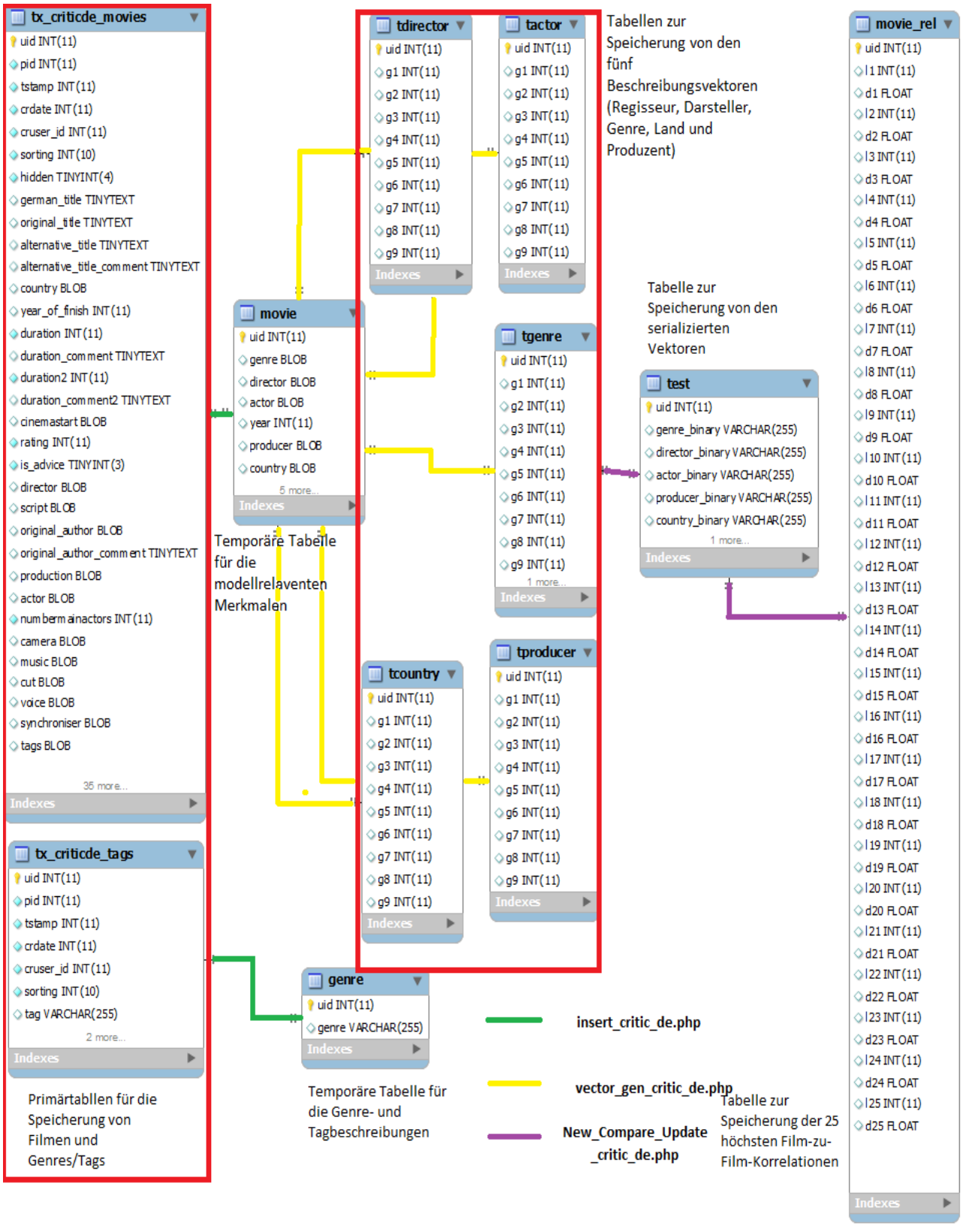


Abbildung 11: Interaktion der MySQL-Tabellen mit den PHP-Skripten

Abbildung 11 zeigt, wie der Filmvergleich nach Ähnlichkeit in drei Schritten erfolgt. Erstens werden nur die modellrelevanten Daten von den Primärtabellen (*tx_critic_de_movies* und *tx_critic_de_tags*) in den temporären Tabellen (*movie* und

genre) zwischengespeichert. Zweitens werden die Vektoren für die fünf Filmmerkmalen in den entsprechenden Tabellen gespeichert (*tdirector, tactor, tgenre, tcountry* und *tproducer*). Drittens werden die Vektoren in Form von serializierten *arrays* in der Tabelle *test* gespeichert und anhand eines Vergleiches dieser Vektoren wird die Film-zu-Film-Korrelationstabelle *movie_rel* erstellt[17].

Die Darstellung der Filmähnlichkeit im Frontend wurde in zwei Varianten vorgeschlagen, wobei endgültig nur eine bestehen wird. Einmal wurde sie als interaktiver Box unter jeder Kritik (Punkt 1 auf der Abb. 12) realisiert und einmal als eine Liste in der rechten Spalte der Seite (Punkt 2 auf der Abb. 12). Man hat sich für die Bezeichnung der Ähnlichkeit im Frontend als „Film-Vernetzung“ entschieden, da „ähnliche Filme“ für die redaktionelle Ähnlichkeit von Filmen (z. B. double Features) vorgesehen worden ist.

[17] Alle MySQL- und PHP-Daten zusammen mit einer Anleitung zur Aufstellung eines Testsystems zu wissenschaftlichen Zwecken können auf dem als Anlage beigefügten CD gefunden werden.

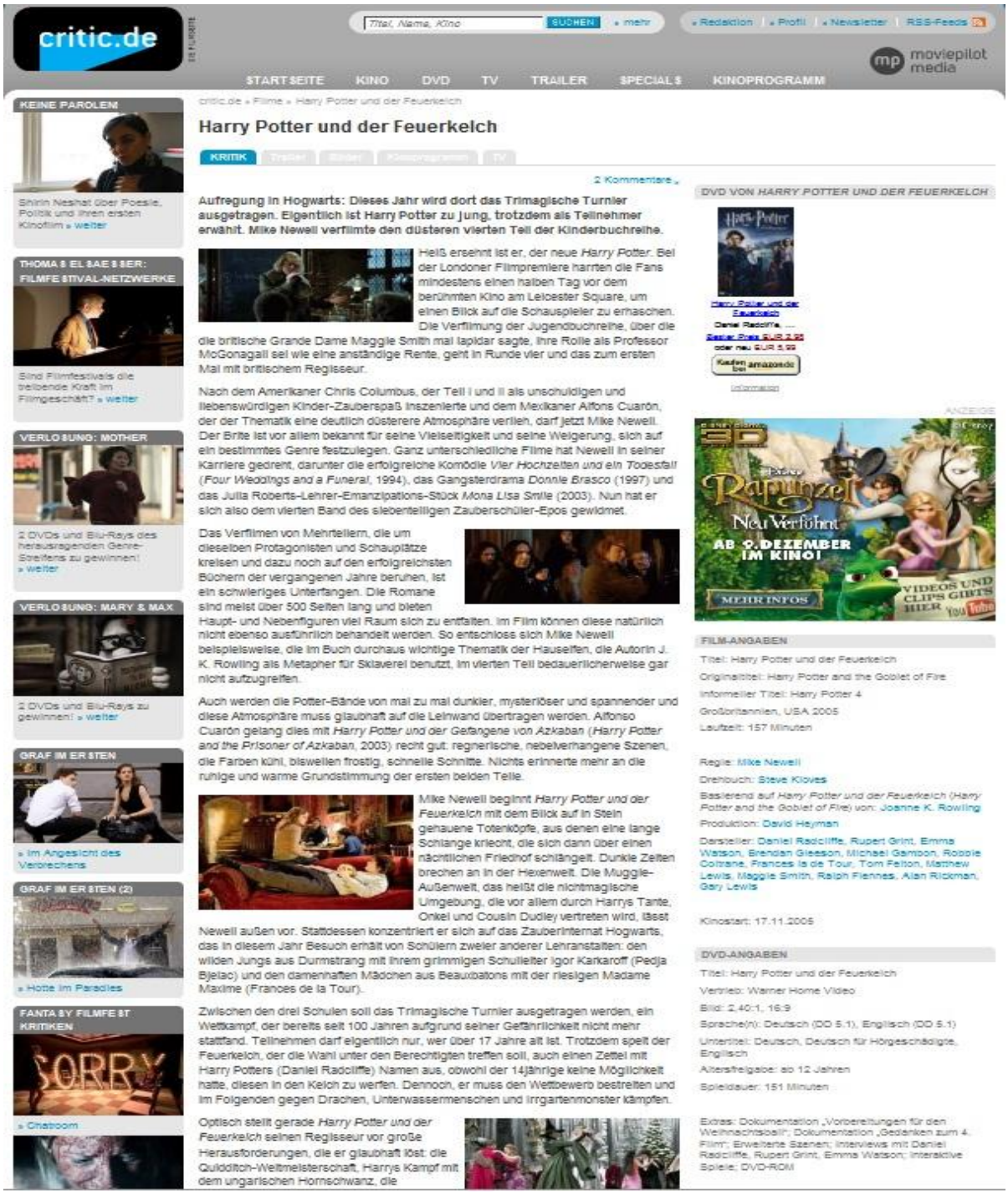

critic.de
Titel, Name, Kino
SUCHEN
» mehr
» Redaktion | » Profil | » Newsletter | RSS-Feeds
moviepilot media
START SEITE
KINO
DVD
TV
TRAILER
SPECIALS
KINOPROGRAMM
KEINE PAROLEN
Shirin Neshat über Poesie, Politik und ihren ersten Kinofilm » weiter
THOMAS EL SAESSER: FILMFESTIVAL-NETZWERKE
Sind Filmfestivals die treibende Kraft im Filmgeschäft? » weiter
VERLOSUNG: MOTHER
2 DVDs und Blu-Rays des herausragenden Genre-Streifens zu gewinnen! » weiter
VERLOSUNG: MARY & MAX
2 DVDs und Blu-Rays zu gewinnen! » weiter
GRAF IM ERSTEN
» Im Angesicht des Verbrechens
GRAF IM ERSTEN (2)
» Hotte im Paradies
FANTASY FILMFEST KRITIKEN
SORRY
» Chatroom
critic.de » Filme » Harry Potter und der Feuerkelch
Harry Potter und der Feuerkelch
KRITIK
2 Kommentare »
Aufregung in Hogwarts: Dieses Jahr wird dort das Trimagische Turnier ausgetragen. Eigentlich ist Harry Potter zu jung, trotzdem als Teilnehmer erwählt. Mike Newell verfilmte den düsteren vierten Teil der Kinderbuchreihe.
Heiß ersehnt ist er, der neue Harry Potter. Bei der Londoner Filmpremiere harrten die Fans mindestens einen halben Tag vor dem berühmten Kino am Leicester Square, um einen Blick auf die Schauspieler zu erhaschen. Die Verfilmung der Jugendbuchreihe, über die die britische Grande Dame Maggie Smith mal lapidar sagte, ihre Rolle als Professor McGonagall sei wie eine anständige Rente, geht in Runde vier und das zum ersten Mal mit britischem Regisseur.
Nach dem Amerikaner Chris Columbus, der Teil I und II als unschuldigen und liebenswürdigen Kinder-Zauberspaß inszenierte und dem Mexikaner Alfons Cuarón, der der Thematik eine deutlich düsterere Atmosphäre verlieh, darf jetzt Mike Newell. Der Brite ist vor allem bekannt für seine Vielseitigkeit und seine Weigerung, sich auf ein bestimmtes Genre festzulegen. Ganz unterschiedliche Filme hat Newell in seiner Karriere gedreht, darunter die erfolgreiche Komödie Vier Hochzeiten und ein Todesfall (Four Weddings and a Funeral, 1994), das Gangsterdrama Donnie Brasco (1997) und das Julia Roberts-Lehrer-Emanzipations-Stück Mona Lisa Smile (2003). Nun hat er sich also dem vierten Band des siebenteiligen Zauberschüler-Epos gewidmet.
Das Verfilmen von Mehrteilern, die um dieselben Protagonisten und Schauplätze kreisen und dazu noch auf den erfolgreichsten Büchern der vergangenen Jahre beruhen, ist ein schwieriges Unterfangen. Die Romane sind meist über 500 Seiten lang und bieten Haupt- und Nebenfiguren viel Raum sich zu entfalten. Im Film können diese natürlich nicht ebenso ausführlich behandelt werden. So entschloss sich Mike Newell beispielsweise, die im Buch durchaus wichtige Thematik der Hauselfen, die Autorin J. K. Rowling als Metapher für Sklaverei benutzt, im vierten Teil bedauerlicherweise gar nicht aufzugreifen.
Auch werden die Potter-Bände von mal zu mal dunkler, mysteriöser und spannender und diese Atmosphäre muss glaubhaft auf die Leinwand übertragen werden. Alfonso Cuarón gelang dies mit Harry Potter und der Gefangene von Azkaban (Harry Potter and the Prisoner of Azkaban, 2003) recht gut: regnerische, nebelverhangene Szenen, die Farben kühl, bisweilen frostig, schnelle Schnitte. Nichts erinnerte mehr an die ruhige und warme Grundstimmung der ersten beiden Teile.
Mike Newell beginnt Harry Potter und der Feuerkelch mit dem Blick auf in Stein gehauene Totenköpfe, aus denen eine lange Schlange kriecht, die sich dann über einen nächtlichen Friedhof schlängelt. Dunkle Zeiten brechen an in der Hexenwelt. Die Muggle-Außenwelt, das heißt die nichtmagische Umgebung, die vor allem durch Harrys Tante, Onkel und Cousin Dudley vertreten wird, lässt Newell außen vor. Stattdessen konzentriert er sich auf das Zauberinternat Hogwarts, das in diesem Jahr Besuch erhält von Schülern zweier anderer Lehranstalten: den wilden Jungs aus Durmstrang mit ihrem grimmigen Schulleiter Igor Karkaroff (Predja Bjelac) und den damenhaften Mädchen aus Beauxbatons mit der riesigen Madame Maxime (Frances de la Tour).
Zwischen den drei Schulen soll das Trimagische Turnier ausgetragen werden, ein Wettkampf, der bereits seit 100 Jahren aufgrund seiner Gefährlichkeit nicht mehr stattfand. Teilnehmen darf eigentlich nur, wer über 17 Jahre alt ist. Trotzdem spielt der Feuerkelch, der die Wahl unter den Berechtigten treffen soll, auch einen Zettel mit Harry Potters (Daniel Radcliffe) Namen aus, obwohl der 14jährige keine Möglichkeit hatte, diesen in den Kelch zu werfen. Dennoch, er muss den Wettbewerb bestreiten und im Folgenden gegen Drachen, Unterwassermenschen und Irrgartenmonster kämpfen.
Optisch stellt gerade Harry Potter und der Feuerkelch seinen Regisseur vor große Herausforderungen, die er glaubhaft löst: die Quidditch-Weltmeisterschaft, Harrys Kampf mit dem ungarischen Hornschwanz, die
DVD VON HARRY POTTER UND DER FEUERKELCH
Harry Potter und der Feuerkelch
Daniel Radcliffe, ...
Kaufen bei amazon.de
Information
ANZEIGE
Disney 3D Blu-ray
Rapunzel
Neu Verföhnt
AB 9. DEZEMBER IM KINO!
MEHR INFOS
VIDEOS UND CLIPS GIBTS HIER
FILM-ANGABEN
Titel: Harry Potter und der Feuerkelch
Originaltitel: Harry Potter and the Goblet of Fire
Informeller Titel: Harry Potter 4
Großbritannien, USA 2005
Laufzeit: 157 Minuten
Regie: Mike Newell
Drehbuch: Steve Kloves
Basierend auf Harry Potter und der Feuerkelch (Harry Potter and the Goblet of Fire) von: Joanne K. Rowling
Produktion: David Heyman
Darsteller: Daniel Radcliffe, Rupert Grint, Emma Watson, Brendan Gleeson, Michael Gambon, Robbie Coltrane, Frances la de Tour, Tom Felton, Matthew Lewis, Maggie Smith, Ralph Fiennes, Alan Rickman, Gary Lewis
Kinostart: 17.11.2005
DVD-ANGABEN
Titel: Harry Potter und der Feuerkelch
Vertrieb: Warner Home Video
Bild: 2,40:1, 16:9
Sprache(n): Deutsch (DD 5.1), Englisch (DD 5.1)
Untertitel: Deutsch, Deutsch für Hörgeschädigte, Englisch
Altersfreigabe: ab 12 Jahren
Spieldauer: 151 Minuten
Extras: Dokumentation „Vorbereitungen für den Weihnachtsball"; Dokumentation „Gedanken zum 4. Film"; Erweiterte Szenen; Interviews mit Daniel Radcliffe, Rupert Grint, Emma Watson; Interaktive Spiele; DVD-ROM

Abbildung 12: Darstellung der Filmähnlichkeit im Frontend (Harry Potter und der Feuerkelch). Letzter Zugriff: 17.01.2011, unter http://www.critic.de/film/harry-potter-und-der-feuerkelch-362/

„Film-Vernetzung“ ist zugleich auch ein Hyperlink, der zu einer Liste mit den 25 ähnlichsten Filmen führt. Diese Liste ist absteigend geordnet. Man hat sich gegen eine quantitative Darstellung der Ergebnisse (Ähnlichkeitsgrad in Prozenten, Sternchen etc.) entschieden, da diese Darstellung, auch eine Erklärung für die Benutzer erfordert, warum und wie man auf diese Zahl oder Ähnlichkeit kommt. Eine semantische Darstellung wurde vorgezogen, da diese bildlich die Ähnlichkeit darstellen kann und keine detaillierte Erklärung über Formel und Datenmodelle erfordert.

Die Filmdaten, die für die Ähnlichkeit relevant sind, werden unter jedem Film aufgelistet (Genres/Tags, Regie, Produzent, Land und Darsteller; Punkt 1 auf der Abb. 13) und die Übereinstimmungen mit dem ursprünglichen Film (Punkt 2 auf der Abb. 13) fett markiert. Auf diese Art und Weise wird dem Nutzer auch erlaubt, sich selbst ein Bild über die Ähnlichkeit zu bilden, indem er die relevanten Daten zu sehen bekommt. Die Liste ist jeweils in fünf mal fünf Filmen geteilt, um eine übersichtliche Darstellung auf der Seite zu gewährleisten.

Die tatsächliche Umsetzung auf dem Server erlaubte ferner einen Einblick in die Performanz des Datenmodells und des Ähnlichkeitsalgorithmus‘ in Form von der euklidischen Distanz. Die Bildung des gesamten Datenmodells, aller Ähnlichkeiten (ca. $6*10^6$ Filmvergleiche mit jeweils 5 Vektoren) und deren Abspeicherung benötigte nur 8 Minuten, bei 50 % Belastung und ca. 200 MB RAM (im Vergleich zum Testsystem mit 90 Minuten). Weder die euklidische Distanz noch das Datenmodell erwiesen sich bei der Anwendung auf diese Datenmenge als zu ressourcenintensiv.

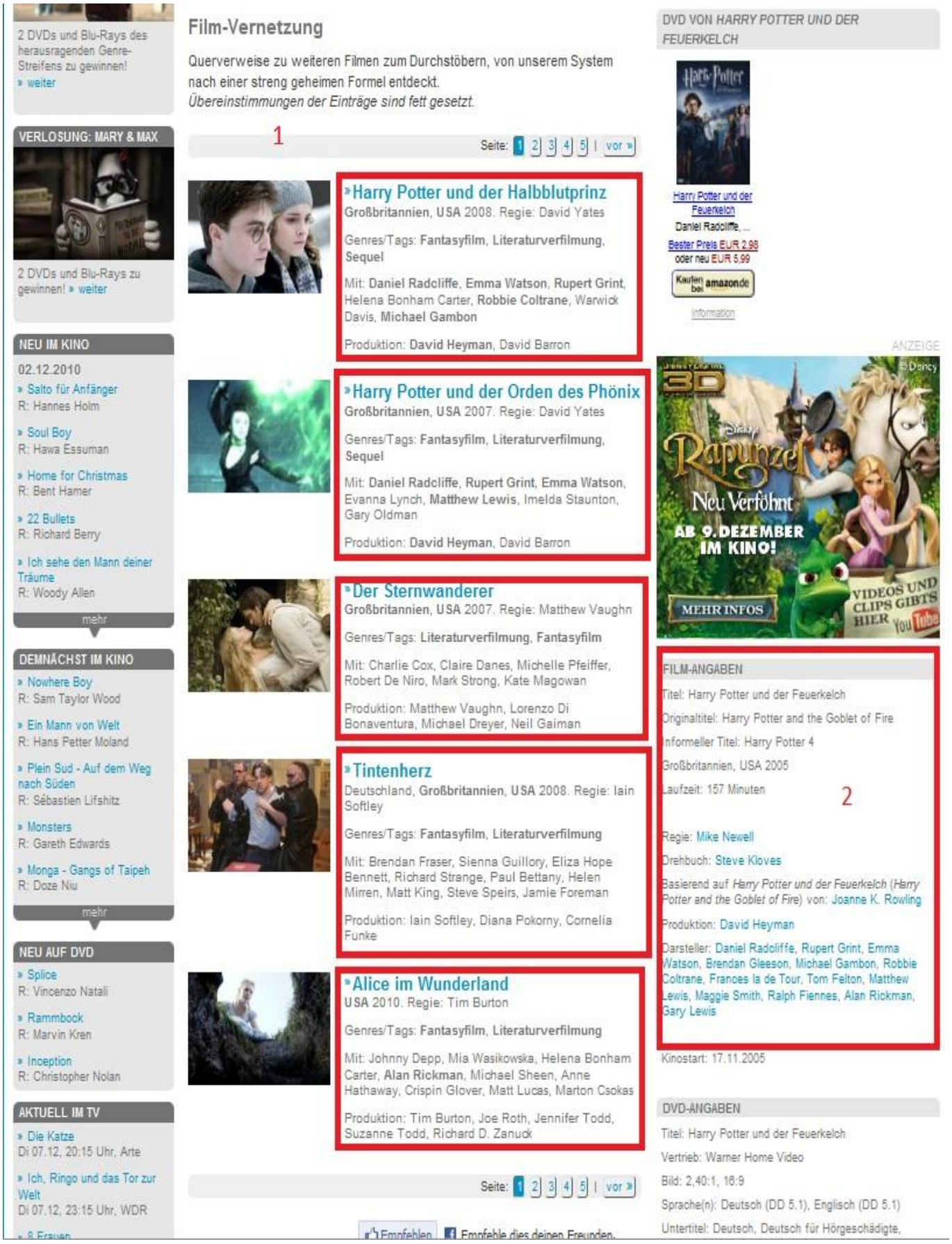


Abbildung 13: Auflistung der ähnlichsten Filme (Harry Potter und der Feuerkelch). Letzter Zugriff: 17.01.2011, unter http://www.critic.de/film/harry-potter-und-der-feuerkelch-362/film-vernetzung/

7 Ausblick

Neben der sachlichen Auseinandersetzung mit dem Thema und der Implementierung des vorgeschlagenen Datenmodells gibt es einige Anmerkungen und Entwicklungen auf dem Feld von Empfehlungssystemen, die nach einer ausführlicheren Diskussion verlangen. Damit soll ein Überblick über Verbesserungsvorschläge und weitere Implementierungsmöglichkeiten für zukünftige Untersuchungen auf diesem Gebiet verschafft werden.

Collaborative filtering ist im Gegensatz zum inhaltsbasierenden Filtern in vielen Hinsichten verbesserungsfähig. Da es sich um ein Verfahren der Statistik handelt, kann dieses auch durch derer Methodik verbessert werden. Dabei ist es am wichtigsten das Signalrauschen (eng. *noise*) zu reduzieren oder zu isolieren, um die tatsächlichen langfristigen Tendenzen zu ermitteln. *Content-based filtering* lässt sich lediglich durch das Ähnlichkeitsmaß, die Qualität und Auswahl der Beschreibungsmerkmale verbessern.

Neuere Technologien und Techniken sowohl im Hardwarebereich als auch in der Programmierung machen zudem Vektoren praktikabler als früher. Vor allem die Verschiebung von Vektorberechnungen vom CPU auf den GPU hat zu phänomenalen Ergebnissen geführt[18]. Diese Fortschritte werden in der Zukunft erlauben, dass immer mehr Daten und aufwendigere Algorithmen für Empfehlungssysteme zur Verfügung stehen können.

7.1 Subjektive Bewertungen im *collaborative filtering*

Bewertungen auf einer Skala als Beschreibung von Zufriedenheit oder Unzufriedenheit eines Benutzers sind an sich nicht absolut zuverlässig. Sie lassen sich von sehr vielen Faktoren positiv oder negativ beeinflussen: Umgebung, Laune des Benutzers und die Reihenfolge der zu bewertenden Elemente haben zum Beispiel eine profunde nachgewiesene Auswirkung auf das menschliche Einschätzungsvermögen (vgl. Masthoff 2005, S. 305; Bell et al. 2008, S.1).

[18] Selbst Intel, der marktführende CPU-Hersteller, hat in einem Versuch, dies zu widerlegen, bestätigt, dass GPUs „nur“ bis 14 Mal besser als CPUs für Vektorberechnungen sind (Vgl. Lee 2010).

Diese Faktoren sind besonders wichtig bei der Bewältigung der Kaltstartproblematik für neue Benutzer. Ihr Profil muss im Idealfall mit dem geringsten Aufwand, schnell und zuverlässig erstellt werden. Daher werden Benutzer meistens mit einer zufälligen oder nicht zufälligen Auswahl von Filmen präsentiert, welche sie bewerten müssen (wie z. B. *moviepilot.de*). Dennoch können die Anzahl an Filmen, deren Präsentationsreihenfolge und Inhalt die Ergebnisse beeinflussen. Daher muss zumindest eine möglichst neutrale Darstellung der Filme (z. B. ohne Zusatzangaben wie Regie oder Jahr, weil diese direkt das Empfinden beeinflussen können), benutzt werden. Die Liste muss zufallsgeneriert sein und die Filme dürfen thematisch nicht miteinander verbunden sein (vgl. Masthoff 2005, S. 305).

In dieser Arbeit wird folgende Vorgehensweise vorgeschlagen. Ein Teil der Kaltstartproblematik lässt sich auch durch ein Stereotypsystem (vgl. Montaner 2003, S. 296) bewältigen, wobei die oben erwähnten Faktoren sich umgehen lassen. Der Benutzer wird dabei mit einer Liste von Benutzerstereotypen präsentiert[19]. Diese Stereotypliste wird bereits erstellte Startprofile beinhalten (z. B. Action-Fan, Sci-Fi-Fan etc.). Dadurch wird der Einstieg für den Benutzer durch geringeren Aufwand und für das System durch bereits existierende und berechnete Profile leichter gemacht.

Die Benutzerdaten müssen außerdem so gut wie möglich von Schwankungen (Signalrauschen) bereinigt werden, um das *collaborative filtering*, indem die Tendenzen sich besser erkennen lassen, zu verfeinern. Eine ganz rudimentäre Problematik ist, dass alle Menschen anders bewerten. Manche sind ziemlich positiv eingestellt und geben oft eine gute Bewertung, andere betrachten Filme eher kritisch und bewerten sie dementsprechend negativ (vgl. Jin, Si 2004, S. 568). Als Folge kann sich zum Beispiel ergeben, dass der eine Benutzer seinen Lieblingsfilm mit maximaler Bewertung von 10 jedoch der andere seinen Lieblingsfilm nur mit 8 oder 9 Punkten bewertet. Ferner kann jede Person eine andere Skala für die Bewertungen benutzen, einige haben eine engere Bewertungsskala, andere wählen von der ganzen Skala (vgl. Resnick et al. 1994, S. 180-181). Man steht nun vor einem Skalierungsproblem, das die Daten verfälscht und unvergleichbar macht. Daher ist es

[19] Eine Art „Persönlichkeitstest“ wäre auch denkbar aber kaum vorteilhaft, da die Selbsteinschätzung der Menschen und ihre Fähigkeit ihrer Affektivität zu beschreiben und zu vorhersagen sehr unpräzise ist (vgl. Wilson, Gilbert 2003, S. 369-375).

erforderlich, diese Bewertungen immer zu normieren[20], entweder anhand eines Mittelwertes von allen Benutzerbewertungen, oder anhand eines Mittelwertes nur auf der Basis von dieser einzelnen Person.

7.2 Zeitbezogene Phänomene in *collaborative filtering*

Im April 2010 wurde in CACM ein Beitrag mit dem Titel *Collaborative Filtering with Temporal Dynamics* von Y. Koren publiziert. Sein Ziel ist es, ein Modell zu entwickeln, das langfristige und kurzfristige Tendenzen bei Bewertungen erkennen und unterscheiden kann. Dabei sollen langfristige Tendenzen aufbewahrt und in ein CF-Modell einbezogen werden, kurzfristige (sog. *noise*) dagegen gilt es zu isolieren (vgl. Koren 2010).

Korens Beobachtungen bezogen auf das *Netflix Dataset* werden mit den folgenden Punkten wiedergegeben:

- Die Bewertungen steigen im Durchschnitt plötzlich von 3,4 auf 3,6 Punkten im Jahr 2004.
- Bewertungen tendieren bezogen auf das Alter der Filme zu steigen – je ferner ein Film in der Vergangenheit liegt, desto höhere Bewertungen werden gegeben.
- Bewertungen sind vom Wochentag abhängig (vgl. dazu auch Chapphannarungsri 2009).

Solche zeitbezogene Abweichungen können sehr negative Auswirkungen haben, weil sie die langfristigen Vorhersagen (hier Empfehlungen) verfälschen können. Im Fall von kurzfristigen Abweichungen können die langfristigen Tendenzen negativ beeinflusst werden oder sie werden vom System gar nicht erkannt.

Was für eine Bedeutung haben diese Abweichungen für das in dieser Arbeit vorgestellte Modell? Das hybride Modell wird nicht so stark davon beeinflusst, da es als Basis ein *content-based* Empfehlungssystem hat. Da die Empfehlungsliste durch kollaboratives Filtern nur ergänzt wird, haben diese zeitbezogenen Diskrepanzen nur

[20] Jin und Si untersuchen die Effektivität verschiedener Normalisierungsmethode für Benutzerbewertungen und stellen fest, dass eine Normalisierungsmethode, welche aus der U-Statistik stammt und die Wahrscheinlichkeit für eine positive Bewertung ermittelt, bessere Ergebnisse liefert als gängige Methode, wie gaußsche Normalisierung (vgl. Jin, Si 2004).

Einfluss auf einem relativ kleinen Teil der Empfehlungen. Dennoch kann eine Berücksichtigung von solchen Deviationen nur vorteilhaft sein.

Es gibt sehr viele Möglichkeiten, den kollaborativen Teil des Systems zu verfeinern, wie die Lösung des Netflix-Wettbewerbs, mit einem Konglomerat von mehr als 100 Algorithmen und Modellen zur Verbesserung der kollaborativen Empfehlung von Filmen, zeigt (vgl. Bell et al. 2008, S.1). Diese müssen dennoch an den jeweiligen Anforderungen mit angemessenem Aufwand angepasst werden.

Obwohl durch die Implementierung der Filmähnlichkeit das Datenmodell und das Ähnlichkeitsmaß als adäquate Lösung validiert wurden, da Filmprofile an sich mit den Benutzerprofilen bezüglich der Vektordarstellung identisch sind, wird die weitere Umsetzung des vorgeschlagenen Systems noch wichtige Erkenntnisse liefern können. Dies wird eine Evaluation der Information-Retrieval-Qualitäten des Systems, wie etwa durch *precision* und *recall,* ermöglichen, welche die Performanz des Hybridsystems im Vergleich zur Performanz der einzelnen Komponenten (*content-based* und *collaborative filtering*) bewertbar machen wird. Auch eine benutzerbezogene Evaluation, zum Beispiel in der Form eines Akzeptanztests, wäre wünschenswert.

Ferner muss die technische Problematik bei größeren Datenmengen untersucht werden, obwohl dieser Punkt für kleine und mittelgroße Domäne eher nachrangig ist. Da die Ähnlichkeitsvergleiche eine quadratische Komplexität mit sich bringen, muss noch eine Lösung für eine große Menge von aktiven Nutzern (z. B. 50.000 Nutzer, 25*108 Vergleiche, ca. 277-mal mehr Rechenaufwand als bei der oben aufgeführten Filmähnlichkeit) gefunden werden. Obwohl nur ein Bruchteil davon täglich aktualisiert werden muss, muss in diesem Fall auf Clusteringverfahren zurückgegriffen werden, um eine größere Anzahl von Filmen und Nutzern zu stemmen.

8 Zusammenfassung

Im Rahmen dieser Arbeit wurden die theoretischen Grundlagen und Ansätze der Empfehlungsgenerierung in Verbindung mit kleinen und mittelgroßen Filmplattformen behandelt. Durch den Entwurf eines Systemmodells für Empfehlungsgenerierung von Filmen am Beispiel der Filmplattform *critic.de* wurde ein praktischer Bezug zu der daraus entstehenden Problematik hergestellt. Die anschließende Implementierung des Filmähnlichkeitssystems validierte das vorgeschlagene Datenmodell zur Repräsentation von Filmen und den Ähnlichkeitsalgorithmus. Dies ermöglichte ferner einen Einblick in die technischen Aspekte bei der Entwicklung von Empfehlungssystemen.

Es wurde gezeigt, dass die Kaltstartproblematik, welche besonders Systeme mit wenigen Benutzern betrifft, nur anhand einer Hybridisierung bewältigt werden kann. Wie Burke in seiner Evaluation von hybriden Empfehlungssystemen feststellt, kompensiert diese genau den Mangel von Daten (v.a. Bewertungen):

“It is quite clear, as others have also shown, that there is an unqualified benefit to hybrid recommendation, particularly in the case of sparse data.” (Burke 2007, S. 403)

Ein meta-hybrides System bestehend aus *content-based und collaborative filtering* wurde als Lösung für die Kaltstartproblematik vorgeschlagen. Dank seines zweistufigen Aufbaus kann diese auch bei einer sehr kleinen Anzahl von Benutzern und Benutzerdaten sinnvolle Empfehlungen generieren. Zusätzlich wurden verschiedene Verbesserungsmöglichkeiten für den kollaborativen Teil des Systems vorgestellt.

Auf dem Gebiet der Empfehlungssysteme haben die kollaborativen Filterungsalgorithmen meistens bessere Ergebnisse geliefert, da inhaltsbasierende Systeme nicht alle Dimensionen der Benutzerpräferenzen bei komplexen Gegenständen, wie Filme, abdecken können. Dennoch, wie diese Arbeit zeigen konnte, sind sie nicht universell einsetzbar, vor allem bei wenigen Bewertungsdaten. Dieses Problem, zusammen mit der Kaltstartproblematik, erfordert neue fallspezifische Lösungen, welche nur durch Hybridisierung hervorzubringen sind, eine Methode, die dank der Vielfalt an Hybridisierungsmöglichkeiten weiterer Forschung bedarf.

9 Abbildungsverzeichnis

10 Tabellenverzeichnis

11 Literaturverzeichnis

ADOMAVICIUS, Gediminas ; KWON, YoungOk: New Recommendation Techniques for Multicriteria Rating Systems. In: *IEEE Intelligent Systems* 22 (2007), S. 48–55. URL http://dx.doi.org/10.1109/MIS.2007.58

ADOMAVICIUS, Gediminas ; SANKARANARAYANAN, Ramesh ; SEN, Shahana ; TUZHILIN, Alexander: Incorporating contextual information in recommender systems using a multidimensional approach. In: *ACM Trans. Inf. Syst* 23 (2005), S. 103–145. URL http://doi.acm.org/10.1145/1055709.1055714

BALABANOVIĆ, Marko ; SHOHAM, Yoav: Fab: content-based, collaborative recommendation. In: *Communications of the ACM* 40 (1997), S. 66–72. URL http://doi.acm.org/10.1145/245108.245124

BILLSUS, Daniel ; PAZZANI, Michael J.: User Modeling for Adaptive News Access. In: *User Modeling and User-Adapted Interaction* 10 (2000), 2-3, S. 147–180. URL http://portal.acm.org/citation.cfm?id=598285.598352

BURKE, Robin: Hybrid Recommender Systems: Survey and Experiments. In: *User Modeling and User-Adapted Interaction* 12 (2002), S. 331–370. URL http://portal.acm.org/citation.cfm?id=586321.586352

BURKE, Robin: Hybrid web recommender systems. In: BRUSILOVSKY, Peter; KOBSA, Alfred; NEJDL, Wolfgang (Hrsg.): *The Adaptive Web : Methods and strategies of web personalization*. Berlin, Heidelberg : Springer-Verlag, 2007, S. 377–408

BUSCOMBE, Edward: The Idea of Genre in the American Cinema. In: GRANT, B. K. (Hrsg.): *Film Genre Reader III :* University of Texas Press, 2003, S. 13–24

CANDILLIER, Laurent ; MEYER, Frank ; BOULLÉ, Marc: Comparing State-of-the-Art Collaborative Filtering Systems. In: *MLDM '07 Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition* (2007), S. 548–562. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.151.6237

CHAPPHANNARUNGSRI, Keittima ; MANEEROJ, Saranya: Combining Multiple Criteria and Multidimension for Movie Recommender System. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists* (2009), S. 698–703. URL http://www.iaeng.org/publication/IMECS2009/IMECS2009_pp698-703.pdf

CLAYPOOL, Mark ; GOKHALE, Anuja ; MIRANDA, Tim ; MURNIKOV, Pavel ; NETES, Dmitry ; SARTIN, Matthew: Combining Content-Based and Collaborative Filters in an Online Newspaper. In: *Proceedings of ACM SIGIR Workshop on Recommender Systems* (1999). URL http://web.cs.wpi.edu/~claypool/papers/content-collab/content-collab.pdf

DIMITROV, Krasen ; WOLFF, Christian: Ein meta-hybrides recommendation system für die webbasierte Filmplattform critic.de. In: *Workshop Audiovisuelle Medien WAM 2010: Digitale Mediendistribution* (2010), S. 63–71. URL http://monarch.qucosa.de/fileadmin/data/qucosa/documents/6040/data/wam10_monarch.pdf

GRANT, B. K. (Hrsg.): Film Genre Reader III : University of Texas Press, 2003

GREG LINDEN ; BRENT SMITH ; JEREMY YORK: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. In: *IEEE Internet Computing* 7 (2003), S. 76–80

HAENELT, Karin: Ähnlichkeitsmaße für Vektoren : *Kursfolien (1. Fassung 15.11.2000)*. URL http://kontext.fraunhofer.de/haenelt/kurs/folien/Haenelt_VektorAehnlichkeit.pdf. – Aktualisierungsdatum: 2007-10-21 – Überprüfungsdatum 2010-11-24

JANNACH, Dietmar: Recommender systems : An introduction. New York : Cambridge University Press, 2010

JIN, Rong ; SI, Luo: A study of methods for normalizing user ratings in collaborative filtering. In: *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (2004), S. 568–569. URL http://doi.acm.org/10.1145/1008992.1009124

JONES, William P. ; FURNAS, George W.: Pictures of relevance: a geometric analysis of similarity measures. In: *J. Am. Soc. Inf. Sci* 38 (1987), S. 420–442. URL http://portal.acm.org/citation.cfm?id=35053.35056

KONSTAN, J. A. Riedl J. Borchers A. and Herlocker J. L.: Recommender Systems: A GroupLens Perspective. In: *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)* (1998), S. 60–64

KOREN, Yehuda: Collaborative filtering with temporal dynamics. In: *Communications of the ACM* 53 (2010), Nr. 4, S. 89–98. URL http://cacm.acm.org/magazines/2010/4/81486-collaborative-filtering-with-temporal-dynamics/fulltext

KRAUSE, Eugene: Taxicab Geometry : An Adventure in non-Euclidean Geometry. New York : Dover, 1986

LAM, Xuan Nhat ; VU, Thuc ; LE, Trong Duc ; DUONG, Anh Duc: Addressing cold-start problem in recommendation systems. In: *Proceedings of the 2nd international conference on Ubiquitous information management and communication* (2008), S. 208–211

LANG, Ken: Newsweeder: Learning to filter netnews. In: *In Proceedings Of The Twelfth International Conference On Machine Learning* (1995), S. 331–339. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.85.7363

LEE, Victor W. ; KIM, Changkyu ; CHHUGANI, Jatin ; DEISHER, Michael ; KIM, Daehyun ; NGUYEN, Anthony D. ; SATISH, Nadathur ; SMELYANSKIY, Mikhail ; CHENNUPATY, Srinivas ; HAMMARLUND, Per ; SINGHAL, Ronak ; DUBEY, Pradeep: Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU. In: *SIGARCH Comput. Archit. News* 38 (2010), S. 451–460. URL http://doi.acm.org/10.1145/1816038.1816021

M. BELL, Robert ; KOREN, Yehuda ; VOLINSKY, Chris: The BellKor 2008 Solution to the Netflix Prize. URL http://www.netflixprize.com/assets/ProgressPrize2008_BellKor.pdf – Überprüfungsdatum 2010-12-30

MANEEROJ, Saranya ; KATO, Yuka ; HAKOZAKI, Katsuya: An Advanced Movie Recommender System Based on High-Quality Neighbors. In: *IPSJ Digital Courier* 1 (2005), S. 181–192

MASTHOFF, Judith: The Pursuit of Satisfaction: Affective State in Group Recommender Systems, Bd. 3538. In: ARDISSONO, Liliana; BRNA, Paul; MITROVIC, Antonija (Hrsg.): *User Modeling 2005 :* Springer Berlin / Heidelberg, 2005 (Lecture Notes in Computer Science), S. 297–306

MONTANER, Miquel ; LÓPEZ, Beatriz ; LA ROSA, Josep Lluís de: A Taxonomy of Recommender Agents on the Internet. In: *Artificial Intelligence Review* 19 (2003), S. 285–330. URL http://dx.doi.org/10.1023/A:1022850703159

NOREAULT, Terry ; MCGILL, Michael ; KOLL, Matthew B.: A performance evaluation of similarity measures, document term weighting schemes and representations in a Boolean environment. In: *SIGIR '80 Proceedings of the 3rd annual ACM conference on Research and development in information retrieval* (1981), S. 57–76. URL http://portal.acm.org/citation.cfm?id=636669.636674

PAZZANI, Michael J.: A Framework for Collaborative, Content-Based and Demographic Filtering. In: *Artificial Intelligence Review* 13 (1999), S. 393–408. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.5215

RESNICK, Paul ; IACOVOU, Neophytos ; SUCHAK, Mitesh ; BERGSTROM, Peter ; RIEDL, John: GroupLens: an Open Architecture for Collaborative Filtering of Netnews. In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (1994), S. 175–186. URL http://doi.acm.org/10.1145/192844.192905

SANTINI, Simone ; JAIN, Ramesh: Similarity Measures. In: *IEEE Trans. Pattern Anal. Mach. Intell* 21 (1999), S. 871–883. URL http://dx.doi.org/10.1109/34.790428

SCHAFER, J. Ben ; KONSTAN, Joseph ; RIEDL, John: Recommender Systems in E-Commerce. In: *EC '99: Proceedings of the First ACM Conference on Electronic Commerce* (1999), S. 158–166. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.2552

SCHAFER, J. Ben ; KONSTAN, Joseph A. ; RIEDL, John: E-Commerce Recommendation Applications. In: *Data Mining and Knowledge Discovery* 5 (2001), S. 115–153. URL http://portal.acm.org/citation.cfm?id=593429.593510

SCHAFER, J. Ben ; KONSTAN, Joseph A. ; RIEDL, John: Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations. In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM 2002)* (2002), S. 43–51. URL http://doi.acm.org/10.1145/584792.584803

TOWLE, Brendon ; QUINN, Clark: Knowledge Based Recommender Systems Using Explicit User Models. In: *Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04* 10 (2000), S. 74–77. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.7542

VAN METEREN, Robin ; VAN SOMEREN, Maarten: Using Content-Based Filtering for Recommendation. In: *Proceedings of MLnetECML2000 Workshop* (2000), 4203/2006, S. 1–10

WILSON, Timothy D. ; GILBERT, Daniel T.: Affective Forecasting. In: P. ZANNA, Mark (Hrsg.): *Advances in experimental social psychology.* Amsterdam : Academic Press, 2003 (35), S. 345–411

ZHANG, Yi ; CALLAN, Jamie ; MINKA, Thomas: Novelty and Redundancy Detection in Adaptive Filtering. In: *SIGIR '02 Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (2002), S. 81–88. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.7268

Erklärung

Ich habe die Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und bisher keiner anderen Prüfungsbehörde vorgelegt.

Regensburg, den 08. Februar 2011 .....

(Vorname Nachname)