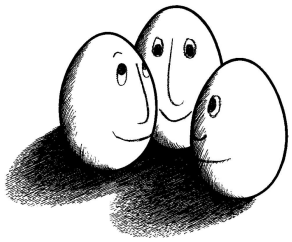
Diplomarbeit

Interessantheit von automatisch generierten Filmempfehlungen

Rüdiger Alberts



Diplomarbeit
am Fachbereich Informatik
der Universität Dortmund

31. August 2005

Betreuer:

Prof. Dr. Katharina Morik
Dipl.-Inform. Michael Wurst

Danksagung

Zuerst möchte ich mich bei Prof. Dr. Katharina Morik und Dipl.-Inform. Michael Wurst für die exzellente Betreuung bedanken. Sie hatten bei Fragen und Problemen stets Zeit für mich und haben mich, als mein Leben außerhalb der Universität im Chaos versank, durch ihre Unterstützung motiviert, mein Studium doch noch zu einem glücklichen Abschluss zu bringen. Mein ganz besonderer Dank gilt all den Freiwilligen, die an den praktischen Versuchen meiner Arbeit teilgenommen und fleißig Filme und Empfehlungen bewertet haben. Ohne ihren Einsatz wäre diese Arbeit nicht möglich gewesen. Insbesondere danke ich Marcus Straßer, Helge Homburg und Antonia Jacobsen für ihre fleißige „Rekrutierarbeit“. Ingo Mierswa möchte ich für seine Unterstützung bei Benutzung der Lernumgebung *YALE* und seine *TeX*-Tipps danken. Keine Frage war ihm zu trivial und kein Problem zu klein, als dass er nicht sofort alles stehen und liegen gelassen und sich um Hilfe bemüht hätte. Außerdem hat er mir als „geborener Lehrer“ auf einfache Weise einen Einblick in das Thema *SVM* verschafft. Andreas Greve danke ich für seine Unterstützung bei diversen Hard- und Software-Problemen, sowie die von ihm angestoßene Diskussion über eine zeitlich optimale Kreuzvalidierung. Ohne seine Anregungen würde die Validierung meines kollaborativen Algorithmus wohl noch heute ergebnislos laufen. Eduard Heinle gilt mein Dank für diverse Anregungen, die mir sehr geholfen haben, als meine Arbeit nicht so richtig „in Schwung“ kommen wollte. Sibylle Schlosser und Axel Dietrich danke ich für die Durchsicht meiner Arbeit in letzter Minute. Last but not least möchte ich meinen besten Freunden Axel Dietrich, Christian Link, Martin Schönleben und Antonia Jacobsen danken, dass sie in schwierigen Situationen stets für mich da waren. Ohne ihre Freundschaft und Unterstützung wäre diese Arbeit nicht zustande gekommen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Information overload, man! As a society we're drowning in a quagmire of vid-clips, e-mail, and sound bytes! We can't absorb it all! There's only one sane solution: BLOW IT UP!

Mad Stan, *Batman Beyond*, 1999

Wir leben im Informationszeitalter. Noch nie war es so einfach, in so kurzer Zeit an eine so große Menge von Informationen[1] zu gelangen wie heutzutage. An jedem Bahnhof kann man unzählige verschiedene Tageszeitungen und Magazine in mehreren Sprachen erhalten. Die meisten Haushalte besitzen einen Kabelanschluss oder eine Satellitenantenne[2] und können somit eine steigende Zahl von Fernsehsendern und Radio-Programmen, ebenfalls in mehreren Sprachen, empfangen. Auch die wieder ansteigende Zahl von Neuerscheinungen bei Büchern trägt dieser Entwicklung Rechnung.[3]
Insbesondere aber die rasante Entwicklung des Internets in den letzten 10 Jahren,[4] hat dazu beigetragen, dass heutzutage mit minimalem Aufwand eine gewünschte Information abgerufen werden kann. Zeitraubende Gänge zu Büchereien und Bibliotheken gehören dank einer steigenden Zahl von Online-Zeitschriften-, Bibliotheken und Buchhandlungen der Vergangenheit an und auch die durch Öffnungszeiten gegebenen Einschränkungen entfallen bei der Nutzung des Mediums Internet.
Schon im Jahr 2004 besaßen mehr als die Hälfte der deutschen Haushalte einen Internetzugang [5] und diese Zahl steigt stetig an. Kompakte und leistungsfähige Notebooks, Laptops und Hand-Held-Computer sowie Mobil-Telefone erlauben schnellen Datenaustausch und Zugriff auf das Internet auch von unterwegs aus.
Wie so oft im Leben hat diese Entwicklung aber auch ihre Nachteile. Die angesprochene Informationsexplosion ging leider nicht mit einem Sprung in der menschlichen Evolution einher,[6] der es dem einzelnen Menschen ermöglichen würde, diese Menge an Information zu erfassen. Vielmehr wird es durch diese wachsende Menge an zur Verfügung stehendem Wissen immer schwieriger, die Information zu extrahieren, die man zu einem bestimmten Zeitpunkt konkret benötigt. Da Menschen physiologisch nur in der Lage sind, einen Bruchteil des heutzutage täglich auf sie einwirkenden Informationsflusses bewusst zu verarbeiten, stellt sich oft das Gefühl von Überlastung und Desorientierung ein. Dies gilt nicht nur im beruflichen Teil des Lebens, wo verschiedenste Weiterbildungsmöglichkeiten angeboten werden, die ein Schritthalten mit dem in allen Berufsbereichen immer schneller vonstatten gehenden technischen Fortschritt ermöglichen sollen und wo Politiker immer häufiger ein „lebenslanges Lernen“ fordern. Auch in der eigentlich zur Entspannung dienenden Freizeit werden wir von diesem “Bombardement” an Informationen nicht verschont. Es gibt unzählige Produkte, die in unserer Freizeit konsumiert und Aktivitäten, die ausprobiert werden wollen. Firmen müssen, um aus dieser großen Masse von zur Auswahl stehenden Produkten und Dienstleistungen herauszustechen, immer aggressivere Werbestrategien betreiben und nutzen dabei auch in zunehmendem Maße moderne Technologien wie z.B. das Internet oder Mobiltelefone, wie eine steigende Zahl von unerwünschten „Spam“-Mails, Werbebannern und Werbe-SMS zeigt. Da jede Firma natürlich von

[1] Eigentlich gibt es keinen Plural des Begriffs „Information“. Da sich der Plural „Informationen“ jedoch sprachlich so eingebürgert hat, wird er hier der Einfachheit halber ebenfalls verwendet.
[2] 2003 besaßen laut [DESTATIS 2004] 36% der Haushalte einen Satellitenanschluss und 53% einen Kabelanschluss. Geht man davon aus, dass Haushalte stets nur eine der beiden Empfangsmöglichkeiten nutzen, so kommt man auf insgesamt 89% Abdeckung.
[3] Trotz konjunkturell schlechter Lage stieg 2004 die Zahl der Neuerscheinungen bei Büchern auf der Frankfurter Buchmesse gegenüber dem Vorjahr um 41% auf 104566, vgl. dazu auch [UHLENDORF 2004].
[4] Die Anzahl der Hosts im Internet stieg nach Angaben einer von der *ISC* bei den [NETWORK-WIZARDS 2005] in Auftrag gegebenen Studie von 4852000 im Januar 1995 auf 317646084 Hosts im Januar 2005 und hat sich somit um den Faktor 65 erhöht.
[5] Vgl. [DESTATIS 2005].
[6] Vgl. [TERVEEN und HILL 2001].

sich selbst behauptet, das jeweils beste Produkt bezogen auf einen bestimmten Bereich anzubieten, fällt es dem Konsumenten immer schwerer, eine Kaufentscheidung zu treffen, um dann z.B. bequem von zu Hause aus in einem der vielen Online-Shops das Produkt seiner Wahl bestellen zu können - die Wahl wird hier zur Qual.
In dieser Situation unübersichtlicher Wahlmöglichkeiten nutzen Menschen als Ausweg das altbewährte Mittel von Empfehlungen: Freunde, Bekannte, aber auch Fachzeitschriften und andere Quellen, von denen man weiß, dass sie einen ähnlichen Geschmack besitzen wie man selbst, werden als Wegweiser zu Produkten genutzt, die für einen selbst möglicherweise auch nützlich sind. Dadurch spart man sich den enormen Zeitaufwand, sich erst in ein Gebiet einzuarbeiten und so viele Quellen wie möglich zu sichten, um allein zu den gewünschten Produkten zu gelangen. Welchen Einfluss Empfehlungen dabei haben, zeigt sich auch daran, dass immer mehr Firmen wie Banken oder Zeitschriftenverlage auf Empfehlungen setzen, um neue Kunden zu gewinnen. In den letzten Jahren hat sich sogar ein eigener Wirtschaftszweig entwickelt, der auf diesem „Empfehlungsgeschäft“ beruht, das so genannte „Network-“ oder „Multi-Level-Marketing“. Dieses beruht auf der Theorie, dass es für Firmen letzten Endes billiger ist, Kunden auch für indirekte Empfehlungen Provisionen auszuzahlen, statt selbst das Geld in teure Werbung zu stecken, deren Kosten einen immer größeren Anteil am Endpreis des Produkts ausmachen und die von den durch Werbung bereits überfluteten und genervten Kunden oft sowieso ignoriert wird. Freunden und Bekannten hingegen, die Produkte empfehlen, vertraut man eher als anonymer Werbung. So kann das gesparte Geld in Forschung oder Infrastruktur der Firma investiert werden. Wie wirksam das Konzept von Empfehlungen dabei ist, zeigen die enormen Wachstumsraten in diesem Bereich. So werden schon jetzt weltweit ca. 100 Mrd. US Dollar durch Network-Marketing umgesetzt, mit steigender Tendenz.[7] In Deutschland noch häufig als illegales „Schneeballprinzip“ angesehen, gibt es immer mehr seriöse Auseinandersetzungen mit diesem Themenbereich[8] und Befürworter aus Wirtschaft und Politik, wie z.B. die Wirtschaftskammer Oberösterreich, das Institut für Mittelstandsforschung in Bonn[9], Ex-US-Präsident Bill Clinton oder den Bundeskanzler von Österreich, Wolfgang Schüssel.
Dieses Potenzial von Empfehlungen, egal ob im wirtschaftlichen oder privaten Bereich, hat jedoch einen großen Nachteil - um das Netz aus Freunden und Bekannten, welches die Quelle für Empfehlungen ist, zu bilden bzw. ein vorhandenes Netz zu pflegen, ist eine nicht unerhebliche Investition von Zeit notwendig. Und gerade mangelnde Zeit ist es ja, die uns überhaupt auf Empfehlungen zurückgreifen lässt. Da ist es von Vorteil, dass Methoden zur Automatisierung des Empfehlungsprozesses bereits existieren, die sich aus einfachen Methoden zur Informationsfilterung entwickelt haben. Suchmaschinen im Internet liefern nach der Eingabe von Schlüsselwörtern nur die im World Wide Web vorhandenen Webseiten zurück, die die angegebenen Schlüsselwörter enthalten und helfen so in einem ersten Schritt, Informationen zu filtern. Verfahren aus dem Bereich des *Information Retrieval* können dann als Weiterentwicklung dieser Filterung genutzt werden, um nach einem entsprechenden Feedback des Benutzers eine Liste von Webseiten als Ergebnis der Suche nach bspw. probabilistischen Gesichtspunkten bzgl. des Nutzens für den Suchenden neu zu sortieren.
Der nächste Schritt in der Entwicklung sind Webbrowser, Mail- und Newsgroup-Programme, die mittels einfacher, meist inhaltsorientierter Filterverfahren arbeiten.[10] Nach einer „Lernphase“, in der anhand des Verhaltens eines Nutzers gewünschte und abgelehnte Objekte identifiziert werden, können später ähnliche Objekte aus der Gesamtmenge an Daten herausgefiltert werden.
Automatisierte *Empfehlungssysteme* schließlich, liefern einem Benutzer eine Liste von Objekten wie z.B. CDs oder Filme, die - basierend auf dem bisherigen Verhalten des Nutzers - höchstwahrscheinlich dessen Interessen entsprechen. Dabei kann es sich bei dem Verhalten des Nutzers um das Durchführen sog. expliziter Bewertungen für einige ihm bekannte Objekte handeln oder um Aktionen die nahe legen, dass der Nutzer bestimmte Objekte mag bzw. nicht mag, was als implizite Bewertung bezeichnet wird. Eine

[7] Siehe [IFM 2002].
[8] Prof. Dr. Michael M. [ZACHARIAS 2001] hat 2001 im Auftrag der Wirtschaftskammer Oberösterreich eine Untersuchung des Network-Marketings durchgeführt und arbeitet nun an einer entsprechenden Studie für Deutschland.
[9] Siehe [IFM 2002].
[10] Auch diese Verfahren sind schon längere Zeit bekannt und mündeten schließlich in inhaltbasierten Empfehlungssystemen, die auch als eigenschaftsbasierte Empfehlungen bezeichnet werden. Nähere Informationen zu diesen Systemen finden sich in Abschnitt 2.3.

positive implizite Bewertung könnte sich bspw. daraus ergeben, dass ein Nutzer ein Objekt, wie z.B. eine DVD kauft, während eine negative implizite Bewertung aus dem Verhalten geschlossen werden könnte, dass der Nutzer ein anderes Objekt, wie bspw. eine Nachricht aus einer Newsgroup sofort nach Erhalt in den Papierkorb verschiebt. Neben dieser Wahl, implizite oder explizite Bewertungen zu verwenden, stehen einem Entwickler heutzutage bei der Implementierung eines Empfehlungssystems viele Möglichkeiten offen. Diese Möglichkeiten betreffen sowohl die Art, wie die Bewertungen eines Nutzers zu einem Modell seiner Präferenzen verarbeitet werden, als auch den Prozess, mittels dem aus dem Präferenzmodell personalisierte Empfehlungen entstehen. Bei all ihren Unterschieden verfolgen Empfehlungssysteme im allgemeinen jedoch gemeinsame Ziele:

- Die bereits erwähnte *Zeitersparnis:* Statt Menschen zu befragen, Zeitschriften zu lesen und auf ähnlich zeitaufwändige Methoden zurückzugreifen um potenziell interessante Objekte zu identifizieren, reicht es bei Empfehlungssystemen meist, einmalig einige bekannte Objekte zu bewerten.
- *Hohe Datenverarbeitungskapazität und Aktualität:* Da Menschen nur eine begrenzte Menge von Daten verarbeiten können, basieren Empfehlungen von Menschen für andere Menschen auch nur auf einer kleinen Teilmenge der möglichen Objekte des jeweiligen Bereichs. Automatisierte Empfehlungssysteme hingegen können wesentlich mehr Daten verarbeiten und haben bzgl. dieser zugrunde liegenden Daten auch eine extrem hohe Wachstumsrate, insbesondere in Zeiten globaler Vernetzung über das Internet. Daher soll dieser „Wissensvorsprung“ genutzt werden - um wesentlich umfangreichere, genauere und aktuellere Empfehlungen zu geben als dies einem Menschen möglich wäre.
- *Objektive Informationsfilterung:* Im normalen Leben wird die unübersichtliche Menge von vorhandenen Objekten bereits durch verschiedene Personen gefiltert. So bestimmen Redakteure, welche Themen in einer Zeitschrift, im Fernsehen oder Radio präsentiert werden, während Kinobesitzer entsprechend entscheiden, welche Filme sie zeigen. Oft basiert diese „subjektive Informationsfilterung“ auf kommerziellen Faktoren. Auch Empfehlungssysteme nehmen eine Filterung der vorhandenen Daten vor, dieser Prozess basiert jedoch auf dem jeweiligen Profil eines Nutzers und ist somit objektiv nachvollziehbar, was Nutzer von den erwähnten subjektiven Empfehlungen unabhängig macht.
- *Erhöhte Kundenbindung- und zufriedenheit für Firmen:* Firmen, die automatisierte Empfehlungssysteme einsetzen, tun dies mit dem Ziel, eine Kunden an sich zu binden und deren Zufriedenheit mit der Firma zu steigern. Statt den Kunden mit allgemeiner Werbung anzusprechen, werden ihm nur solche Produkte angeboten, die ihn wirklich interessieren. Kauft der Kunde ein empfohlenes Produkt, so ist die Wahrscheinlichkeit groß, dass er mit dem Kauf und damit mit der Firma zufrieden sein wird.

Neben diesen erreichbaren Zielen bringen automatisierte Empfehlungssysteme jedoch auch einige Gefahren mit sich:

- *Abhängigkeit vom Empfehlungssystem:* Kein Computerprogramm ist perfekt und fehlerfrei. Begibt man sich zu sehr in Abhängigkeit von Empfehlungssystemen, können möglicherweise interessante Objekte übersehen werden.
- *Ersetzen des sozialen Kontakts:* Das Einholen von Empfehlungen von anderen Menschen hat auch eine positive soziale Komponente. Diese können Empfehlungssysteme nicht ersetzen.
- *Ausschluss von Nischenobjekten:* Viele Empfehlungssysteme arbeiten nach statistischen Prinzipien. Dadurch besteht die Gefahr, dass wenig bewertete „Nischenobjekte“ nicht empfohlen werden.
- *Demotivation der Nutzer durch hohen Anfangsaufwand:* Einige Empfehlungssysteme brauchen eine Anlaufphase, um akkurate Empfehlungen generieren zu können. Ist ein Anwender von den ersten Empfehlungen eines Systems enttäuscht, so kann es dazu kommen, dass er das System fortan meidet und ihm wertvolle Empfehlungen entgehen.
- *Datenmissbrauch:* Empfehlungssysteme verwalten mit den Präferenzmodellen von Anwendern sensible Daten, die natürlich auch missbraucht werden können.

- *Manipulation von Bewertungen:* Bewertungen können aus kommerziellem Interesse heraus manipuliert werden, um den Verkauf eigener Produkte zu fördern.[11]

Um die angesprochenen Ziele möglichst vollständig zu erreichen und die Gefahren zu minimieren, werden im Bereich der Empfehlungssysteme umfangreiche Forschungen betrieben, bei denen neue Verfahren entwickelt oder bestehende Verfahren verbessert werden. Bei der Bewertung des Erfolgs der Neu- bzw. Weiterentwicklungen hat sich jedoch eine problematische Routine entwickelt, die dazu führt, dass das Hauptanliegen von Empfehlungssystemen, mit dem man alle genannten Ziele zusammenfassen kann, aus den Augen verloren wird, nämlich den Menschen in der praktischen Anwendung von Nutzen zu sein. Eine Untersuchung dieses Problems und mögliche Lösungsansätze haben das Entstehen dieser Arbeit motiviert.

1.1 Motivation

Automatisierte Empfehlungssysteme erfreuen sich seit ca. 15 Jahren großer Aufmerksamkeit und Beliebtheit, denn sie bieten Menschen Unterstützung bei dem Problem, aus einer unübersichtlichen Menge von Objekten desselben Typs, wie z.B. Filmen, CDs oder Restaurants, jene herauszufiltern, die für sie potenziell interessant sind.
Durch soziologischen und technologischen Fortschritt in Form von Globalisierung, Vernetzung, erleichtertem Zugriff auf Waren aller Art und effektivere sowie intensivere Methoden der Produktwerbung wird dieses Problem und damit das Interesse an Empfehlungssystemen zusätzlich intensiviert, was auch am steigenden kommerziellen Einsatz solcher Systeme erkennbar ist. Insbesondere auf sog. kollaborativen Verfahren basierende Empfehlungssysteme haben sowohl im wissenschaftlichen, im privaten, als auch im kommerziellen Bereich einen regelrechten Siegeszug angetreten, da sie ein typisch menschliches Verhalten zur Lösung des oben angesprochenen Problems operationalisiert haben. Sie identifizieren Menschen mit ähnlichem Geschmack bzgl. eines bestimmten Objekttyps und holen nachfolgend Empfehlungen von diesen Personen ein.
Diese automatisierten Empfehlungen sind jedoch nicht immer korrekt, so wie auch die Empfehlung einer menschlichen Person mit vermeintlich ähnlichem Geschmack falsch sein kann (schließlich gleicht kein Mensch dem anderen und somit ist die Wahrscheinlichkeit, dass zwei Personen mit absolut identischem Geschmack existieren, eher gering). Trotzdem stellen kollaborative Empfehlungssysteme gegenüber konventionellen Verfahren einen Vorteil dar, da sie ein großes Manko von bis dato aus anderen Bereichen bekannten und auf den Bereich von Empfehlungen übertragbaren Verfahren beheben. Aus dem Bereich des *Information Retrieval* lange bekannte inhalts- bzw. eigenschaftsorientierte Methoden sind bei der Generierung von Empfehlungen auf eine kleine Teilmenge von Objekten beschränkt, die durch die betrachteten Eigenschaften bestimmt wird, während kollaborative Verfahren solchen Beschränkungen nicht unterliegen. Daher ist der vermehrte Einsatz kollaborativer Verfahren im wesentlichen nachvollziehbar und hat sich äußerst positiv auf die Entwicklung besserer Empfehlungsverfahren ausgewirkt. Bei der Beschäftigung mit dem Thema „Empfehlungssysteme“ fällt jedoch bei vielen wissenschaftlichen Texten eine Fokussierung auf kollaborative Verfahren auf, was sich in dem immer noch häufig als Synonym für Empfehlungssysteme gebrauchten Begriff des „kollaborativen Filterns“ zeigt.[12]
Trotz allem persönlichen Interesses des Autors an dem Prinzip kollaborativer Empfehlungssysteme soll hier gezeigt werden, dass dieses Prinzip auf dem Weg zum „perfekten Empfehlungssystem“[13] nicht den letzten Schritt darstellt. Der Vorteil kollaborativer Verfahren auch außergewöhnliche Empfehlungen zu generieren, wird nämlich durch die Gefahr erkauft, dass diese außergewöhnlichen Empfehlungen auch viel Rauschen, d.h. ungültige Empfehlungen enthalten. Somit sollten positive Eigenschaften des kollaborativen Filterns wie das Generieren ungewöhnlicher Empfehlungen beibehalten und durch Hinzunahme weiterer, von anderen Typen von Empfehlungsverfahren eingesetzter Techniken die Nachteile minimiert werden, was zu Hybridsystemen führt.

[11] Siehe [RESNICK und VARIAN 1997].
[12] Vgl. auch Abschnitt 2.2.
[13] Falls die Implementierung eines perfekten Systems überhaupt möglich ist.

Ein weiteres Phänomen welches im Zusammenhang mit wissenschaftlichen Abhandlungen über Empfehlungssysteme auffällt ist eine weitgehende Beschränkung auf „hypothetische Bewertungsmaße“, wenn es um die Optimierung der generierten Empfehlungen geht. „Hypothetisch“ bedeutet hier, dass bereits vorhandene Bewertungsdaten der Form „Nutzer A hat Objekt i mit der Bewertung x versehen“ herangezogen werden, um die Vorhersagegenauigkeit eines Empfehlungsverfahrens zu bestimmen. Dabei wird die von einem Empfehlungssystem vorhergesagte Bewertung für ein Objekt i mit der tatsächlich abgegebenen Bewertung verglichen. Abweichungen können dann durch Maße wie den mittleren absoluten Fehler[14] oder quadratischen Fehler[15] ermittelt werden. Diese Bewertungsmaße bleiben jedoch insofern hypothetisch, dass sie nur die Qualität eines Empfehlungssystems auf bereits vorhandenen Daten berechnen und nichts über die letztendliche Zufriedenheit eines Nutzers mit einem Empfehlungssystem aussagen. Die konkrete Nützlichkeit von automatisch generierten Empfehlungen für einen bestimmten Nutzer kann nur dieser selbst bewerten. Waren die Untersuchungen in den Anfängen von Empfehlungssystemen, insbesondere bei Systemen kollaborativer Art, neben genauen hypothetischen Qualitätsbewertungen vor allem auch durch Untersuchungen des praktischen Einsatzes geprägt, so wird heutzutage eine Evaluierung neu entwickelter Verfahren meist nur unter „Laborbedingungen“ vorgenommen, indem die erwähnten hypothetischen Bewertungsmaße verwendet werden. Dies liegt im Falle kollaborativer Verfahren sicherlich auch daran, dass zu Anfang noch keine Bewertungsdaten in zuvor genannter Form vorlagen. Da kollaborative Empfehlungssysteme von ihrer Funktionsweise her aber gerade auf eine gewisse Mindestanzahl genau solcher Bewertungsdaten angewiesen sind, um überhaupt Bewertungsvorhersagen generieren zu können, waren praktische Einsätze damals unumgänglich. Mittlerweile stehen jedoch mehrere umfangreiche Datensammlungen für die unterschiedlichsten Objekttypen wie Filme, Musikstücke oder Witze zur Verfügung, was praktische Versuche mit realen Nutzern bei einer reinen Berechnung der Vorhersagegenauigkeit eines Empfehlungsverfahrens überflüssig macht. Des weiteren gilt für alle Arten von Empfehlungssystemen, dass eine praktische Untersuchung häufig einen extremen zusätzlichen Zeit- und Arbeitseinsatz erfordert, wie der Autor im Verlauf der Arbeit selbst erfahren musste. Es müssen Versuchspersonen gewonnen werden, ihnen muss für die Bewertung ein ansprechendes und einfach zu bedienendes Interface zur Verfügung gestellt werden und im Fall mehrerer Arbeitsschritte wie in dieser Arbeit müssen die für den Versuch gewonnenen Personen ständig neu motiviert werden.
Trotz dieses erhöhten Aufwands sind jedoch nach Ansicht des Autors praktische Untersuchungen unumgänglich, da die Anwendung im realen Leben letztendlich die Motivation war, automatisierte Empfehlungssysteme überhaupt zu entwickeln. Rein hypothetische Bewertungsmaße haben für die praktische Anwendung nicht immer die Bedeutung, die ihnen oft zugemessen wird. Nur wenige Forscher weisen auf dieses Problem hin, wie z.B. [HERLOCKER et al. 2004]:

> *“[...] algorithmic improvements in collaborative filtering systems may come from different directions than just continued improvements in mean absolute error.”*

Für die praktische Evaluierung von Empfehlungssystemen sowie die Verwendung alternativer Bewertungsmaße sprechen außerdem noch weitere Argumente. Nach Untersuchungen von [HERLOCKER et al. 2004] unterschreiten alle neueren Verfahren im Bezug auf das am häufigsten in diesem Bereich benutzte Bewertungsmaß, den *MAE* einen bestimmten Wert nicht, den [HERLOCKER et al. 2004] auch als “magische Grenze” bezeichnen, so dass bzgl. hypothetischer Bewertungsmaße die Entwicklung verbesserter Empfehlungsverfahren einen Stillstand erreicht zu haben scheint. Außerdem lassen weitere Untersuchungen von [HERLOCKER et al. 2004] darauf schließen, dass Anwendern andere Faktoren wichtiger sein könnten, als nur die reine Genauigkeit eines Empfehlungsverfahrens. Das schließt auch die Tatsache ein, dass bei den angesprochenen „Laborversuchen“ häufig noch um Verbesserungen der Vorhersagegenauigkeit im unteren einstelligen Prozentbereich gerungen wird. Doch es ist anzuzweifeln, dass ein Nutzer bei praktischer Anwendung eines Empfehlungsverfahrens eine Verbesserung des *MAE* um bspw. 0.01 überhaupt subjektiv wahrnehmen kann.

Um diesen Argumenten Rechnung zu tragen, soll in dieser Arbeit ein kollaboratives Verfahren als Vertreter

[14] Engl.: „Mean Absolute Error“ (MAE).
[15] Engl.: „Mean Squared Error“ (MSE).

von „state-of-the-art“-Empfehlungssystemen implementiert und sowohl mit Hilfe hypothetischer Maße als auch durch Nutzer in einer realen Anwendung bewertet werden, um die Frage zu klären, ob ein gutes Ergebnis in Laborversuchen zwangsläufig auch zu einer guten Bewertung durch Anwender im wirklichen Einsatz führt. Dem Einwand von [HERLOCKER et al. 2004] in Form des zuvor benutzten Zitats folgend, wird daher ein Maß eingeführt, das eine Bewertung des praktischen Nutzens von Empfehlungen ermöglicht, die durch automatisierte Empfehlungssysteme erstellt wurden. Die „Interessantheit“ von Empfehlungen, wie sie in Abschnitt 3.3 definiert wird, soll die Zufriedenheit von Nutzern bei der Anwendung eines Empfehlungsverfahrens dokumentieren. Außerdem soll anhand der Interessantheitsbewertungen der Nutzer, wie sie sich aus den Versuchen dieser Arbeit ergeben, der Begriff der Interessantheit aus praktischer Sicht untersucht werden. Die Ergebnisse aus diesen Untersuchungen sollen dann dazu verwendet werden, ein bzgl. der Interessantheit der generierten Empfehlungen verbessertes Verfahren zu entwickeln.
Neben der praktischen Untersuchung soll diese Arbeit jedoch auch einen theoretischen Beitrag zum Forschungsbereich automatisierter Empfehlungssysteme leisten. Bei der Einarbeitung in den Themenbereich gab es insofern Schwierigkeiten, dass ein einheitlicher theoretischer Rahmen fehlte. Die bereits erwähnte Fokussierung auf kollaborative Filterverfahren blendet andere Möglichkeiten der Empfehlungsgenerierung weitgehend aus. Dies führt zu einer Aufteilung des Forschungsbereichs in mehrere Gruppen, die sich hauptsächlich mit kollaborativen, eigenschaftsbasierten, regelorientierten oder anderen Verfahren beschäftigen[16] und selten übergreifend arbeiten. Wissenschaftliche Veröffentlichungen, die versuchen einen allgemeinen Überblick zu allen existierenden Verfahren zu geben, findet man hingegen kaum.
Deshalb soll in dieser Arbeit ein kurzer allgemeiner Überblick des aktuellen Standes im Forschungsbereich automatisierter Empfehlungssysteme gegeben werden, der bereits existierende Kategorisierungen sowie andere Informationen über Methoden in diesem Bereich in einem einheitlichen Rahmen zusammenfasst, um einen leichteren Einstieg in das Themengebiet zu gewährleisten.

[16] Diese Verfahren werden alle in Abschnitt 2.3 erklärt.

Zusammenfassend können die Ziele dieser Arbeit dann wie folgt beschrieben werden:

1. *Überblick*. Erstellung eines Überblicks und einer einheitlichen Klassifikation für Verfahren zur automatischen Generierung von Empfehlungen (Kapitel 2).

2. *Praxisbezogenere Qualitätsmaße sollen hypothetische Maße ergänzen*. Untersuchung der These, dass die letztendliche Nützlichkeit (bezogen auf das Ergebnis des *Data-Mining*-Prozesses[17] aus Sicht des Nutzers) von automatisierten Empfehlungssystemen nicht nur durch übliche hypothetische Bewertungsmaße ermittelt werden kann, sondern neue, auf die praktische Anwendung bezogene Maße gefunden und ergänzend eingesetzt werden müssen.
   Diskussion dieser These aus zwei verschiedenen Blickwinkeln:

   a) *Qualitativ:* Darstellung des Problems hypothetischer Bewertungsmaße wie des mittleren absoluten Fehlers aus Sicht der Literatur und Ansätze für alternative praxisbezogenere Maße.
   Eigene Überlegungen zu diesem Thema und Einführung des konkreten Maßes der „Interessantheit“ von Empfehlungen (Kapitel 3).

   b) *Empirisch:* Implementierung eines kollaborativen „state-of-the-art“ Empfehlungsverfahrens. Verwendung dieses Verfahrens und des zuvor definierten Interessantheitsmaßes in einer praktischen Anwendungsumgebung. Empirischer Nachweis der These, dass eine hohe hypothetische Gegenauigkeit nicht automatisch auch eine große Nützlichkeit aus Sicht von Nutzern (repräsentiert durch das Maß der Interessantheit) nach sich zieht (Kapitel 4 und 5).

3. *Hybridsysteme verbessern die Interessantheit*. Interessante Empfehlungen (Abschnitt 3.3) zeichnen sich durch wenig inhaltlichen Bezug zu den Filmen aus, die ein Nutzer im allgemeinen gut bewertet. Kollaborative Empfehlungssysteme mit ihrer Fähigkeit zur Generierung außergewöhnlicher Empfehlungen erzeugen damit gerade solch interessante Empfehlungen. Leider erzeugen sie gleichzeitig aber auch viel Rauschen, d.h. ungültige Empfehlungen. Eigenschaftsbasierte Systeme hingegen empfehlen solche Objekte, die eine große inhaltliche Ähnlichkeit zu den Objekten haben, die der Anwender gut bewertet hat. Bei vernünftiger Wahl der betrachteten inhaltlichen Eigenschaften erzeugen diese Verfahren kaum Rauschen, aber auch keine außergewöhnlichen Empfehlungen. Kollaborative und eigenschaftsbasierte Verfahren allein haben also nicht das Potenzial, interessante Empfehlungen ohne Rauschen zu erzeugen. Die Kombination aus beiden Verfahren jedoch erscheint vielversprechend. Deshalb wird in Kapitel 6 die These untersucht, dass durch ein entsprechendes Hybridsystem die Interessantheit der durch ein Empfehlungsverfahren generierten Empfehlungen verbessert werden kann.

1.2 Überblick

Nachdem die Gründe für das Thema dieser Arbeit vorgestellt wurden, soll hier kurz der weitere Verlauf skizziert werden:

1. *Überblick und Kategorisierung*
   In Kapitel 2 wird zunächst eine genaue Definition des Begriffs „Empfehlungssysteme“ erarbeitet. Um die besonderen Gegebenheiten im Forschungsbereich der Empfehlungssysteme und die daraus resultierenden Probleme besser zu verstehen, die letzten Endes diese Arbeit motiviert haben, wird

[17] *Data Mining*: Siehe z.B. [Witten und Frank 1999] oder [Mitchell 1997], S. 17.

außerdem ein kurzer geschichtlicher Abriss der Entwicklung automatisierter Empfehlungssysteme gegeben. Eine zusammenfassende Klassifikation von Empfehlungssystemen hilft dabei, sich einen Überblick der existierenden Möglichkeiten zu verschaffen und Empfehlungssysteme gegenüber anderen Methoden abzugrenzen. Häufig benutzte Empfehlungsverfahren werden anhand des Klassifikationssystems ebenso eingeordnet, wie die konkret in dieser Arbeit benutzten Empfehlungssysteme gegenüber anderen Verfahren abgegrenzt werden.

2. *Hypothetische Maße sind zur Qualitätsbewertung von Empfehlungssystemen nicht ausreichend - qualitative Diskussion*
   Entsprechend dem Thema dieser Arbeit spielt die Qualitätsbewertung automatisierter Empfehlungssysteme eine herausragende Rolle und wird in Kapitel 3 ausführlich behandelt. Dabei wird die Frage geklärt, welche Anforderungen bzgl. der Qualität ganz allgemein an Empfehlungsverfahren gestellt werden. Die ganz spezifischen Vor- und Nachteile kollaborativer und eigenschaftsbasierter Verfahren, wie sie in den praktischen Versuchen dieser Arbeit Verwendung finden, werden aufgelistet, da diese Einfluss auf die Qualität von Bewertungen haben. Weiterhin wird untersucht, ob verschiedene Empfehlungsverfahren einfach miteinander verglichen werden können und welche Aussagekraft solche Vergleiche haben. Die Faktoren, die bei solch einem Vergleich beachtet werden müssen werden ebenfalls genannt. Es wird diskutiert, ob rein hypothetische Bewertungsmaße ausreichend für eine Evaluierung sind oder auch alternative praxisbezogene Maße verwendet werden sollen. Auf dieser Diskussion und eigenen Überlegungen in Abschnitt 3.3 basierend, wir dann schließlich ein eigenes praxisorientiertes Maß, das Maß der „Interessantheit“ von Empfehlungen definiert.

3. *Hypothetische Maße sind zur Qualitätsbewertung von Empfehlungssystemen nicht ausreichend - empirischer Nachweis*

   - In Kapitel 4 wird die Umgebung beschrieben, in der die Berechnung der hypothetischen Vorhersagegenauigkeit und die praktischen Versuche durchgeführt wurden. Die benutzten Datenbanken und Hilfsprogramme werden ebenso vorgestellt, wie die Typen von verwendeten Empfehlungssystemen und die konkreten Bewertungsmaße, die zur hypothetischen und praktischen Evaluierung dieser Systeme zum Einsatz kamen. Die Verwendung dieser speziellen Komponenten wird dabei stets begründet.
   - Eine genaue Beschreibung des für die Versuche verwendeten kollaborativen Empfehlungsalgorithmus schließt sich in Kapitel 5 an. Bevor die Implementierung des Algorithmus vorgestellt wird, werden die theoretischen Grundlagen dieses speziellen kollaborativen Verfahrens skizziert und eine Begründung für die Auswahl des Algorithmus geliefert. Die bei der Implementierung eingebauten Parameter des Algorithmus sowie die Probleme bei der Implementierung und deren Lösungen werden skizziert. Anschließend werden die Ergebnisse der praktischen Versuche insbesondere unter Betrachtung der Qualität des kollaborativen Algorithmus bezogen auf die Maße Vorhersagegenauigkeit und Interessantheit präsentiert. Weitere Untersuchungen des direkten Zusammenhangs zwischen Vorhersagegenauigkeit und Interessantheit, sowie zusätzlich aus den praktischen Versuchen gewonnene Ergebnisse wie z.B. die mangelnde Divergenz von Empfehlungslisten für verschiedene Nutzer ergänzen das Bild.

4. *Hybridsysteme liefern interessantere Filmempfehlungen*
   In Kapitel 6 wird dem kollaborativen Algorithmus ein alternatives Verfahren zur Generierung von Empfehlungen gegenübergestellt, das aus den praktischen Erfahrungen mit dem kollaborativen Ansatz entstanden und als Verbesserung des Grundansatzes gedacht ist. Aufbauend auf den Empfehlungen des kollaborativen Verfahrens wird eine inhaltsbasierte Filterstufe entwickelt, so dass sich ein Hybridsystem ergibt. Ziel ist es, mit diesem Hybridsystem die Interessantheit der Vorhersagen zu steigern.
   Wie im Kapitel 5 werden die Kriterien, die zur Auswahl dieses speziellen Verfahrens führten ebenso dargestellt, wie die theoretischen Grundlagen und die Implementierung samt benutzter Parameter, auftretender Probleme und Lösungen. Parallel zum Kapitel 5 schließen die Versuchsergebnisse die

Darstellung des Hybridsystems ab. Die Qualität des Hybridsystems bezogen auf die Genauigkeit und Interessantheit wird besonders betrachtet, aber auch hier werden wieder zusätzlich gewonnene Erfahrungen präsentiert.

5. Abgeschlossen wird die Arbeit schließlich in Kapitel 7 mit einer Diskussion der Ergebnisse dieser Arbeit und sich daraus ergebenden Erkenntnissen, die für weiterführende Forschungen genutzt werden können.

Begriffe:

Einige Begriffe werden in der folgenden Arbeit immer wieder benutzt werden, um wiederholte umständliche Beschreibungen zu vermeiden. Diese Begriffe sollen hier definiert werden:

Objekte bezeichnet allgemein Instanzen aus dem Bereich, für den Empfehlungen generiert werden sollen, also z.B. für den Bereich Musik bestimmte CDs. Viele Autoren wissenschaftlicher Publikationen verwenden auch den Begriff „Produkt“. Da es jedoch nicht nur kommerzielle Anwendungen von Empfehlungssystemen gibt, wurde der allgemeinere Begriff „Objekt“ gewählt, es sei denn, es handelt sich wirklich um einen kommerziellen Kontext.

Nutzer(in)/Benutzer(in)/Anwender(in) Personen, die ein automatisiertes Empfehlungssystem benutzen.

Vorhersage Ein Anwender wird bei der Erstbenutzung eines Empfehlungssystems meist eine Menge von ihm bekannten Objekten bewerten müssen, damit das System daraus die Präferenz dieses Nutzers bezogen auf den jeweiligen Bereich ermitteln kann. Darauf basierend können für Objekte, die dem Nutzer unbekannt sind, Abschätzungen der Bewertung berechnet werden, d.h. das System macht eine Vorhersage darüber, wie der Anwender das unbekannte Objekt konkret bewertet hätte, wenn er es bereits kennen würde. Diese Vorhersage kann je nach benutzter Bewertungsskala z.B. eine Zahl wie 100, eine nominale Bewertung wie „sehr gut“ oder eine binäre Einteilung wie „empfehlenswert/nicht empfehlenswert“ sein.

Zielobjekt das Objekt, für das bezogen auf einen bestimmten Nutzer eine Vorhersage generiert werden soll.

Zielnutzer(in) entsprechend der/die Nutzer(in), für den/die die Vorhersage bezogen auf ein bestimmtes Zielobjekt gemacht werden soll.

Vergleichsobjekt Objekte, mit denen ein Zielobjekt (z.B. bzgl. der Eigenschaften) verglichen wird.

Vergleichsnutzer(in) entsprechender Begriff für Personen, die ein Empfehlungssystem benutzen.

Zur weiteren Vermeidung umständlicher Formulierungen wird die Verwendung solcher Begriffe wie Zielnutzer(in) oder Anwender(in) mal in männlicher und mal in weiblicher Form geschehen.

2 Empfehlungssysteme

„But I need a recommendation, Spock, no vague warnings.“

Capt. Kirk, *Star Trek: Where No Man Has Gone Before*, 1966

Thema dieser Arbeit ist die Untersuchung von automatisierten Empfehlungssystemen bzw. den von ihnen gelieferten Ergebnissen, sowie des praktischen Nutzens dieser Ergebnisse. Deshalb widmet sich dieses Kapitel ausführlich den allgemeinen Gegebenheiten im Zusammenhang mit Empfehlungssystemen. Dabei wird der Leserin und dem Leser immer wieder ein Aspekt begegnen, der schon bei der Darstellung der dieser Arbeit zugrunde liegenden Motivation angesprochen wurde und der ein Grundproblem bei jeder Beschäftigung mit dem Thema Empfehlungssysteme darstellt: Es handelt sich um den fehlenden einheitlichen Rahmen.
Vor allem das kollaborative Filtern wird auch heute noch oft mit dem Begriff „Empfehlungssystem“ gleichgesetzt und eine allgemein anerkannte Kategorisierung der verschiedenen Verfahren existiert nicht.
Dieses Problem wird schon bei der Definition des Begriffs „Empfehlungssystem“ deutlich, die den Anfang dieses Kapitels bildet. Da das Problem historisch begründet sind, folgt im zweiten Teil des Kapitels eine Darstellung der Geschichte von Empfehlungssystemen von den - noch nicht lange zurückliegenden - Anfängen bis heute. Behoben wird das Problem dann durch ein sich anschließendes eigenes Klassifikationssystem für Empfehlungsverfahren. Dieses kann in der Folge sowohl dazu genutzt werden, in der Literatur oft genannte Typen von Empfehlungsverfahren einzuordnen, als auch Verfahren aus anderen Bereichen mit Empfehlungsverfahren im allgemeinen und den in dieser Arbeit entwickelten Empfehlungssystemen im speziellen zu vergleichen.

2.1 Definition

Da der 1997 erschienene Artikel von [RESNICK und VARIAN 1997] nach dem Start der rasanten Entwicklung kollaborativer Empfehlungssysteme der erste Versuch war, einen Überblick dieses Bereichs zu geben und in dem Artikel auch erstmals der allgemeine Begriff „Empfehlungssystem“ anstatt des bis dato benutzten, spezielleren Begriffs „kollaboratives Filtern“ auftauchte, wird die Definition von Resnick und Varian für den Begriff „Empfehlungssystem“ auch heute noch gern benutzt:

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

Nachteil dieser Definition ist jedoch, dass sie auf kollaborative Empfehlungsverfahren zugeschnitten wurde und allenfalls noch auf Verfahren, die demographisch arbeiten erweitert werden kann. Es ist also eine allgemeine Definition nötig, wie sie z.B. [MOONEY und ROY 2000] bieten:

> *„Recommender systems improve access to relevant products and information by making personalized suggestions based on previous examples of a user's likes and dislikes.“*

Diese Definition beschränkt sich nicht auf ein spezielles Verfahren, ist jedoch wiederum zu allgemein, da hier im Gegensatz zur Definition von [TERVEEN und HILL 2001] -

> *„A computational recommender system automates or supports part of the recommendation process.“*

- der Aspekt der Automatisierung fehlt und der Begriff „recommender system“ in der Definition von [MOONEY und ROY 2000] auch durch „humans“ ersetzt werden könnte.
Außerdem weist [BURKE 2002] in seiner Definition

> *„[...] any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options.“*

darauf hin, dass neben dem schon in der Definition von [MOONEY und ROY 2000] vorhanden Begriff „individualized“ auch die Begriffe „interesting or useful“ (die bei [MOONEY und ROY 2000] fehlen) wichtig sind, um Empfehlungsverfahren von einfachen *Information Retrieval*-Systemen oder Suchmaschinen abzugrenzen.
Daher werden die genannten Aspekte in einer eigenen Definition vereinigt:

> *„Mittels Computern implementierte Empfehlungssysteme sind Systeme, die basierend auf Präferenzen eines Nutzers automatisierte, auf diesen speziellen Nutzer personalisierte Empfehlungen interessanter oder nützlicher Objekte/Informationen aus einer großen Menge von Wahlmöglichkeiten generieren. Die Präferenzen eines Nutzers werden dabei durch Interaktion mit diesem gewonnen.“*

Die Formulierung der “Interaktion” wurde deswegen gewählt, um auch implizite Bewertungen von Objekten zuzulassen. Zudem werden zwar in Abschnitt 2.3 der Vollständigkeit halber auch Systeme erwähnt, bei denen Empfehlungen nicht automatisch generiert werden oder nicht personalisiert sind, die erstellte Definition zeigt jedoch (neben dem Titel dieser Arbeit), dass solche Systeme hier nicht betrachtet werden.

2.2 Geschichte der Empfehlungssysteme

Die Inanspruchnahme der Ehre, das erste automatisierte Empfehlungssystem entwickelt und der Öffentlichkeit zur Verfügung gestellt zu haben, ist Gegenstand diverser Rechtsstreite, da Empfehlungssysteme immer mehr kommerzielle Bedeutung gewinnen und diverse Firmen und Personen Patente für die von ihnen entwickelten Systeme angemeldet haben.[1] Meist wird John B. Hey als Erfinder von Empfehlungssystemen, genauer gesagt von kollaborativen Empfehlungssystemen genannt.[2]
Als ehemaliger *MIT*-Student und Fan von Filmen wollte Hey auf für ihn interessante Filme aufmerksam gemacht werden, ohne sich erst durch Inhaltsangaben und Kritiken unzähliger Filme arbeiten zu müssen. Er entwickelte zwei kollaborative Empfehlungsverfahren[3], die später in Videotheken eingesetzt wurden und Kunden nach Eingabe von Bewertungen für zuvor ausgeliehene Filme neue Filme empfahlen. Hey brachte diese technologischen Grundlagen 1987 als Mitbegründer der Firma *LikeMinds*[4] ein, die das Empfehlungssystem *MovieCritic* entwickelte.
Aber auch Gary Robinson, Begründer der Firma *Microvox Systems*, nimmt für sich in Anspruch, den ersten kollaborativen Algorithmus erstellt zu haben und zwar schon 1986, auch wenn er damals kein Patent darauf angemeldet hatte.

[1] So wurde bspw. laut [HEISE 2003] *Amazon* 2003 von der texanischen Firma *Pinpoint* wegen Patentrechtsverletzung verklagt, da *Amazon* kurz zuvor das automatisierte Erzeugen von Kaufempfehlungen zum Patent angemeldet hatte, die Firma *Pinpoint* jedoch bereits seid 1987 ein entsprechendes Patent hält.
[2] Siehe z.B. [FREEDMAN 1998], [PERRY 2002], [GOLDBERG et al. 2001], [FORSTINGER 1999].
[3] Patentnummern 4,870,579 und 4,996,642 in den Jahren 1987 und 1989, die Patentbeschreibungen können im Internet nach Eingabe der genannten Patentnummern abgerufen werden.
[4] Nach einer Fusion heißt die Firma mittlerweile Macromedia.

Das System hieß angeblich *212-ROMANCE* und war ein über das Telefon mittels Tonwahltasten bedienbares Datingsystem, bei dem Kunden potenzielle Partner über ein Verfahren empfohlen wurden, das nach denselben Prinzipien gearbeitet haben soll wie *MovieCritic*. Robinson beruft sich dabei auf den Quellcode des Programms auf den damaligen Originaldisketten sowie einen Ausdruck, beide mit vom System/Drucker automatisch generierten Datumsangaben, sowie diverse Zeugen.[5]
Fünf Jahre später ging die Firma *Muze* einen anderen Weg, der den inhaltsorientierten Empfehlungssystemen (siehe Abschnitt 2.3) nahe steht.[6] Dort katalogisierte man Bücher und Musikstücke und referenzierte diese mit bestimmten Eigenschaften wie Thema oder Genre. Dafür wurden Personen mit professionellem Hintergrund wie z.B. Autoren oder Literaturwissenschaftler angestellt.
So berichtet Kitty Florey, Autorin und Senior Fiction Editor von *Muze*, wie sie 1983, als sie nebenbei als Angestellte in einem Buchhandel arbeitete, ein Gespräch zwischen zwei Kunden mitbekam, die vor der Auslage mit Bestsellern standen. Der eine der Kunden klagte darüber, dass er eigentlich Bestseller nur deshalb kaufen würde, weil er nicht wüsste, welche Bücher er interessant finden könnte.
Nach diesem Vorfall von dem Wunsch erfüllt, dass man ein System entwickeln müsste, welches Kunden nach Eingabe diverser Eigenschaften automatisch entsprechende Bücher empfiehlt, wirkte Florey 1991 bei *Muze* an der Erfüllung dieses Wunsches mit, indem sie das erste einer Reihe von Büchern mit diversen Eigenschaften belegte und diese in das System von *Muze* eintippte.
Dieses System findet laut *Muze* „Muster im Rauschen“, d.h. benutzt mathematische Formeln um jene Bücher oder Musik zu finden, die die höchste Wahrscheinlichkeit besitzen, den vom Konsumenten angegebenen Präferenzen bzw. dessen Kaufhistorie zu entsprechen. Neben vielen Büchereien greifen bekannte Firmen wie *Amazon* oder *Microsoft* auf das System von *Muze* zurück.
Wirklich bekannt geworden sind automatische Empfehlungssysteme jedoch durch das *TAPESTRY*-System von [GOLDBERG et al. 1992], welches dieser 1992 innerhalb der *Palo Alto Research Group* entwickelte. *TAPESTRY* wurde primär für E-Mails entwickelt, konnte aber auch im Bezug auf Newsgroup-Nachrichten angewendet werden und diente dazu, der stetig steigenden Menge an täglichen Neueingängen im persönlichen virtuellen Postfach eines Nutzers zu begegnen. Das *TAPESTRY*-System enthielt neben einer „Indexer“-Komponente, die Schlüsselwörter einer Nachricht extrahierte und als Indizes abspeicherte auch eine „Kommentar“-Komponente, die Kommentare einzelner Nutzer zu bestimmten Nachrichten separat und mit Verweisen auf die jeweiligen Nachrichten speicherte.[7] Mittels einer eigenen Abfragesprache *TQL* konnten Benutzer des Systems Filter für eingehende Nachrichten, basierend auf den Schlüsselwörtern der Indexer-Komponente und den Kommentaren anderer Nutzer einrichten. Durch die Kommentare wurde somit eine Art Bewertungssystem eingerichtet. Nachteil von *TAPESTRY* war jedoch, dass man nur dann den vollen Nutzen aus dem System ziehen konnte, wenn man die anderen Nutzer des Systems kannte, wie z.B. bei der Benutzung in einer Firma oder an einem Universitätslehrstuhl. In diesem Fall konnte man z.B. einen Nachrichtenfilter einrichten, der alle Nachrichten bis auf jene mit den Schlüsselwörtern „information technology“ und „news“ herausfilterte und verwarf, die von dem Kollegen John Doe mit dem Kommentar „worth to read“ versehen worden waren.
TAPESTRY leitete eine rasante Entwicklung im Bereich von Empfehlungssystemen ein und legte irreführenderweise lange Jahre den von [GOLDBERG et al. 1992] geprägten Begriff „kollaboratives Filtern“ als Synonym für Empfehlungssysteme fest.[8]
Die Akzeptanz dieses Synonyms mag dadurch zu erklären sein, dass der kollaborative Ansatz zu jener Zeit etwas völlig neues darstellte, während andere Ansätze meist aus bereits existierenden Forschungsbereichen länger bekannt waren, wie z.B. inhaltsorientierte Empfehlungssysteme, deren Techniken dem *Information Retrieval* entnommen wurden. Oder wie [HILL et al. 1995] es ausdrückten: „*Current human-computer interfaces largely ignore the power of the social strategy.*“
So ist das Hauptaugenmerk der folgenden Jahre vor allem auf die Weiterentwicklung des kollaborati-

[5] Siehe entsprechende Seite im Internet..
[6] Vgl. [FREEDMAN 1998].
[7] Daher leitet sich auch der Begriff „Annotation in Context“ ab, der oft im Zusammenhang mit Aufgaben von Empfehlungssystemen erwähnt wird. Siehe dazu auch Abschnitt 3.2.
[8] Obwohl *TAPESTRY* genau genommen ja nicht nur ein kollaboratives System, sondern durch die Indexer-Komponente eine Kombination von inhaltsorientiertem und kollaborativem System war.

ven Ansatzes gerichtet. Als nachfolgende richtungsweisende Veröffentlichungen und Systeme gelten dabei das *GroupLens*-System von [RESNICK et al. 1994], *RINGO* von [SHARDANAND und MAES 1995], sowie *BELLCORE* von [HILL et al. 1995].
Das Forschungsprojekt *GroupLens*[9], das 1992 an der Universität Minnesota ins Leben gerufen wurde, befasst sich größtenteils mit automatisiertem kollaborativen Filtern. Bekannt wurde *GroupLens* durch das gleichnamige Filtersystem für Newsgroups zwei Jahre später. Hierbei handelte es sich um eine Erweiterung der ursprünglichen *Netnews*-Architektur, um sogenannte „Better Bit Bureau"-Server (*BBB*), die numerische Bewertungen von Newsgroup-Artikeln durch andere Nutzer sammelten und als Berechnungsgrundlage für vorhergesagte Bewertungen für einen speziellen Nutzer verwendeten. Die Berechnungen beruhten dabei auf der Heuristik, dass Nutzer, die in der Vergangenheit bzgl. der Bewertung bestimmter Artikel übereinstimmten, wahrscheinlich auch bei neuen Artikeln eine ähnliche Bewertung abgeben würden. Das System war insofern eine Erweiterung von *TAPESTRY*, dass ein Nutzer die anderen Personen, aus deren Bewertungen seine vorhergesagte Bewertung errechnet wurde, nicht mehr persönlich kennen musste. Dadurch wurde die Verwendung von Pseudonymen ermöglicht, die Privatsphäre von Nutzern geschützt und die Möglichkeiten automatisierter kollaborativer Empfehlungssysteme durch wesentlich größere Gemeinschaften von Nutzern erheblich erweitert.
Das *RINGO*-System, am 1. Juli 1994 zur Verfügung gestellt, war ein Empfehlungssystem für Musik, dass sowohl über ein Web-Interface als auch per E-Mail bedient werden konnte, mit einer 7-stufigen Bewertungsskala arbeitete und verschiedene Funktionen anbot (Liste (nicht) empfehlenswerter Musik-Stücke, Vorhersage konkreter Bewertung für ein einzelnes Stück, Abspeichern/Abrufen von Kommentaren).
Nachteil des Systems war ein hoher Anfangsaufwand, da man vor Benutzung des Systems 125 Musiker/Musikgruppen - teilweise per Zufall, teilweise nach dem Kriterium „meistbewertet" zusammengestellt - bewerten musste, wobei das Feedback der Nutzer zeigte, dass erst eine doppelte bis dreifache Zahl von Anfangsbewertungen zu guten Vorhersagen führte.
Für das *BELLCORE*-System zur Empfehlung von Videos war der Anfangsaufwand für eine Nutzerin sogar noch höher. Das System hatte kein eigenes Interface, sondern basierte auf dem Austausch von E-Mails. Nach Senden einer Mail mit dem Betreff „ratings" an die Domäne *videos@bellcore.com* bekam eine Nutzerin eine Mail mit einer alphabetischen Liste von 500 Videos zugeschickt. 250 der Titel wurden per Zufall zusammengestellt, der Rest bestand aus Mainstream-Filmen. Danach mussten alle 500 Videos mittels einer zehnwertigen Skala bewertet werden, wobei nicht bekannte Videos dementsprechend zu kennzeichnen waren. Nach Zurücksenden dieser Bewertungen bekam die Anwenderin wiederum eine Mail mit diversen Informationen zurückgeschickt. Diese Informationen beinhalteten eine Liste von empfohlenen „must-see"-Filmen, sowie weitere Empfehlungen aufgeteilt nach Genres. Neu an *BELLCORE* war, dass die Empfehlungsliste einer Nutzerin zusätzliche Informationen in Form einer Liste von Nutzern mit dem ähnlichstem Geschmack bezogen auf die Nutzerin enthielt. Zu jedem dieser Nutzer war zudem der von *BELLCORE* berechnete Korrelationswert angegeben. Außerdem konnten mehrere Personen sich zusammen eine einzige Empfehlungsliste zusenden lassen, die Empfehlungen enthielt, die den Geschmack all dieser Personen treffen sollten.
1997 propagierten dann [RESNICK und VARIAN 1997] die Verwendung des allgemeineren Begriffs „recommender system" (Empfehlungssystem) anstatt des bis dato benutzten Ausdrucks „kollaboratives Filtern". Dabei ging es ihnen bei der Einführung des Begriffs vor allem darum, auch anonyme statt personifizierter Empfehlungen sowie das Herausstellen empfehlenswerter Produkte statt nur das Herausfiltern unwichtiger bzw. nicht empfehlenswerter Produkte in einem Begriff zu erfassen. Trotzdem haben sie durch Einführung dieses Begriffs, der schnell allgemein übernommen wurde, maßgeblich dazu beigetragen, auch auf völlig anderen Prinzipien basierende Empfehlungssysteme wieder in das Licht der Aufmerksamkeit zu rücken, obwohl diese alternativen, auf althergebrachten Methoden beruhenden Systeme nie denselben Bekanntheitsgrad für sich beanspruchen konnten, wie die kollaborativen Verfahren. Dementsprechend waren solche Systeme auch nicht so zahlreich vertreten wie ihre kollaborativen Pendants. Beispiele solch alternativer Systeme sind *NewsWeeder* von [LANG 1995] für ein eigenschaftsorientiertes Verfahren, ein für

[9] Nähere Informationen finden sich in [RESNICK et al. 1994] oder auf der *GroupLens*-Webseite. Innerhalb der *GroupLens*-Forschung hat sich auch der *MovieLens*-Datensatz herausgebildet, der in dieser Arbeit benutzt wird.

Forschungszwecke entwickeltes demographisches System von [PAZZANI 1999], *Tête-a-Tête*[10] als Vertreter eines nützlichkeitsbasierten Systems und *Entree* von [BURKE 2002], ein wissensbasiertes Verfahren.
NewsWeeder war ein System zur Filterung von Newsgroup-Texten, wobei die Filterung anhand von Schlüsselwörtern erfolgte, die als Eigenschaften des jeweiligen Textes herangezogen wurden.
[PAZZANI 1999] entwickelte ein demographisches System zur Empfehlung von Restaurants, um die Performance dieses Systems mit dem kollaborativer und eigenschaftsorientierter Verfahren zu vergleichen. Dabei griff er auf den existierenden *Winnow*-Algorithmus[11] zurück, um demographische Daten wie Heimatstadt, Alter und ethnische Zugehörigkeit aus den Webseiten der Anwender des Systems zu extrahieren. Diese Daten wurden im Zusammenhang mit den von den Nutzern abgegebenen Bewertungen für Restaurants benutzt, um solche Restaurants zu empfehlen, die Anwender mit ähnlichem demographischem Hintergrund gut bewertet hatten.
Tête-a-Tête ist ein System für Kauf- und Verkaufs-Agenten, das nach der *Multi Attribute Utility-Theorie* (*MAUT*)[12] arbeitet und bei dem ein Anwender seine Präferenzen bzgl. verschiedener Eigenschaften von Produkten (z.B. Preis und Lieferzeit) angeben muss. Basierend auf diesen Präferenzen erstellen die Agenten Empfehlungen in Form von Produktangeboten, die der Anwender wiederum bzgl. einzelner Kriterien kritisieren und damit die erstellte Nützlichkeitsfunktion verfeinern kann.
Entree wurde 1996 als Restaurant-Führer für die Besucher der *Democratic National Convention* entwickelt und basierte auf Methoden des *Case-Based Reasonings* (*CBR*).[13] Benutzerinnen nannten dabei ein Restaurant oder eine Menge von Kriterien (z.B. *französische Küche*, *junges Publikum*) als Einstiegspunkt und bekamen eine Liste von empfohlenen Restaurants zurückgeliefert. Durch Kritik dieser Liste bezogen auf einzelne Eigenschaften (z.B. *zu teuer*) durch die Nutzerin konnten die Empfehlungen weiter verfeinert werden.
Unabhängig von der verwendeten Methode (kollaborativ, eigenschaftsbasiert, demographisch, etc.) zur Generierung von Empfehlungen wurden die meisten Empfehlungssysteme anfangs im Kontext der Forschung entwickelt und eingesetzt. Ende der 90er Jahre erkannte dann schließlich auch die Wirtschaft das Potenzial solcher Verfahren, da diese durch Personalisierung die Methode liefern, um die *Mass Customization*, die in dem gleichnamigem Buch von [PINE II 1993] gefordert wird, zu erreichen. Durch die häufige Verflechtung von Wirtschaft und Forschung wurde somit auch das wissenschaftliche Interesse am Einsatz von Empfehlungssystemen im kommerziellen Rahmen gesteigert. 1999 weisen [SCHAFER et al. 1999] bereits in einer auf diesen neuen Bereich spezialisierten Untersuchung auf die steigende Wichtigkeit von Empfehlungssystemen im E-commerce hin:

> *„Recommender systems are changing from novelties used by a few E-commerce sites, to serious business tools that are reshaping the world of E-commerce.“*

Zu dieser Zeit hatten Unternehmen verschiedenste Empfehlungsverfahren bereits in ihre Web-Auftritte integriert, um den Umsatz von Büchern[14], Musik-CDs[15], Filmen[16] oder sogar Kleidung[17] zu steigern.
Ende der 90er Jahre setzte neben der Kommerzialisierung auch eine weitere Entwicklung ein. Einige wenige Forscher hatten schon relativ früh die Vorteile einer Integration der verschiedenen Verfahren mittels Hybridsystemen entdeckt (z.B. wurde *Fab*, ein Hybrid aus inhaltbasiertem und kollaborativem Verfahren, bereits 1994 von [BALABANOVIC und SHOHAM 1997] entwickelt und in den folgenden Jahren eingesetzt), größtenteils arbeiteten die Forschungsgruppen jedoch spezialisiert an einer bestimmten Empfehlungsmethode, vorwiegend dem kollaborativen Ansatz, ohne die Verwendung anderer Verfahren in Betracht zu ziehen. Erst langsam realisierte die Mehrheit, dass eine Verbindung der verschiedenen Verfahren in einem System die

[10] Siehe u.a. [MAES et al. 1999].
[11] Siehe [LITTLESTONE 1988], [LITTLESTONE 1989] bzw. [LITTLESTONE 1991].
[12] Siehe [KEENEY und RAIFFA 1976].
[13] Siehe z.B. [KOLODNER 1993].
[14] Z.B. Amazon.
[15] Z.B. CDNOW.
[16] Z.B. Moviefinder.
[17] Z.B. Levis Style Finder.

spezifischen Vorteile der Einzelkomponenten bündeln und die Nachteile kompensieren könnte. Mittlerweile installiert sich langsam ein Zukunftstrend im Bereich der Empfehlungsverfahren, der genau solch eine rasante Entwicklung annehmen könnte, wie einst die kollaborativen Filterverfahren nach ihrer Einführung. Abschließend soll Abb. 2.1 zur anschaulichen Darstellung der Entwicklung von Empfehlungssystemen aus Sicht der Forschung herangezogen werden. Diese Abbildung wurde mit Hilfe der ACM digital library

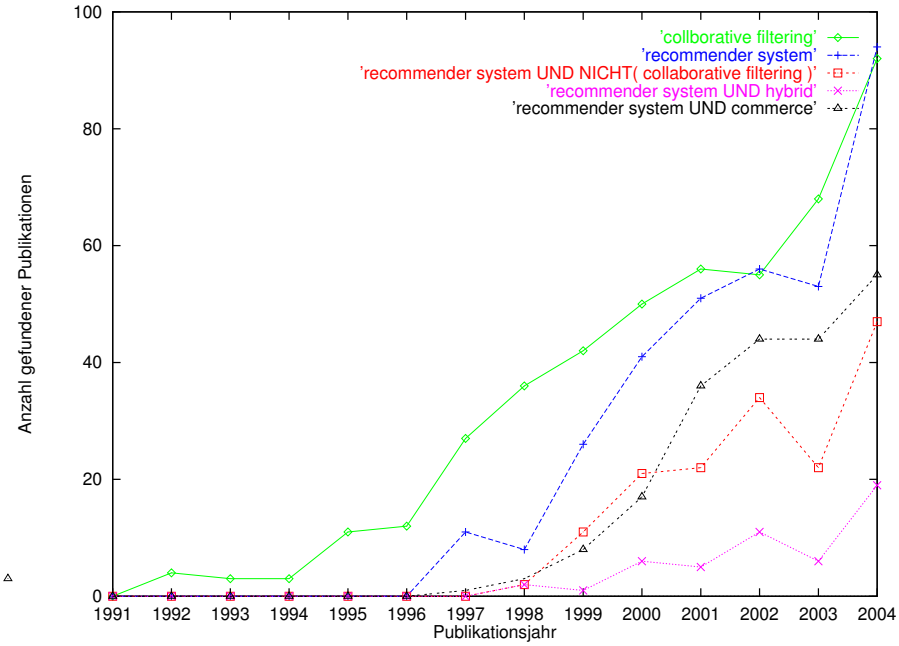


Abbildung 2.1: Geschichtliche Entwicklung von Empfehlungssystemen

erstellt, die eine Suche über allen in dieser digitalen Bibliothek gespeicherten wissenschaftlichen Texten anbietet. Es wurden dabei fünf Suchläufe für die Phrasen „collaborative filtering“, „recommender system“, „recommender system“ ohne gleichzeitiges Vorkommen von „collaborative filtering“, „recommender system“ in boolescher *UND*-Verbindung mit dem Begriff „commerce“, sowie „recommender system“ in *UND*-Verbindung mit dem Begriff „hybrid“ durchgeführt und die Anzahl der gefundenen Publikationen gezählt. Die Ergebnisse werden durch die Graphen in Abb. 2.1 repräsentiert. Die genannten Suchbegriffe durften dabei sowohl im Titel, als auch im Text vorkommen und die unterschiedlichen Relevanzeinstufungen wurden ignoriert, da bei einigen Tests bekannte und erwartete Texte mit einer sehr niedrigen Relevanz bewertet wurden. Weiterhin durfte es sich bei den Treffern um beliebige wissenschaftliche Texte handeln. Diese graphisch dargestellten Suchergebnisse erheben dabei keinerlei Anspruch auf Vollständigkeit (nicht alle wissenschaftlichen Publikationen werden in der *ACM* erfasst), man kann an ihnen jedoch einige der zuvor genannten geschichtlichen Entwicklungen anschaulich nachvollziehen.
Zum einen sieht man die Dominanz der kollaborativen Systeme und des Begriffs des „kollaborativen Filterns“, der meist synonym zu Empfehlungssystemen im allgemeinen gebraucht wird. Dies beginnt mit der Publikation zu dem *TAPESTRY*-System 1992, wo dieser Begriff eingeführt wurde und erfährt seinen richtigen Durchbruch und Beginn einer rasanten Entwicklung im Jahre 1995 als Folge der Veröffentlichungen über die *GroupLens*-, *RINGO*- und *BELLCORE*-Systeme.
Systeme ohne kollaboratives Filtern sind besonders zu Anfang aber auch heutzutage unterrepräsentiert. Der allgemeinere Begriff „Empfehlungssystem“ findet erst 1997 Einführung in die wissenschaftlichen Kreise, angeregt durch das Paper von [RESNICK und VARIAN 1997]. Der neue Begriff verbreitet sich schnell, kann sich jedoch nicht endgültig durchsetzen.
Untersuchungen über den Einsatz von Empfehlungssystemen im kommerziellen Kontext sind anhand der

Graphik erst ab Ende der 90er Jahre überhaupt auszumachen, seit Beginn des neuen Milleniums steigt das Interesse jedoch enorm an, was die wirtschaftliche Bedeutung von Empfehlungsverfahren widerspiegelt. Auch Abhandlungen über Hybridsysteme nehmen ab 2000 langsam zu. Sie sind insgesamt selten vertreten, aber der Sprung der veröffentlichten Publikationen von 2003 auf 2004 gibt Anlass zur Hoffnung auf eine zunehmende Integration und Kooperation der verschiedenen Strömungen im Bereich der Empfehlungssysteme.

2.3 Klassifikation von Empfehlungssystemen

In Abschnitt 1.1 wurde erwähnt, dass der Bereich der Empfehlungssysteme kein einheitliches Klassifikationssystem besitzt, was es für Interessierte sehr schwer macht, sich einen anfänglichen Überblick zu verschaffen. Bestehende Kategorisierungen beschränken sich oft auf einen ganz speziellen Bereich (z.B. kollaborative Systeme[18]) oder stammen aus den Anfangstagen der Empfehlungssysteme[19] und sind somit bei weitem nicht mehr aktuell. Deshalb soll hier kurz und knapp ein eigenes Klassifikationssystem vorgestellt werden, welches eine Unterteilung von Empfehlungssystemen anhand der wichtigsten möglichen Eigenschaften vornimmt. Die allgemeine Situation bei Empfehlungssystemen - gleich welcher Art - sieht man in Abb. 2.2. Nutzer des Empfehlungssystems bewerten über ein Interface ihnen bekannte Objekte aus einer Menge von im System erfassten Objekten. Aus diesen Bewertungen, den Objekten und anderen dem System zur Verfügung stehenden Daten[20] werden dann in einem Generierungsprozess Empfehlungen für die Nutzer erstellt, die ebenfalls über das Interface ausgegeben werden. Die wichtigsten Komponenten des Systems anhand derer eine Klassifikation vorgenommen werden kann sind somit beteiligte Nutzer, Objekte, durchgeführte Bewertungen, generierte Empfehlungen, der Prozess, der diese Bewertungen generiert und das Interface, über das System und Nutzer miteinander kommunizieren. Weitere Details finden sich zum Beispiel in den Publikationen von [Terveen und Hill 2001], [Resnick und Varian 1997], [Herlocker et al. 2004], [Burke 2002], [Sarwar et al. 2000b] oder [Breese et al. 1998].

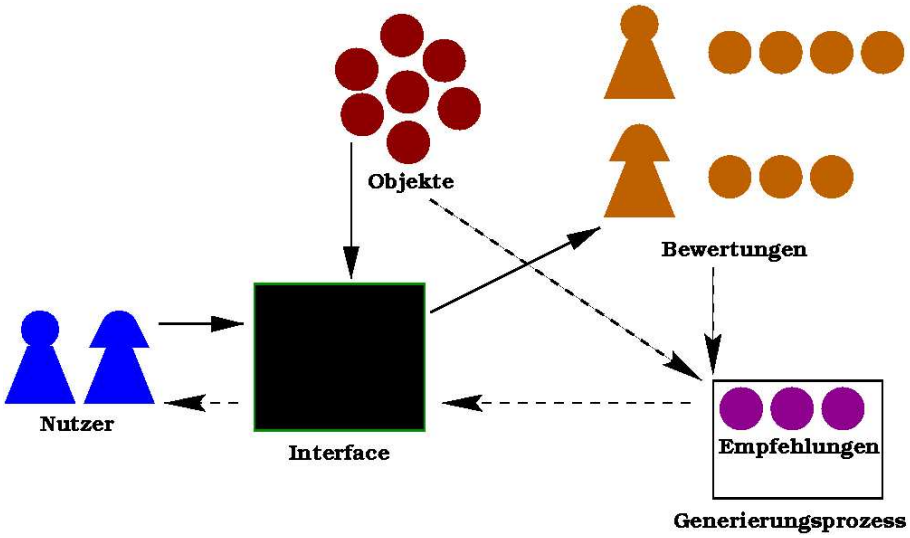


Abbildung 2.2: Komponenten eines Empfehlungssystems

1. **Nutzer**

[18] Siehe [Terveen und Hill 2001].
[19] Bspw. [Resnick und Varian 1997].
[20] In der Abb. der Einfachheit halber nicht extra dargestellt.

- *statisch/dynamisch:* Die Anzahl der Nutzer kann vorab festgelegt sein (z.B. Angehörige einer Firma oder Institution), es können aber auch Veränderungen der Nutzergruppe erlaubt werden.
- *Zugang:* Oft eine direkte Folge des Kriteriums *Interface* → *Kontext* (s.u.). Der Zugang zu einem Empfehlungssystem kann beschränkt sein, z.B. auf Angehörige einer Institution, in der das System als Automat aufgestellt ist oder Personen innerhalb einer Firma oder einer Einrichtung, wo das System im Intranet läuft. Umgekehrt kann aber bspw. auch ein webbasiertes System für alle Personen mit einem Internetzugang offen sein. Wichtig ist in diesem Zusammenhang auch, wie die zugangsberechtigten Personen bzgl. der Gesamtbevölkerung verteilt sind. Viele Systeme sind auf eine strikt abgegrenzte „community" beschränkt.
- *Rolle:* Die Rolle von Nutzern eines System kann auf die der Empfehlungen Suchenden beschränkt sein, es ist aber auch eine symmetrische Beziehung möglich, wo Anwender gleichzeitig auch Empfehlende sein können. Zudem kann es Systeme geben, wo sich die Rolle eines Nutzers ändern kann.

2. **Objekte**

   - *Typ:* Objekte in einem Empfehlungssystem können von verschiedenem Typ sein, wie z.B. Bücher, Videos oder CDs. Auch gibt es Systeme, in den verschiedene Typen gleichzeitig angeboten werden können.
   - *statisch/dynamisch:* Die Menge der Objekte eines Systems kann fest oder einer stetigen Aktualisierung unterworfen sein. Im zweiten Fall sind folgende Kriterien zu beachten:
     - *Zuwachs:* Zahl der Objekte, die pro Aktualisierung dem System hinzugefügt werden.
     - *Frequenz:* Häufigkeit, mit der eine Aktualisierung stattfindet.
   - *Lebensdauer:* Bei vielen Systemen bleibt ein einmal hinzugefügtes Objekt stets erhalten (wie z.B. bei Empfehlungssystemen für Filme). Systeme mit hoher Frequenz und/oder Aktualität können Objekte jedoch mit einer begrenzten Lebensdauer versehen. Dies gilt vor allem für kommerzielle Systeme, bei denen selten gekaufte Produkte aus wirtschaftlichen Gründen aus dem Produktkatalog entfernt werden können.

3. **Bewertungen**

   - *Quelle:* Bewertungen müssen nicht direkt vom gleichen System stammen, in dem sie auch eingesetzt werden. Alternativ können auch Bewertungen von ehemaligen Nutzern eines anderen Systems Verwendung finden oder implizite Bewertungen mittels *Data Mining*-Techniken aus öffentlich zugänglichen Quellen wie Foren oder Webseiten extrahiert werden.
   - *Typ:* Die Form, in welcher die Bewertungen abgegeben werden. Beispiele sind numerische oder nominale Werte, auch ganze Texte, wie Kommentare zu Büchern oder CDs bei *Amazon* sind vorstellbar.
   - *Granularität*: Anzahl der möglichen unterschiedlichen Bewertungen für ein Objekt. Bei den praktischen Versuchen dieser Arbeit wird z.B. eine fünfwertige Bewertungsskala von 1 bis 5 benutzt.
   - *Dimension:* Ein Objekt kann als Ganzes bewertet werden, man kann aber auch verschiedene Dimensionen des Objektes separat bewerten, wie z.B. die Qualität und Lieferzeit eines Produkts. In diesem Fall können weitere mögliche Klassifikationen anhand der folgenden Unterkriterien getroffen werden:
     - *isoliert/Synthese:* Im späteren Verlauf des Empfehlungsprozesses können die Bewertungen der einzelnen Dimensionen ebenfalls isoliert betrachtet werden. Eine andere Möglichkeit ist, aus den Einzelbewertungen eine Gesamtbewertung zu errechnen.
     - *Gewichtung:* Natürlich können die einzelnen Bewertungen wahlweise auch entsprechend ihrer Bedeutung unterschiedlich gewichtet werden.

- *explizit/implizit/gemischt:* Implizite und explizite Bewertungen wurden bereits in Kapitel 1 erklärt. Auch eine Mischform aus beiden Bewertungstypen ist denkbar, um z.B. anfänglich spärlich vorhandene explizite Bewertungen mit impliziten Bewertungen anzureichern. Wenn implizite Bewertungen benutzt werden, kann sich eine weitere Klassifikation anhand der folgenden Kriterien anschließen:
  - *Handlungstypen:* Die Art der Aktionen des Anwenders, die für die impliziten Bewertungen herangezogen wird (z.B. Verweildauer auf einer Webseite, Kaufen eines Produktes). Eine gute und häufig benutzte Abhandlung zu diesem Thema bietet [NICHOLS 1998].
  - *Abbildung:* Die Art, auf die Handlungen auf die einzelnen Werte der gewählten Bewertungsskala abgebildet werden.
  - *offen/verdeckt:* Eine Nutzerin hat meistens Kenntnis davon, dass ihre Aktionen zum Sammeln impliziter Bewertungen dienen. *Spyware* jedoch arbeitet stets verdeckt.[21]

  Wird eine Mischform aus expliziten und impliziten Bewertungen benutzt, sind weitere mögliche Unterscheidungskriterien gegeben:
  - *isoliert/Synthese:* Siehe *Bewertungsdimensionen*.
  - *Gewichtung:* Siehe *Bewertungsdimensionen*.
  - *Lebensdauer:* Die Mischung aus expliziten und impliziten Bewertungen kann dauerhaft benutzt werden, man kann jedoch auch nach einiger Zeit die Verwendung auf einen Bewertungstyp beschränken. Wenn z.B. genügend explizite Bewertungen gesammelt wurden, um akkurate Empfehlungen generieren zu können, kann auf implizite Bewertungen als Kompensation des *cold-start*-Problems[22] verzichtet werden.
- *Mindestzahl:* Die Anzahl der Objekte, die bewertet werden müssen, damit später Empfehlungen generiert werden können. Einfache statistische Systeme (s.u.) brauchen gar keine Objektbewertungen, das oben beschriebene *BELLCORE*-System verlangte in dieser Hinsicht eine Menge Arbeit von einer Anwenderin.
- *Frequenz:* Die Häufigkeit, mit der Empfehlungen von Nutzern abgegeben werden. Systeme mit hoher Frequenz bieten oft eine bessere Vorhersagequalität, weil mehr Daten zur Empfehlungsgenerierung zur Verfügung stehen.
- *flüchtig/persistent:* Die Bewertungen einer Nutzerin können dauerhaft zur Verfügung stehen, so dass sie und damit das Profil der Nutzerin erweitert werden können. Einige Systeme, wie z.B. das *gnod*-System beschränken jedoch die Gültigkeit der Bewertungen auf eine Sitzung.
- *Aufwand:* Das Kosten-Nutzen-Verhältnis für Empfehlungen. Implizite Bewertungen erfordern zum Beispiel gar keinen Aufwand vom Nutzer, jedes Objekt anhand 20 verschiedener Dimensionen zu bewerten dagegen umso mehr.

4. **Empfehlungen**

- *Typ:* Empfehlungen können verschiedene Typen, wie z.B. Objekt oder Link haben. Der Typ wiederum beeinflusst stark die Präsentation von Empfehlungen (siehe entsprechendes Kriterium unter *Interface*).
- *personalisiert/allgemein:* Nicht immer sind Empfehlungen bzgl. der Nutzerinnen personalisiert. Bei einfachen statistischen Systemen werden für jede Person dieselben Empfehlungen generiert, wie z.B. in Form einer Liste der meistgekauften Produkte. Nach der in Abschnitt 2.1 gegebenen Definition gehören diese Systeme nicht zu den automatisierten Empfehlungssystemen, der Vollständigkeit halber und zur Abgrenzung werden sie aber hier genannt.

[21] Auch bei *Spyware* handelt es sich um eine Art von Empfehlungssystem. Handlungen eines Anwenders werden als implizite Bewertungen interpretiert, die zur Erstellung eines Nutzerprofils führen. Empfehlungen werden dann an den Hersteller der Spyware übermittelt, der diese für das Versenden personalisierter Werbemails nutzt. Allgemeine Informationen zu *Spyware* finden sich z.B. in [TITTEL 1999].

[22] Das *cold-start*-Problem wird in Kapitel 3 beschrieben.

- *Quelle:* Entsprechend der angesprochenen Definition müssen Empfehlungssysteme von Computern generiert werden. Auch hier soll aber darauf hingewiesen werden, dass es Systeme gibt, bei denen Menschen die Empfehlungen auswählen (vgl. z.B. *recommender support systems* bei [TERVEEN und HILL 2001] oder *Editor's choice* bei [SARWAR et al. 2000b]). Auch Mischsysteme sind möglich.
- *Empfänger:* Nicht immer ist der Nutzer, aufgrund dessen Bewertungen die Empfehlungen erstellt wurden, auch der Adressat dieser Empfehlungen. Als Alternative sei hier wieder die Spyware erwähnt. Weiterhin können Empfehlungen sowohl für Einzelpersonen, als auch für Gruppen von Nutzern, wie Paare oder Freunde generiert werden. Auch per Broadcast an alle Nutzer eines Systems versandte Empfehlungen sind denkbar.
- *Zugang:* Das Anrecht auf das Abrufen von Empfehlungen ist nicht selbstverständlich. Da die Qualität vieler Empfehlungssysteme maßgeblich von der Menge insgesamt abgegebener Bewertungen abhängt, kann das Recht von Nutzern, Empfehlungen zu erhalten auch an bestimmte Bedingungen, wie das Bewerten einer bestimmten Mindestzahl von Objekten pro Sitzung geknüpft sein.
- *privat/kommerziell/Forschung:* Empfehlungen können Teil eines privaten Systems, eines Forschungssystems oder eines kommerziellen Systems sein. Das Einsatzgebiet kann dabei umfangreiche Auswirkungen auf die Art der generierten Empfehlungen haben. Um Kunden nicht durch falsche Empfehlungen zu verlieren, neigen kommerzielle Systeme bspw. dazu, auf allzu ungewöhnliche Empfehlungen aus Sicht der Präferenzen einer Nutzerin zu verzichten.
- *automatisch/manuell:* Bei einigen Systemen, insbesondere solchen kommerzieller Art, werden die Empfehlungen automatisch generiert. Andere Systeme liefern Empfehlungen nur auf Anfrage. In diesem Fall stellt sich dann die Frage, ob der Empfänger der Empfehlungen die Anfrage initiiert oder eine andere Person.
- *Relation:* Einzelne Empfehlungen können in Relation zueinander stehen oder auch voneinander unabhängig sein. Ein Beispiel für eine Relation zwischen einzelnen Empfehlungen ist gegeben, wenn diese bzgl. bestimmter Eigenschaften gleich sein müssen (siehe z.B. *Empfehlungssequenz* in Abschnitt 3.2).
- *Kosten:* Die Kosten einer falschen Empfehlung oder einer potenziellen Empfehlung, die aber nicht erfolgt ist. Kosten können im günstigsten Fall nur verlorene Zeit sein, aber auch hohe materielle oder auch persönliche Verluste umfassen.
- *Abdeckung:* Der Grad, zu dem Empfehlungen für jedes vom System erfasstes Objekt generiert werden können. Dies ist auch ein wichtiges Qualitätskriterium für Empfehlungssysteme und wird in Abschnitt 3.2 ausführlich behandelt.
- *Varianz Nutzerpräferenzen:* Die Varianz innerhalb der Präferenzen einzelner Nutzer, sofern diese Varianz bekannt ist.
- *Feedback (ja/nein):* Durch Feedback bzgl. der generierten Empfehlungen können nachfolgende Empfehlungen verbessert werden. Leider bieten nicht alle Systeme diese Möglichkeit.

5. **Generierungsprozess**

- *Quelldaten:* Empfehlungen basieren auf Daten, die aus verschiedenen Quellen stammen können:
  - *nur Bewertungen:* Nur die Bewertungen, d.h. Tupel bestehend aus Nutzer, Objekt und Bewertung werden zum Generieren von Empfehlungen herangezogen.
  - *Objektattribute:* Bestimmte Eigenschaften der bewerteten Objekte bzgl. einer Menge vorher festgelegter Kriterien, wie bspw. mitwirkende Schauspieler oder Regisseur bei Filmen.
  - *erweiterte Attribute:* Auch Attribute, die nicht direkt etwas mit den Objekten zu tun haben, werden herangezogen wie z.B. die Zuverlässigkeit einer Firma, bei der ein Produkt bestellt

wird. Bei kommerziellen Systemen kann es sich um Informationen darüber handeln, wie oft ein Produkt gekauft wurde oder wieviele Produkte eines Typs noch im Lager vorhanden sind.

  - *demographische Daten:* Dies sind persönliche Daten der Nutzer wie z.B. Alter, Geschlecht, Beruf oder nationale Zugehörigkeit.
  - *Hintergrundwissen:* Nach Art von Expertensystemen kann auch Hintergrundwissen mit in die Generierung von Empfehlungen einbezogen werden. Ein Beispiel wird im Zusammenhang mit dem *Entree*-System weiter unten angeführt.

- *Aufbereitung der Daten (speicherbasiert/modellbasiert/gemischt):* Einige Systeme bereiten die benutzten Daten vor dem Generieren von Empfehlungen erst auf, während andere auf diesen Schritt verzichten.
  Sog. *Lazy Learner* generieren Empfehlungen direkt aus den gesammelten und im Speicher liegenden Daten, ohne diese zuvor aufzubereiten. Andere Systeme hingegen erstellen zuerst ein Modell der Präferenzen der Nutzerinnen, um sich anschließend der benutzten Daten zu entledigen. Auch eine Mischform ist möglich, wo einzelne Modellwerte im Voraus berechnet, die grundlegenden Daten aber weiterhin gespeichert werden.

- *Repräsentationsform:* Sowohl bei speicherbasierten als auch modellbasierten Systemen können unterschiedliche Repräsentationsformen der Nutzerpräferenzen gewählt werden. Beispiele werden ebenfalls weiter unten angeführt.

- *Lernverfahren:* Wenn Modelle der Präferenzen erstellt werden, kommt dabei oft ein Verfahren zum Erlernen dieses Vorhersagemodells zum Einsatz. Beispiele sind *Expectation Maximization* (*EM*, siehe [DEMPSTER et al. 1977]) für probabilistische Modelle oder *Support Vector Machines* (*SVM*, s.u.).

- *Identifizieren geeigneter Objekte:* Nachdem die endgültige Repräsentationsform der Nutzerpräferenz festgelegt ist, müssen Empfehlungssysteme empfehlenswerte Objekte identifizieren. Oft werden dabei potenzielle Nachbarn anhand eines festgelegten Distanzmaßes gesucht. Es können verschiedenste Distanzmaße zum Einsatz kommen. Einige, die auch bei der Qualitätsbewertung von Empfehlungssystemen Einsatz finden, werden später in Kapitel 3 vorgestellt.

- *Erstellen der Empfehlung:* Beschreibt den Prozess, der aus den identifizierten „Nachbarn“ Empfehlungen generiert. Dabei kann auch eine Gewichtung einzelner Nachbarn vorgenommen werden.

- *Kompensation fehlender Daten:* Fast alle Systeme haben Probleme bei der Empfehlungsgenerierung, wenn zuwenig Daten vorhanden sind. Einige Systeme kompensieren dieses Problem durch verschiedene Maßnahmen, wie z.B. die Verwendung sinnvoller Defaultwerte für die Vorhersage von Bewertungen.

- *Einzelsystem/Hybrid:* Die Empfehlungen können von einem einzelnen System generiert werden, man kann jedoch auch mehrere verschiedene Systeme miteinander kombinieren. Die Art der Kombination der Einzelkomponenten erlaubt eine weitere Klassifikation dieser Hybridsysteme:

  - *gewichtet:* Die Vorhersagen der Einzelkomponenten werden gewichtet und zu einem einzigen Wert kombiniert.
  - *umschaltend:* Für eine Vorhersage wird das Einzelsystem benutzt, welches aufgrund seiner Eigenschaften am besten für die Vorhersage geeignet ist. So kann z.B. bei der Kombination eines kollaborativen und eines eigenschaftsbasierten Systems dann auf das eigenschaftsbasierte System umgeschaltet werden, wenn das kollaborative System wegen geringer Dichte der Bewertungen[23] keine Nutzer mit ähnlicher Präferenz bzgl. des aktuellen Zielnutzers identifizieren kann.

[23] Siehe auch Kapitel 3.

  - *gemischt:* Die Vorhersagen der Einzelsysteme werden nicht zu einem Wert verschmolzen, sondern gemeinsam (z.B. in einer Liste) dargestellt.
  - *Kombination von Eigenschaften:* Attribute verschiedener Systeme bzgl. eines Objekts werden miteinander kombiniert und als Gesamteingabe eines Einzelsystems benutzt. So können kollaborative Daten einfach als weitere Eigenschaften eines Objekts interpretiert und mit den normalen Eigenschaften eines eigenschaftsbasierten Systems kombiniert werden, so dass ein induktiver Regellerner mit diesen kombinierten Eigenschaften ein genaueres Präferenzmodell erlernen kann.
  - *Kaskade:* Die Empfehlungen eines Einzelsystems dienen als Eingabe für das nächste System, um von ihm verfeinert zu werden.
  - *Anreicherung von Eigenschaften:* Ähnlich wie die Kombination von Eigenschaften, nur dass hier nicht die Quelldaten eines Systems als zusätzliche Eigenschaften benutzt werden, sondern die vorhergesagte Bewertung bzw. Klassifikation eines Objekts durch das erste Empfehlungssystem als zusätzliche Eigenschaft herangezogen wird.
  - *Meta-Level:* Das von einem modellbasierten System gelernte Modell wird als Eingabe des anderen Systems benutzt. So können bspw. die von einem System erstellten Vektormodelle, in dem Eigenschaften von Objekten durch einzelne Vektoren repräsentiert werden und wo jeder einzelne Nutzer durch eines dieser Vektormodelle dargestellt wird, von einem kollaborativen System benutzt werden, um Präferenzen einzelner Nutzer (repräsentiert durch die Vektormodelle) miteinander zu vergleichen.

6. **Interface**

   - *Kontext:* Der Kontext, in dem das System eingesetzt wird. Beispiele sind webbasierte Applikationen auf heimischen Rechnern oder Automatensysteme in einer Videothek.
   - *Begrenzung einer Sitzung:* Eine Sitzung, in der Personen Objekte bewerten oder Empfehlungen empfangen kann in irgendeiner Art und Weise begrenzt sein, wie beim bereits erwähnten *gnod*-System, wo eine Sitzung auf eine feste Zahl von Empfehlungen begrenzt wird. Viele Systeme verfügen allerdings nicht über eine solche Einschränkung.
   - *keines/extern/eingebettet:* Nicht immer stellt das System ein spezielles Interface für Anwender zur Verfügung (mit impliziten Bewertungen arbeitende Spyware tut dies aus naheliegenden Gründen nicht). Das Interface kann aber auch in bereits bestehende Applikationen integriert sein, wie z.B. das *GroupLens*-System (s.o.), welches existierenden Newsreadern lediglich Bewertungsknöpfe und Bewertungsvorhersagen hinzufügte.
   - *fest/modular:* Die erwähnten *recommender support systems* stellen lediglich einen Werkzeugkasten zur Verfügung, um menschlichen Empfehlenden eine Möglichkeit zu bieten, ihre Empfehlungen zu präsentieren. Die meisten Systeme arbeiten allerdings mit einem festen unveränderlichen Interface.
   - *online/offline:* Ein System kann sowohl online in direkter Interaktion mit einer Anwenderin arbeiten oder der Austausch von Bewertungen und Empfehlungen kann offline erfolgen, z.B. per E-Mail wie beim bereits beschriebenen *BELLCORE*-System.
   - *Datenschutz (anonym/pseudonym/real):* Viele Nutzer schätzen es aus Datenschutzgründen nicht, dass sie in einem Empfehlungssystem mit ihrem realen Namen auftreten müssen. Die meisten Systeme bieten daher die Möglichkeit, Pseudonyme zu verwenden oder ganz anonym zu agieren. Manche Systeme wie das oben erwähnte *gnod*-System bieten sogar nur anonymen Zugang. Neben dem Namen geben Nutzer mit ihren Objektbewertungen auch andere höchst sensitive Daten an. Beim Einsatz von demographischen Systemen werden weitere persönliche Daten übermittelt. Daher muss auch der Schutz dieser Daten durch das System beurteilt werden.
   - *Darstellung des Bewertungsprozesses:* Die Art, in der das Interface den interaktiven Prozess der Bewertung von Objekten visuell präsentiert (Bewertungsknöpfe, Eingabefelder, etc.).

- *Präsentation von Empfehlungen:* Die visuelle Darstellung der generierten Empfehlungen. Möglichkeiten sind Listen von Empfehlungen, die nach den vorhergesagten Bewertungen für die Einzelobjekte sortiert sind oder nur einzelne Objekte mit ihren vorhergesagten Bewertungen. Diese Präsentation kann aufgrund der Aufgaben, die ein Empfehlungssystem erfüllt, viele Variationen haben. Ausführlich werden mögliche Aufgaben von Empfehlungssystemen in Abschnitt 3.2 dargestellt.
- *Zusatzinformationen:* Einige Systeme stellen neben den Empfehlungen auch zusätzliche nützliche Informationen bereit. Dabei unterscheidet man:
  - *objektbezogene Informationen:* Werden einem Anwender ihm unbekannte Objekte empfohlen, so ist die Anzeige zusätzlicher Informationen zum Objekt, wie z.B. des Autors oder Genres bei Büchern sinnvoll.
  - *interne Informationen:* Zusätzliche Daten, die einer Nutzerin eine bessere Interpretation einer Empfehlung erlauben, können die Qualität von Empfehlungssystemen entscheidend verbessern. Das erwähnte *BELLCORE*-System zeigte neben den Empfehlungen auch eine Liste jener Nutzer mit ihren berechneten Korrelationswerten an, die bzgl. ihrer Präferenz der aktuellen Zielnutzerin am ähnlichsten waren. [HERLOCKER et al. 2004] untersuchte auch Systeme, die zu jeder Empfehlung den Vertrauenswert anzeigten, der den Grad des Vertrauens angibt, den die Systeme zu ihren eigenen Empfehlungen haben (näheres dazu in Kapitel 3.). Auch untersuchten [HERLOCKER et al. 2000] Methoden, die Nutzern das Zustandekommen von spezifischen Empfehlungen bei kollaborativen Verfahren erklären.
- *Kommunikationsfähigkeit:* Empfehlungssysteme können die Kommunikation fördern, indem ihr Interface Möglichkeiten bietet, dass Anwender z.B. über Foren miteinander in Kontakt treten können.

Empfehlungssysteme, die mit dem vorgestellten System klassifiziert werden, können nicht nur untereinander besser verglichen werden, sondern ermöglichen auch den Vergleich mit anderen Methoden, die nicht direkt dem Bereich der Empfehlungssysteme zuzuordnen sind, aber auf diesen Bereich hin modifiziert werden können. Im Folgenden sollen für beide Verwendungsmöglichkeiten Beispiele vorgestellt werden.

2.3.1 Beispiele für Klassifikationen

Da in der Literatur immer wieder einige häufig benutzte Typen von Empfehlungssystemen benannt werden, sollen diese im Folgenden kurz aufgeführt und anhand einiger der oben genannten Kriterien klassifiziert werden. Für weiterführende Erläuterungen bzgl. dieser Empfehlungsverfahren ist die Publikation von [BURKE 2002] zu empfehlen.

- *Statistische Empfehlungssysteme*
  Dieser Typ von Empfehlungssystemen findet bei [SARWAR et al. 2000b] Erwähnung und stellt wohl die einfachste Art Empfehlungen zu produzieren dar. Diese Systeme ziehen die vorhandenen Bewertungsdaten aller Nutzer und evtl. erweiterte Attribute wie Verkaufsdaten als Quelle für einfache statistische Erhebungen heran. Die daraus resultierenden Empfehlungen können sowohl ansatzweise personalisiert sein („Personen, die Bücher dieses Autoren gekauft haben, haben auch folgende Bücher gekauft“) als auch gar nicht personalisiert (Bestseller des Monats).

- *Kollaborative Empfehlungssysteme*
  Kollaborative Empfehlungssysteme, manchmal auch „soziale Empfehlungssysteme“ genannt, erzeugen personalisierte Empfehlungen, indem als Quelldaten nur die Bewertungen der Nutzer des Systems verwendet werden. Dabei werden die Nutzerpräferenzen oft ohne weitere Aufbereitung, d.h. speicherbasiert als Nutzer-Objekt-Matrix repräsentiert, wobei Bewertungen von Objekten durch Nutzer die Einträge dieser Matrix darstellen. Die meisten Verfahren dieses Typs identifizieren potenzielle Nachbarn einer Zielnutzerin, indem sie zur Laufzeit die Bewertungen aller anderen Vergleichsnutzer

betrachten und anhand eines Distanzmaßes wie z.B. der Korrelation[24] Nutzer mit ähnlicher Präferenz bzgl. der Zielnutzerin heraussuchen. Die Bewertungen dieser Nutzer gehen dann gewichtet nach ihrer Ähnlichkeit in die vorhergesagte Bewertung für das Zielobjekt ein. In diesem Bereich existieren auch einige modellbasierte Ansätze wie der von [SARWAR et al. 2000a], wo *Singular Value Decomposition* angewendet wird, um dem *sparsity*-Problem[25] entgegen zu wirken und die Skalierung zu verbessern. Auch gemischte kollaborative Systeme, die Teile eines Modells im Voraus berechnen existieren. Der in dieser Arbeit in Kapitel 5 benutzte kollaborative Ansatz nach [YU et al. 2003] berechnet z.B. zuerst Korrelationsgewichte für Paare von Objekten, was einige Zeit in Anspruch nehmen kann. Beim Erstellen der Empfehlungen kann dann neben den Bewertungen der Nutzer schnell auf diese vorberechneten Gewichte zugegriffen werden.

- *Eigenschaftsbasierte Empfehlungssysteme*
  Diese Empfehlungssysteme, die oft auch als „inhaltsbasierte Systeme“ bezeichnet werden, nutzen als Quelldaten neben den Bewertungen des Zielnutzers nur die Attributwerte der Objekte, die vom Zielnutzer als gut bewertet wurden. Diese vergleichen sie mit den Attributwerten anderer Objekte. Objekte mit ähnlichen Attributwerten werden dann dem Zielnutzer empfohlen. Häufig wird ein Modell verwendet, in dem Attributwerte einzelner Objekte als Vektoren dargestellt werden. Solch ein *Vektorraum-Modell* wird auch als eigenschaftsbasierte Komponente des Hybridsystems in Kapitel 6 benutzt.
  Der Spezialfall eigenschaftsbasierter Empfehlungssysteme, die *regelbasierten Verfahren*, erzeugt hingegen ein Modell der Präferenzen des Zielnutzers, indem Regeln erlernt werden, die alle gut bewerteten Objekte beschreiben und gleichzeitig alle schlecht bewerteten Objekte ausschließen.

- *Demographische Empfehlungssysteme*
  Bei diesem Typ von Empfehlungssystem werden zu einer Zielnutzerin ähnliche Personen über demographische Informationen als Quelldaten identifiziert. Dieser Zielnutzerin werden dann Objekte empfohlen, die Vergleichsnutzer mit ähnlichen demographischen Daten als gut bewertet haben, d.h. Personen, die in etwa in ihrem Alter sind, in derselben Region wohnen oder derselben Berufsgruppe angehören.

- *Nützlichkeitsbasierte Empfehlungssysteme*
  Bei diesen Systemen wird die Präferenz eines Nutzers durch eine Nützlichkeitsfunktion modelliert. Dafür muss der Zielnutzer ein einziges (Modell-)Objekt bzgl. mehrerer Dimensionen bewerten, wobei die Dimensionen nicht nur aus Attributen des Objekts, sondern auch aus anderen, also erweiterten Attributen bestehen können. Die Bewertungen geben dann die Wichtigkeit an, die die einzelnen Attribute für den Zielnutzer haben und werden bei einem Vergleich mit anderen Objekten entsprechend gewichtet. Die Bewertung kann bzgl. des Interfaces über einen Fragenkatalog erfolgen oder aber der Nutzer muss die Nützlichkeitsfunktion direkt definieren.

- *Wissensbasierte Empfehlungssysteme*
  Im Prinzip sind diese Systeme den nützlichkeitsbasierten Verfahren sehr ähnlich, bis auf die Tatsache, dass zusätzlich Hintergrundwissen als Quelldaten genutzt wird. So beinhaltet dieses Hintergrundwissen für das bereits erwähnte *Entree*-System zum Empfehlen von Restaurants z.B., dass die „Pazifische New Wave“-Küche sowohl asiatische, als auch französische Einflüsse enthält. Somit können einer Anwenderin, die asiatische Küche mag, auch Restaurants der Kategorie „Pazifische New Wave“-Küche empfohlen werden.

2.3.2 Vergleich mit anderen Verfahren

Die erstellte Klassifikation ermöglicht es auch, die in dieser Arbeit benutzten Empfehlungssysteme gegenüber anderen, ähnlichen Ansätzen abzugrenzen. Im Folgenden sollen deshalb einige wichtige alternative

[24] Verschiedene Korrelationsmaße werden in Kapitel 3 vorgestellt.
[25] Siehe Kapitel 3.

Verfahren vorgestellt werden.

Clustering nach verschiedenen (überlappenden) Eigenschaften
[UNGAR und FOSTER 1998] schlagen für kollaboratives Filtern eine spezielle *Clustering*-Methode[26] vor, die auf einem probablilistischen Modell beruht und Objekte anhand der Nutzer die sie bewertet haben gruppiert, während die Nutzer wiederum in Gruppen solcher Personen aufgeteilt werden, die ähnliche Gruppen von Objekten bewertet haben. Die Parameter des Modells sind die Wahrscheinlichkeiten, dass ein Nutzer/Objekt zu einer bestimmten Gruppe (*Cluster*) gehört und die Nutzerin aus einer bestimmten Nutzergruppe Objekte aus einer bestimmten Objektgruppe bewertet hat, wobei diese Parameter durch geeignete Methoden (siehe [UNGAR und FOSTER 1998]) abgeschätzt werden. Im Gegensatz zum Clustering kollaborativer Verfahren (Gruppieren von Nutzern mit ähnlichem Geschmack) haben die in diesem Modell benutzten statistischen Wahrscheinlichkeiten den Vorteil, dass Nutzer in verschiedenen Nutzergruppen gleichzeitig auftauchen können (wenn sie verschiedene unterschiedliche Präferenzen haben) und damit kein „Gray Sheep"-Problem[27] auftritt. Zudem kann das Clustering nach beliebig vielen verschiedenen Attributen durchgeführt werden, so dass bspw. auch Hintergrundwissen in Form von inhaltlichen Objekteigenschaften genutzt werden kann, um die Objekte vorab nach diesen Eigenschaften zu clustern (z.B. Filme nach mitwirkenden Schauspielern). Dadurch ist solch ein Clustering verglichen mit kollaborativen Verfahren weit weniger anfällig für geringe Datendichten (*sparsity*, siehe auch Kapitel 3). Trotzdem wurde dieses Modell hier weder in seiner Grundform benutzt, da in dieser Arbeit „typische" kollaborative Verfahren untersucht werden sollten, noch in erweiterter Form mit inhaltlichen Attributen, wo es zu einem kollaborativ-eigenschaftsbasierten Hybrid vergleichbar geworden wäre. Außerdem hätte es sich dabei eher um einen Eigenschaften anreichernden Hybrid, also ein untrennbares Gesamtsystem gehandelt, das nicht auf dem gewählten kollaborativen System aus Kapitel 5 hätte aufbauen können, wodurch Vergleiche der Versuchsergebnisse schwierig geworden wären.

Clustering von Nutzern nach Interessensmustern
Einen ebenfalls statistischen Ansatz verfolgt [HOFMANN 2001]. Dabei wird ein normales, im kollaborativen Filtern benutztes Bewertungstupel bestehend aus Nutzer, Objekt und Bewertung um eine vierte Komponente erweitert, die den „Grund" angibt, warum der Nutzer diese Wertung für das Objekt abgegeben hat. Jeder Grund gibt ein bestimmtes allgemeines Interessensmuster wieder und fasst damit im Gegensatz zu Standardverfahren im kollaborativen Bereich die Bewertungen eines Nutzers in Gruppen gleichen Interesses zusammen, was die Realität, in der ein Nutzer verschiedenste Interessen haben kann, besser widerspiegelt und genauere Vorhersagen ermöglicht, die zudem durch die Interessen erklärt werden können. Das daraus berechnete probabilistische Modell gibt für jeden Nutzer die Wahrscheinlichkeit an, mit der er und eine Teilmenge der von ihm bewerteten Objekte einem bestimmten Interessensmuster zuzurechnen ist. Diese Dekomposition der Nutzerbewertungen ermöglicht eine wesentlich kompaktere Darstellung der Präferenzen eines Nutzers, verkleinert damit den benötigten Speicherbedarf und - aufgrund der Tatsache, dass in Tests optimale Ergebnisse mit max. 100 verschiedenen Interessensmustern erzielt wurden - die Anzahl der benötigten Vergleiche für eine Vorhersage, was zu einer allgemeinen Beschleunigung führt. Es handelt sich hier also um ein modellbasiertes kollaboratives Verfahren, dass theoretisch auch in dieser Arbeit hätte benutzt werden können, jedoch ging der Autor am Anfang davon aus, dass sich das letzten Endes benutzte Verfahren nach [YU et al. 2003] speicherbasiert implementieren ließe,[28] womit im Gegensatz zum Modell von [HOFMANN 2001] keine Trainingszeit benötigt worden wäre.

Alternatives IBL-Verfahren - Klassifikationsgenauigkeit statt Korrelation der Bewertungen
[AHA et al. 1991] gehen bzgl. der Verbesserung des Standard-*IBL*-Verfahrens[29] einen ähnlichen Weg wie

[26] Einen allgemeinen Überblick bzgl. *Clustering*-Verfahren bieten z.B. [JAIN et al. 1999].
[27] Das „Gray Sheep"-Problem wird in Kapitel 3 beschrieben.
[28] Erst später stellte sich heraus, dass aus Performancegründen doch ein gemischt speicher- und modellbasierter Ansatz beim Algorithmus von [YU et al. 2003] verwendet werden musste. Siehe auch Abschnitt 5.1.2.
[29] *IBL* steht für *Instanzenbasiertes Lernen* und wird in Kapitel 5 genauer berschrieben.

[YU et al. 2003]. Auch sie reduzieren den Speicherverbrauch, der sich bei normalen *IBL*-Verfahren als Folge des Speicherns aller Trainingsinstanzen ergibt, indem sie nur solche Instanzen speichern, die für eine annähernd korrekte Vorhersage aller bisher gesehenen Trainingsbeispiele unbedingt nötig sind. Ihr *IB2* genanntes verbessertes Verfahren speichert dafür nur die falsch klassifizierten Beispiele, da diese nahe der Grenze liegen, die positive und negative Beispiele voneinander trennt und damit eine Approximation dieser Grenze darstellen. Um Rauschen zu vermindern, speichern sie in ihrer nochmals verbesserten *IB3*-Version zusätzlich die bisherige Klassifikationsgenauigkeit der gespeicherten Instanzen. Ihre „wait and see“ genannte Methode entfernt dann in jedem Schritt all die gespeicherten Instanzen, die bzgl. ihrer Klassifikationsgenauigkeit nicht akzeptabel und somit als Rauschen aufzufassen sind. Hier wird also sowohl die Minimierung des Rauschens als auch des benötigten Speichers mit einer Instanzenselektion erreicht, die auf der Klassifikationsgenauigkeit beruht. [YU et al. 2003] hingegen selektieren Instanzen aufgrund ihrer „generellen Rationalität“, die letzten Endes auf der Korrelation von Bewertungen als Maß beruht. Außerdem setzen sie zur Minderung des Rauschens nicht nur die Selektion von Instanzen, sondern auch die Gewichtung von Eigenschaften dieser Instanzen ein, wozu ebenfalls die Korrelation von Bewertungen benutzt wird. Ein Vergleich der Performance des Ansatzes von [AHA et al. 1991] im Gegensatz zu dem von [YU et al. 2003] wäre sicherlich interessant, würde aber den Rahmen dieser Arbeit sprengen.

Regelbasierte Empfehlungssysteme - Finden außergewöhnlicher Empfehlungen

Regelbasierte Empfehlungssysteme werden in einigen wissenschaftlichen Publikationen gerne als eigenständige Kategorie in diesem Bereich angeführt. Wie oben bereits erwähnt, handelt es sich bei ihnen jedoch nur um eine spezielle Form modellbasierter eigenschaftsorientierter Systeme. Regelbasierte Verfahren arbeiten mit den inhaltlichen Eigenschaften von Objekten und erstellen daraus eine kompakte, in Regeln gefasste Generalisierung, die alle positiven Trainingsbeispiele umfasst und alle negativen Beispiele ausschließt. Sie eignen sich daher besonders, um Anwendern von Empfehlungssystemen eine verständliche Begründung für das Generieren einer Empfehlung zu liefern. Wie eigenschaftsbasierte Systeme haben auch regelorientierte Verfahren bei normaler Anwendung den Nachteil, dass sie keine außergewöhnlichen Empfehlungen liefern können.[30]

Man kann regelbasierte Systeme jedoch auch so einsetzen, dass gerade außergewöhnliche Empfehlungen gefunden werden. So benutzt [MORIK 2002] eine Vorgehensweise, bei der zuerst globale Regeln gelernt werden, die meist nicht interessant sind. Dazu wird die Gesamtmenge aller Beispiele genutzt (im Fall von Empfehlungssystemen für Filme wären dies alle Filme in der Datenbank) und die Suche nach Regeln bzgl. der Komplexität handhabbar gemacht, indem die Regelsprache syntaktisch eingeschränkt, die Dimensionalität der Beispiele (d.h. die Anzahl der betrachteten Eigenschaften von Filmen) reduziert und eine kostengünstige Lernstrategie benutzt wird. Dieser erste Schritt dient dazu, potenziell interessante Instanzen (Filme) auszusortieren, die sich entweder dadurch auszeichnen, dass sie Ausnahmen einer der gelernten globalen Regeln sind, von keiner der gelernten Regeln abgedeckt werden oder als negative Beispiele die Akzeptanz einer eigentlich erwarteten globalen Regel verhindern. In einem zweiten Schritt wird dann auf dieser meist kleinen Menge von Beispielen erneut gelernt, um Regeln für diese „Outlier“ zu finden. Dazu wird die Intensität der Suche nach Regeln erhöht, indem die zuvor genannten Einschränkungen bzgl. der Syntax der Regelsprache, der Dimensionalität der Beispiele und der Komplexität der Lernstrategie gelockert werden. [MORIK 2002] benutzt dazu eine Version des *Rule Discovery Tool* (*RDT/DM*)[31] die in *Java* implementiert ist und direkt auf *SQL*-Datenbanken eingesetzt werden kann. Prinzipiell hätte auch diese regelbasierte Methode anstelle des eigenschaftsorientierten Verfahrens in Kapitel 6 eingesetzt werden können, um die Interessantheit von Empfehlungen zu verbessern. Allerdings ist die Auswahl der Regelschemata, die von *RDT/DM* benutzt werden, um die Syntax der Regelsprache einzuschränken, ebenso kritisch, wie die Auswahl der betrachteten Attribute (Eigenschaften) für Instanzen (Filme), der Abbildungen von Relationen und Attributen in der Datenbank auf Prädikate der Regeln und des benutzten Akzeptanzkriteriums für Regeln. Die Sorgfalt, die auf diesen Auswahlprozess hätte verwendet werden müssen, hätte den zeitlichen Rahmen dieser Arbeit gesprengt.

[30] Vgl. Abschnitt 3.1

[31] *RDT* wird ebenfalls in der Publikation von [MORIK 2002] beschrieben.

Verbindung von kollaborativen und inhaltlichen Daten, Instanzen und relationalen Fakten
RIBL, das *Relational Instance-Based Learning*-System von [EMDE und WETTSCHERECK 1996], verbindet den großen Vorteil eines regelbasierten Lernsystems, komplexe Situationen durch Relationen einfach auszudrücken mit den Vorzügen instanzenbasierter Lernverfahren, auch mit kontinuierlichen Attribut- und Klassifikationswerten umgehen zu können und tolerant gegenüber fehlenden Attributwerten oder Rauschen in Attributwerten zu sein. Dazu wird *RIBL* als externe Erweiterung des Wissensentdeckungs- und Lernsystems *MOBAL* (siehe [MORIK et al. 1993] und [SOMMER et al. 1994]) betrieben und erstellt aus den von *MOBAL* gelieferten Fakten für jedes Beispielfakt eine entsprechende Beispielinstanz für den *RIBL*-Algorithmus. Solch eine Instanz ist die Menge aller Fakten, die mind. ein Argument aus dem Beispielfakt enthalten oder über andere Argumente mit einem Argument des Beispielfakts verbunden sind. Die max. Tiefe bis zu der diese Verbindung reichen darf, kann dabei vom Anwender angegeben werden. Ähnlichkeiten zwischen zwei Instanzen werden berechnet, indem jeweils die Werte zweier Argumente der Instanzen und die Objekte miteinander verglichen werden, die aufgrund einer Relation mit diesen Argumenten verbunden sind, wobei der Vergleich dieser Objekte wieder aus dem Vergleich der Werte ihrer Argumente und der mit ihnen verbundenen Objekte besteht, usw. Die Tiefe der miteinander verbundenen Argumente wird beim Vergleich entsprechend berücksichtigt (je größer die Tiefe, desto geringer die Gewichtung). Für die letztendliche Klassifikation einer neuen Instanz werden dann die Klassifikationen der k aufgrund ihrer berechneten Ähnlichkeit (Distanz) zur Instanz nächsten Nachbarn herangezogen. Auch besteht die Möglichkeit, Prädikate (der Fakten in der eine Instanz repräsentierenden Menge) und deren Attribute zu gewichten. Dazu wird ein bestehender Gewichtungsalgorithmus, der als Eingabe nur die Ähnlichkeit zweier Instanzen bzw. Attribute benötigt, entsprechend modifiziert, so dass auch hier die Möglichkeit berücksichtigt wird, dass bestimmte Argumente in bestimmten Tiefen große Relevanz und in anderen gar keine Relevanz besitzen (z.B. das Argument `männlich` für die `hat_Enkelsohn`-Relation).
Bezogen auf die Empfehlungssysteme dieser Arbeit hätte man sich die Tatsache zunutze machen können, dass *RIBL* sowohl auf „normalen“ *IBL*-geeigneten Daten, als auch relationalen Daten eingesetzt werden kann. So hätten kollaborative Daten nach entsprechender Aufbereitung in relationale Form (z.B. das Prädikat `Bewertung(U,O,R)` mit den Argumenten `U` für Nutzer, `O` für Objekt und `R` für Bewertung) mit inhaltlichen Informationen kombiniert werden können. So hätte ähnlich wie bei den erwähnten Interessensmustern von [HOFMANN 2001] (s.o.) mittels solcher Fakten wie

```
Bewertung(u1,o,r1), Bewertung(u2,o,r2), Genre(o,g)
```

bei der Berechnung von Vorhersagen berücksichtigt werden können, dass Nutzer in einem Interessensgebiet (hier ein bestimmtes Genre) miteinander korrelieren, in anderen Gebieten dagegen gar nicht. Allerdings sind die exakten Methoden der Instanzengenerierung und Ähnlichkeitsberechnung in *RIBL*, wie oben grob zusammengefasst, sehr kompliziert und somit hätte im Gegensatz zum selbst erstellten Hybridsystem in Kapitel 6 die Tatsache, warum bestimmte Einstellungen zu Verbesserungen oder Verschlechterungen führen, schwer begründet werden können.

Alternatives Verbessern der Interessantheit durch Co-Training von rein kollaborativem und eigenschaftsbasiertem Verfahren
[BLUM und MITCHELL 1998] liefern einen interessanten Ansatz für Lernumgebungen, in denen verschiedene Komponenten der zu klassifizierenden Instanzen von verschiedenen voneinander unabhängigen Lernverfahren benutzt werden können, um - genügend Trainingsbeispiele vorausgesetzt - die Klassifikationen der Gesamtinstanzen korrekt vorherzusagen. Fehlen in solch einer Umgebung gerade ausreichend viele klassifizierte Beispiele zum Training, aber hat man dafür umso mehr unklassifizierte Beispiele vorliegen, so kann man diese unklassifizierten Beispiele dazu nutzen, die Trainingsmenge anzureichern. Seien bspw. Instanzen x mit zwei Komponenten $x1$ und $x2$ gegeben, d.h. $x = (x1, x2)$ und gebe es zwei verschiedene zu erlernende Klassifikatoren $h1$ und $h2$ die alle Instanzen richtig klassifizieren, wobei $h1$ seine Klassifikation für eine Instanz x nur anhand Betrachtung der Komponente $x1$ vornimmt, während $h2$ Instanzen x nur aufgrund der Komponente $x2$ klassifiziert. Sei außerdem ein weiterer gesuchter Klassifikator h gegeben, der die Instanzen x richtig klassifiziert, dabei aber die Gesamtinstanz x statt einzelner Komponenten betrachtet.

Dann wird beim *Co-Training* die vereinfachende Annahme getroffen, dass alle in der Realität gegebenen Instanzen konsistent sind, d.h. für die Instanz $x = (x1, x2)$ die Komponenten $x1$ und $x2$ so gegeben sind, dass $h1$ die Instanz x aufgrund von $x1$ genau derselben Klasse k zuordnet, wie $h2$ anhand der Komponente $x2$ oder $h1(x1) = h2(x2) = h(x) = k$. Basierend auf dieser Annahme können dann mit Hilfe der wenigen vorliegenden, klassifizierten Trainingsbeispiele ungenaue Klassifikatoren $h1$ und $h2$ erlernt werden. Nach diesem initialen Training wählt man per Zufall eine bestimmte Anzahl der unklassifizierten Instanzen aus, lässt $h1$ und $h2$ Klassifikationsvorhersagen bzgl. dieser Instanzen vornehmen und fügt diese Instanzen mit ihrer vorhergesagten Klassifizierung der Trainingsmenge hinzu, woraufhin man anhand dieser angereicherten Trainingsmenge die Klassifikatoren $h1$ und $h2$ neu erlernt. Dieses Vorgehen führt man eine vorher festgelegte Anzahl von Iterationen aus, um wesentlich genauere Klassifikatoren zu erlernen als nur anhand der wenigen inital klassifizierten Trainingsbeispiele möglich.
Im Rahmen dieser Arbeit wäre eine verbesserte Vorhersage der Interessantheit von Empfehlungen denkbar. Für die praktischen Versuche der Arbeit wurde ein kollaborativer Empfehlungsalgorithmus (siehe Kapitel 5) implementiert und die von ihm generierten Empfehlungen wurden von den Teilnehmern der Versuche bzgl. ihrer Interessantheit bewertet. Um die Arbeit für die Teilnehmer in Grenzen zu halten, wurden von jedem nur 20 dieser Bewertungen (Klassifikationen) gesammelt, also eine sehr begrenzte Anzahl. In Kapitel 6 wurde ein eigenschaftsbasierter „Filteraufsatz“ für den kollaborativen Algorithmus und damit ein Hybridsystem erstellt, um potenziell uninteressante Empfehlungen des kollaborativen Algorithmus herauszufiltern und so die Interessantheit zu verbessern. Alternativ dazu hätte man auch im zweiten Schritt ein rein inhaltbasiertes Verfahren implementieren und das *Co-Training* für dieses Verfahren und das kollaborative Verfahren anhand der 20 ersten Bewertungen pro Nutzer durchführen können, um anschließend ein gemischtes Hybridsystem aus beiden Verfahren zu erzeugen. Aufgrund der Ergebnisse des praktischen Versuchs mit dem kollaborativen Algorithmus erschien das in Kapitel 6 gewählte Hybridsystem jedoch sinnvoller (siehe Abschnitt *Auswahlkriterien* in Kapitel 6).

Generalisierung eines kollaborativen-eigenschaftsbasierten Hybrids für Empfehlungssysteme
[TSOCHANTARIDIS und HOFMANN 2002] gehen einen anderen Weg als [BLUM und MITCHELL 1998] bei der Nutzung unklassifizierter Instanzen als Trainingsbeispiele. Anstatt unterschiedliche Klassifikatoren separat auf den vorhandenen klassifizierten Beispielen lernen zu lassen, untersuchen sie die Beziehungen der einzelnen Klassifikatoren untereinander, um diese gleichzeitig erlernen zu können (*Polycategorial Classification*).[32] In einem ersten Schritt berechnen sie ein probabilistisches Modell, dass aufgrund der erwähnten Abhängigkeiten für jede Instanz aus der Trainingsmenge eines der Klassifikatoren die Wahrscheinlichkeit für diesen Klassifikator angibt, eine bestimmte Klassifikation für die betrachtete Instanz zu generieren. Danach werden die Trainingsmengen für die einzelnen Klassifikatoren mit aufgrund des Modells probabilistisch klassifizierten Instanzen aus der Menge der zuvor unklassifizierten Instanzen angereichert. Diese angereicherten Trainingsmengen werden dann als Eingabe für die bekannten *Support Vector Machines* (siehe [VAPNIK 1995]) benutzt, die auf dem Prinzip der *margin maximization* beruhen und versuchen, den Abstand zwischen der Hyperebene, die Beispiele verschiedener Klassen voneinander trennt und den klassifizierten Trainingspunkten zu maximieren. D.h. Ziel von *SVM*s ist es, alle Instanzen der Trainingsmenge korrekt zu klassifizieren, dabei die Funktion, die die Instanzen der verschiedenen Klassen voneinander trennt jedoch so „einfach“ wie möglich zu halten. Eine Erweiterung der *SVM*s, die *transduktiven SVM*s [33] (*TSVM*) beachten zusätzlich unklassifizierte Daten, um verlässlichere Schätzungen des optimalen Diskriminanten, der für die Trennung der verschiedenen Klasseninstanzen zuständig ist, zu erhalten. Idee ist dabei, dass ein Diskriminant der nur wenig Abstand zu unklassifizierten Instanzen aufweist, keine gute Separierung der einzelnen Klassen bietet, unabhängig davon, welcher Klasse die betrachteten Instanzen letztendlich angehören. [TSOCHANTARIDIS und HOFMANN 2002] erweitern diese *TSVM*s entsprechend, um auch die von ihnen benutzten probabilistischen Klassifikationen verarbeiten zu können.
Mit dieser Methode generalisieren [TSOCHANTARIDIS und HOFMANN 2002] den bei Empfehlungssystemen (und auch hier in dieser Arbeit) stattfindenden Versuch, kollaborative und inhaltliche Daten von In-

[32] Wobei sie von der Annahme ausgehen, dass solche Abhängigkeiten zwischen den Klassifikatoren existieren.
[33] Siehe [JOACHIMS 1999].

stanzen miteinander zu verbinden. Die verschiedenen Klassifikatoren entsprechen dabei den verschiedenen Nutzern, die eine Teilmenge von Objekten gemeinsam bewertet haben. Ähnliche Bewertungen für diese Objekte zeigen dabei Abhängigkeiten der Klassifikatoren (Nutzer) untereinander auf und können benutzt werden, um die Bewertungen nicht bekannter Objekte für eine Zielnutzerin vorherzusagen. Unterschied zu dem in dieser Arbeit benutzten Ansatz von [YU et al. 2003][34] ist, dass Yu statt eines probabilistischen Maßes die Korrelation zwischen den Bewertungen von Anwendern benutzt.
Bei [TSOCHANTARIDIS und HOFMANN 2002] kommen die inhaltlichen Eigenschaften wie sie im Hybridsystem in Kapitel 6 zusätzlich betrachtet werden als Eigenschaftsvektoren zusammen mit ihren probabilistischen Klassifikationen, die als Eingabe der *SVM*s dienen, zum Einsatz. Da dieser Ansatz eine Generalisierung der Situation bei Empfehlungssystemen darstellt, hätte man ihn auch alternativ zum Hybridsystem in Kapitel 6 einsetzen können, aufgrund der Komplexität von *SVM*s (z.B. Wahl des richtigen Kernels, siehe [SCHÖLKOPF und SMOLA 2002]) hätte dies jedoch den Rahmen dieser Arbeit gesprengt.

[34] Siehe Kapitel 5.

3 Qualitätsbewertung von Empfehlungssystemen

„Evaluation, Mr. Spock.“ - “Fascinating!“

Capt. James T. Kirk & Cmdr. Spock,
Star Trek: The Motion Picture, 1979

Die Qualitätsbewertung von Empfehlungssystemen ist von zentraler Bedeutung für die Forschung und Entwicklung in diesem Bereich, denn neu entwickelte Verfahren bzw. Hypothesen zur Verbesserung der Leistung von bestehenden Empfehlungsverfahren müssen bzgl. ihrer Wirksamkeit und Gültigkeit in der Praxis überprüft werden. Dabei steht meist die Verwendung hypothetischer und bewährter Bewertungsmaße im Vordergrund. Diese Maße zur Qualitätsbewertung eines Verfahrens unter „Laborbedingungen“ dienen dazu, vor einer genaueren und zeitaufwändigen Untersuchung der Qualität des Verfahrens in einem realen Kontext dessen grundsätzliche Wirksamkeit einzuschätzen. Oft stehen auch nicht genug Zeit und andere Ressourcen zur Verfügung, um das Verfahren überhaupt in einer praktischen Anwendungsumgebung zu überprüfen, so dass man sich auf die Laborversuche beschränkt. Dies scheint auch insofern legitim zu sein, da in den Anfangstagen automatisierter Empfehlungssysteme bereits umfangreiche praktische Untersuchungen durchgeführt wurden, die die grundlegende Bewährtheit dieser Systeme im täglichen Einsatz gezeigt haben.
Diese Denkweise führt jedoch zu zweierlei Problemen - die Vielfalt heutiger Empfehlungssysteme erschwert aussagekräftige Vergleiche und die rein hypothetische Bewertung der Systeme ist nicht ausreichend. Erschöpften sich die Aufgaben von Empfehlungssystemen bei ihrer Einführung mit der Vorhersage einer Bewertung für ein bestimmtes Zielobjekt und einen bestimmten Zielnutzer bzw. das Generieren einer Liste von Empfehlungen für den Zielnutzer, so sind die an Empfehlungsverfahren gestellten Aufgaben mit der Zeit immer vielfältiger geworden. Unterschiedliche Aufgaben stellen jedoch auch unterschiedliche Anforderungen an ein System, so dass nicht automatisch die Standardmaße zur Evaluierung eingesetzt werden können, wie es noch vor 10 Jahren möglich war. Soll z.B. die Verbesserung der Qualität ein und desselben Systems untersucht werden, wobei dieses System eine personalisierte Sequenz von Musikstücken für eine Nutzerin bezogen auf ihre augenblickliche Stimmung erstellt, so reicht es nicht aus, einfach den durchschnittlichen absoluten Fehler der vorhergesagten Bewertungen für die einzelnen Stücke der Sequenz zu berechnen. Es muss vielmehr die Sequenz als Ganzes bewertet werden, da die Bewertungen einzelner Stücke vom Empfehlungssystem zwar richtig vorhergesagt worden sein können, diese Stücke jedoch möglicherweise eine Stimmung erzeugen, die zu der der übrigen Stücke nicht passt.
Noch schwieriger wird es, wenn nicht Verbesserungen eines Systems evaluiert werden sollen, sondern es zum Vergleich verschiedener Systeme kommt. Diese können sich nicht nur bzgl. der von ihnen erfüllten Aufgaben voneinander unterscheiden. Fast alle Systeme benötigen zur Erstellung eines Nutzerprofils, welches für die Vorhersage unerlässlich ist, auch Daten verschiedenen Typs[1] als Grundlage, die sich erheblich voneinander unterscheiden können. Ein kollaboratives Empfehlungsverfahren mit einer Datenbasis, die viele Nutzer aber wenig Objekte enthält, wird grundlegend andere Ergebnisse liefern als ein gleichartiges System mit wenig Nutzern aber vielen Objekten,[2] so dass hier das verwendete Maß zum Qualitätsvergleich genau abgewogen werden muss. Zudem beginnen Forscher, die sich zunächst auf Empfehlungssysteme eines bestimmten Typs (z.B. auf kollaborative Verfahren) konzentriert haben zunehmend damit, sich auch völlig anderen Systemtypen gegenüber (wie bspw. eigenschaftsbasierten Systemen) zu öffnen. Daher sind

[1] Wie sie in Abschnitt 2.3 beschrieben wurden.
[2] Weil sich die Datendichte unterscheidet.

Vergleiche solch unterschiedlicher Systeme heutzutage fast an der Tagesordnung. Diese Entwicklung mündet zudem in Hybridsystemen, in denen die verschiedenen Einzelverfahren miteinander verschmolzen werden. Da solch unterschiedliche Systeme Nutzerprofile meist auch auf völlig verschiedene Weise erzeugen, verschärft sich dort der Einfluss verschiedener Datenbasen auf den Qualitätsvergleich noch erheblich.
Schließlich führt auch die Beschränkung auf rein hypothetische Bewertungsmaße zu Problemen, wie dies schon in Abschnitt 1.1 ausgeführt wurde. Forscher wie [HERLOCKER et al. 2004] und andere weisen nicht umsonst darauf hin, dass alle hypothetischen Qualitätsmaße letzten Endes Details sind und die wirkliche Qualität eines Empfehlungssystems vor allem durch die Zufriedenheit der Nutzer bestimmt wird (insbesondere bei kommerziellen Systemen, wo die Verkaufszahlen ein Maß für diese Zufriedenheit sind). Daher müssen neue Evaluierungsmaße entwickelt werden und zum Einsatz kommen, die die Qualität auf alternative und vor allem praxisbezogene Weise erfassen. Die Diskussion solcher auf reale Anwendungen zugeschnittene Maße dient dabei als Grundlage für die Entwicklung des Maßes der „Interessantheit“ in Abschnitt 3.3, welches eine zentrale Rolle in dieser Arbeit spielt.
Diese Entwicklung vom Allgemeinen und Theoretischen zum Speziellen und Praktischen soll dabei durch den Verlauf dieses Kapitels widergespiegelt werden: Zunächst werden allgemeine Anforderungen aufgezählt, wie sie heutzutage an Empfehlungssysteme gestellt werden. Die Vielfalt dieser Anforderungen zeigt auf, wie komplex der Begriff der Qualität eines Empfehlungssystems ist. Obwohl das Hauptthema dieser Arbeit die Bewertung der Qualität bezogen auf die generierten Empfehlungen ist, so gibt es doch zahlreiche andere Qualitätskriterien, die sich teilweise sogar gegenseitig ausschließen. Schon die Gewichtung der Implementierung eines Systems auf die Erfüllung einer bestimmten Anforderung hin muss bei der Bewertung der Gesamtqualität berücksichtigt werden. Auch kann es Anforderungen geben, die nicht (vollständig) erfüllbar sind, da Empfehlungssysteme bestimmte allgemeine Schwächen mit sich bringen, die die maximal erreichbare Qualität einschränken. Diese Schwächen sollen zusammen mit den Stärken ebenfalls kurz aufgezeigt werden. Die Eigenschaften kollaborativer und eigenschaftsorientierter Empfehlungsverfahren werden dabei noch einmal gesondert betrachtet, da in dieser Arbeit ein kollaboratives Verfahren in Kapitel 5 und ein inhaltsorientierter Aufsatz für das kollaborative System in Kapitel 6 Verwendung finden.
Nach diesen allgemeinen Grundlagen folgt der Hauptteil des Kapitels, welcher der Qualitätsbewertung von Empfehlungssystemen und deren Abhängigkeit von bereits angesprochenen Faktoren wie den zu erfüllenden Aufgaben und der Art der zugrunde liegenden Daten gewidmet ist. Auf diesen unterschiedlichen Ausgangssituationen basierende Maße zur hypothetischen Qualitätsbewertung werden vorgestellt, anschließend wird auf alternative praxisbezogene Evaluierungsmaße eingegangen. Das Kapitel schließt mit der Herleitung des Interessantheitsmaßes ab, das in dieser Arbeit eine zentrale Rolle spielt.

3.1 Bewertungskriterien

Auch wenn die Beurteilung der Qualität der von einem Empfehlungssystem generierten Empfehlungen von entscheidender Bedeutung ist, so gibt es doch noch andere Bewertungsdimensionen, deren Bedeutung oft unterschätzt wird. Dies zeigt sich auch im Rahmen der praktischen Versuche dieser Arbeit. Daher sollen auch diese anderen Qualitätskriterien angeführt werden. Dabei wird schnell klar, dass nicht alle Kriterien zu hundert Prozent gleichzeitig erfüllbar sind, sondern dass es zwischen einzelnen Kriterien einen „trade off“ gibt. Bemüht man sich ein Kriterium zu erfüllen, so geht dies nur auf Kosten eines anderen Kriteriums und umgekehrt. Trotzdem sollte man bei dem Design eines Empfehlungssystems stets alle Qualitätskriterien im Hinterkopf behalten, denn nur ein in allen Bereichen ausgewogenes System wird von den Menschen (für die es letztlich entwickelt wurde) auch angenommen werden.

- *Zuverlässigkeit*: Eine Nutzerin soll sich darauf verlassen können, dass die vom Empfehlungssystem generierten Empfehlungen inkl. der vorhergesagten Bewertungen korrekt und vollständig sind. Im Einzelnen geht es dabei um folgende Punkte:
  - *Genauigkeit*: Die durch das Empfehlungssystem vorhergesagten Bewertungen sollen den tatsächlichen Bewertungen einer Nutzerin möglichst nahe kommen. Dieses Qualitätsmaß für Empfehlungssysteme ist mit Abstand am intensivsten untersucht worden. Einen Überblick über mög-

liche Maße für dieses Kriterium gibt Abschnitt 3.2. Eigene Erfahrungen sind in den Abschnitten 5.2 und 6.2 zu finden.

  - *Vollständigkeit*: Es soll für möglichst alle vorhandenen Objekte der Datenbasis eine Vorhersage generiert werden können. Ein hierfür geeignetes Qualitätsmaß namens *Abdeckung* (*Coverage*) wird in Abschnitt 3.2 definiert.
  - *Robustheit gegenüber Fehlern*: Unvollständige bzw. fehlerhafte Daten (Rauschen) können zu Problemen bei der Generierung von Empfehlungen führen. Das Empfehlungssystem sollte diesem Problem gegenüber wenig anfällig sein.

- *Nützlichkeit:* Die für einen bestimmten Nutzer des Empfehlungssystems generierten Empfehlungen sollen für ihn persönlich einen hohen Nutzen haben. Um diesen Nutzen zu messen, gibt es verschiedene, vorwiegend neue Maße, die in Abschnitt 3.2 vorgestellt werden. In dieser Arbeit wird zu diesem Zweck das Maß der *Interessantheit* definiert und benutzt (siehe Abschnitt 3.3). Die Erfahrungen mit diesem Qualitätsmaß lassen sich in den Abschnitten 5.2 und 6.2 einsehen.

- *Hohe Geschwindigkeit*: Die Empfehlungen sollen schnell generiert werden. Bei einer großen zu verarbeitenden Datenmenge kann die Erfüllung dieses Kriteriums zum Problem werden. Dies zeigen auch die eigenen Erfahrungen in den Abschnitten 5.1.2 und 6.1.2.

- *Geringer Ressourcenverbrauch*: Die Ressourcen des Rechnersystems, auf dem das jeweilige Empfehlungssystem läuft, sollen möglichst wenig beansprucht werden. Mögliche Ressourcen sind dabei CPU-Zeit, Haupt- und Festplattenspeicher, Anzahl von Zugriffen auf Ein- und Ausgabegeräte oder Belastung des Netzwerks. Ist das Empfehlungssystem webbasiert, so sind die genannten Ressourcen - bis auf die letztgenannte - für eine Nutzerin weniger wichtig als für den Administrator des Systems. Für die Nutzerin wiederum ist dann mehr die Aufbaugeschwindigkeit der Webseite interessant. Zwischen Ressourcenverbrauch und Geschwindigkeit gibt es meist einen trade off, wie auch die Abschnitte 5.1.2 und 6.1.2 zeigen.

- *Skalierbarkeit*: Sind Geschwindigkeit und Ressourcenverbrauch zufriedenstellend, so heißt dies nicht, dass dieser Zustand immer bestehen bleibt. Vielmehr sollen Geschwindigkeit und Ressourcenverbrauch des Empfehlungssystems auch durch eine starke Erweiterung der Menge der zu empfehlenden Objekte (und bei kollaborativen und demographischen Verfahren der Menge der Nutzer) nicht zu stark negativ beeinflusst werden. Da bei den meisten Empfehlungssystemen ein ständiges Wachstum der Datenmenge auftritt, ist diese Anforderung für die Zukunftsfähigkeit eines Systems unabdingbar. Wie wichtig dieses Problem im praktischen Einsatz von Empfehlungssystemen ist, wird in den Abschnitten 5.1.2 und 6.1.2 deutlich werden.

- *Benutzerfreundlichkeit*: Damit Empfehlungssysteme überhaupt benutzt werden, müssen sie anwenderfreundlich implementiert sein. Erfahrungen mit vielen der nachfolgend genannten Kriterien wurden in den praktischen Versuchen dieser Arbeit gewonnen und werden in den Kapiteln 5.2, 6.2 und 7 angesprochen. Die Benutzerfreundlichkeit wird durch verschiedene Faktoren bestimmt:

  - *Universelle Lauffähigkeit*: Das Empfehlungssystem sollte auf möglichst vielen Rechnern und Betriebssystemen einsetzbar sein bzw. in entsprechend vielen Versionen für unterschiedliche Konfigurationen vorliegen. Webbasierte Empfehlungssysteme, d.h. Systeme für die man nur einen Internetanschluss und einen Webbrowser benötigt, sind hierfür ideal, zudem die meisten Menschen heutzutage einen Zugang zum Internet besitzen (vgl. Kapitel 1).
  - *Einfache Installation*: Die Installation des Empfehlungssystems sollte möglichst einfach gehalten sein, da potenzielle Nutzer ansonsten abgeschreckt werden. Auch hier ist ein webbasiertes System ideal, da meist keine zusätzliche Installation nötig ist. Zudem sollten möglichst wenige Installationsvoraussetzungen bestehen. Wie wichtig dieses Kriterium sein kann, zeigen die eigenen Erfahrungen in den Kapiteln 5.2, 6.2 und 7.

  - *Gutes Interface*: Das Interface des Empfehlungssystems sollte intuitiv zu bedienen sein, insbesondere wichtig für Nutzer mit geringer Erfahrung im Computerbereich. Weiterhin sollten die präsentierten Informationen übersichtlich sein und nach verschiedenen Kriterien (alphabetisch, nach Bewertungen, bei Büchern nach Genre und Autor) geordnet werden können.
  - *Unterstützende Daten*: Diese wurden schon bei der Klassifikation in Abschnitt 2.3 unter dem Kriterium *Interface → Zusatzinformationen* erwähnt.
  - *Einfacher Bewertungsprozess*: Die Bewertungen für Objekte sollten auf möglichst einfache Art und Weise abgegeben werden können. Ideal ist eine Einbindung in bestehende Arbeitsabläufe, z.B. bei Newsreadern, wo der Anzeige einer bestimmten Nachricht ein oder mehrere Bewertungsknöpfe hinzugefügt werden können. Dem Nutzer kann durch Verwendung impliziter Bewertungen auch gänzlich die Bewertungsarbeit erspart werden.
  - *schnelle Einsatzfähigkeit*: Viele Empfehlungssysteme müssen erst eine gewisse Anzahl von Daten (Bewertungen von bekannten Objekten) erheben, bevor die ersten Empfehlungen generiert werden können. Diese „Anlaufphase“ sollte möglichst kurz gehalten werden (z.B. durch ersatzweise Verwendung von sinnvollen Defaultwerten).

- *Datenschutz:* Der Grad, in dem sensible Daten geschützt und die Möglichkeit der Anonymisierung geboten wird muss beurteilt werden.

Wie bereits erwähnt, gibt es zwischen vielen der aufgezählten allgemeinen Qualitätskriterien trade offs. Daher kann es vorkommen, dass ein Qualitätskriterium, das mit den generierten Empfehlungen an sich auf den ersten Blick scheinbar nichts zu tun hat, die Qualität dieser Empfehlungen dennoch beeinflusst. Ist ein System z.B. im Bezug auf die Skalierbarkeit optimiert, so können Maßnahmen zum Erreichen dieser Skalierbarkeit, wie bspw. Datensampling die Qualität von Empfehlungen verschlechtern (die „Instanzenselektion“ des kollaborativen Algorithmus aus Kapitel 5 widmet sich diesem Problem). Benötigt ein Algorithmus persönliche Daten, wie ein demographisches Empfehlungsverfahren, so kann eine Beschränkung dieser Daten aus Datenschutzgründen die Arbeitsweise des Systems negativ beeinflussen. Insofern ist vor der Qualitätsbeurteilung eines Empfehlungssystems bzgl. der generierten Empfehlungen die Möglichkeit der Beschränkung dieser Qualität durch solche „äußeren“ Faktoren im Gedächtnis zu behalten. Eine weitere Beeinflussung der Empfehlungsqualität ist wiederum durch die Wahl des Verfahrens gegeben, nach dem die Empfehlungen generiert werden. Die verschiedenen möglichen Typen von Empfehlungsverfahren, die dabei zur Wahl stehen, wurden in Kapitel 2.3 vorgestellt. Jedes dieser Verfahren hat seine spezifischen Stärken und Schwächen, die sich ebenfalls positiv oder negativ auf die Qualität der generierten Empfehlungen auswirken können. Hier sollen nun aus anfangs erwähnten Gründen die speziellen Vor- und Nachteile von kollaborativen und eigenschaftsbasierten Systemen genannt werden.
Jeder Vor- und Nachteil enthält dabei am Anfang in Klammern das Kürzel *K*, wenn der Vor-/Nachteil das kollaborative System betrifft und entsprechend das Kürzel *E* für eigenschaftsbasierte Systeme.

Vor- und Nachteile von kollaborativen und eigenschaftsbasierten Empfehlungssystemen

Kollaborative Empfehlungssysteme haben vor allem deshalb einen solchen Bekanntheits- und Beliebtheitsgrad erreicht, weil sie ein grundlegendes Problem der eigenschaftsorientierten Verfahren beheben - bezogen auf die Eigenschaften der bewerteten Objekte können Nutzern auch solche Objekte empfohlen werden, die bzgl. der Eigenschaften keine oder nur wenig Ähnlichkeiten mit den zuvor als gut bewerteten Objekten aufweisen. Aber auch die inhaltsbasierten Systeme haben ihre speziellen Vorteile.

Vorteile

- *Analyse der Objekte übernimmt der Nutzer (K).* Dadurch, dass kollaborative Verfahren lediglich die Bewertungen von Nutzern verarbeiten und nicht die Eigenschaften von Objekten, kann die schwierigere Analyse der Objekte an sich, die der Bewertung vorausgeht, dem Nutzer überlassen werden. Die sich daraus ergebenden Vorteile sind im Einzelnen:

  - *„Cross-Genre“/“Outside the Box“-Fähigkeit*: Hiermit wird der bereits erwähnte Hauptvorteil kollaborativer Systeme gekennzeichnet. Außergewöhnliche Empfehlungen können „den Horizont der Nutzerin erweitern“ und haben somit einen ganz besonders großen Nutzen.
  - *Relevanzbeurteilung vereinfacht*: Menschen können die Relevanz von Objekten besser beurteilen als Computer. Sie haben, wenn es bspw. um Texte oder Beschreibungen von Objekten geht, gegenüber inhaltsorientierten Verfahren einen klaren Vorteil, da für sie Besonderheiten der Sprache kein Problem darstellen, während Computer unter anderem Schwierigkeiten mit Synonymen oder Polysemen haben (siehe [RESNICK et al. 1994]).
  - *Beurteilung verschiedener Dimensionen möglich*: Menschen können im Gegensatz zu Computern auch andere Dimensionen eines Objekts beurteilen, wie z.B. Qualität statt der Quantität (vgl. [RESNICK et al. 1994]) und die gleichzeitige Wertung vieler verschiedener Dimensionen in eine einzige Bewertung einfließen lassen.

- *Kein Domänenwissen nötig (K,E)*: Empfehlungsverfahren, die nützlichkeits- oder wissensbasiert arbeiten benötigen umfangreiches Wissen über die Domäne, aus der die Objekte stammen, um Empfehlungen generieren zu können. Kollaborative Verfahren orientieren sich nur an den Bewertungen von Nutzern, während eigenschaftsbasierte Verfahren nur die Eigenschaften von Objekten vergleichen. Für beide Verfahren ist zusätzliches Wissen über die Domäne nicht nötig.

- *Implizite Bewertungen ausreichend (K,E)*: Im Gegensatz zu wissens- oder nützlichkeitsbasierten Empfehlungssystemen, denen die Vorlieben eines Nutzers explizit mitgeteilt werden müssen, damit Empfehlungen generiert werden können, sind implizite Bewertungen für kollaborative und eigenschaftsbasierte Verfahren völlig ausreichend.

- *Zeit verbessert Qualität (K,E)*: Je länger ein Empfehlungssystem von einer Person genutzt wird, desto mehr Bewertungen wird diese Person abgegeben haben. Somit stehen mit zunehmender Zeit immer mehr Informationen zur Verfügung, mit denen immer besser Nachbarn (kollaborativ: Nutzer, eigenschaftsbasiert: Objekte) mit ähnlichen Präferenzen (kollaborativ) bzw. Eigenschaften (eigenschaftsbasiert) ermittelt werden können, womit wiederum die Genauigkeit der generierten Vorhersagen steigt.

Nachteile

- *Fehlende Daten (K,E)*: Eigenschaftsorientierte Verfahren und kollaborative Verfahren haben beide unter fehlenden Daten zu leiden, wobei allerdings die kollaborativen Verfahren anfälliger sind. Das sog. *cold-start*-Problem, manchmal auch *ramp-up*- oder *startup*-Problem genannt, tritt auf, wenn ein neuer Nutzer oder ein neues Objekt einem Empfehlungssystem hinzugefügt werden. In diesem Fall sind anfangs zuwenig Daten für den Nutzer bzw. das Objekt vorhanden, um akkurate Empfehlungen zu generieren. Der Extremfall des *cold-start*-Problems liegt beim Erststart eines Empfehlungssystems vor, wenn noch gar keine Daten gesammelt wurden. Aber auch eine mangelnde Dichte der Daten kann Probleme verursachen. Dies wird nun genauer ausgeführt:

  - *Neuer Nutzer (K,E)*: Kollaborative und eigenschaftsbasierte Systeme müssen neue Nutzer dazu auffordern, eine Anzahl von bekannten Objekten zu bewerten, damit Daten für einen Vergleich mit anderen Nutzern/Objekten gegeben sind. Je weniger Objekte ein Nutzer bewertet hat, desto ungenauer werden die Empfehlungen sein, die das System für ihn erstellt. Viele Nutzer scheuen diesen Anfangsaufwand, womit sich ein Teufelskreis ergibt: Ohne partizipierende Nutzer gibt es wenig Bewertungen, aus denen akkurate Empfehlungen generiert werden können. Diese mangelnde Qualität wiederum lässt mehr Nutzer das System meiden. Darum benutzen viele kollaborative Systeme (wie auch das in dieser Arbeit eingesetzte, siehe Kapitel 4) bereits in anderen Systemen gesammelte Bewertungen. Dies wird auch als *seeding* bezeichnet. Für eigenschaftsbasierte Systeme besteht diese Möglichkeit nicht.

  - *Neues Objekt (K)*: Wird ein neues Objekt der Datenbank hinzugefügt, indem ein Nutzer es zum ersten Mal bewertet, ergeben sich daraus für kollaborative Verfahren zwei Probleme: Fordern andere Nutzer eine Vorhersage für dieses neue Objekt an (wenn z.B. eine Rangliste von Empfehlungen für alle vom Nutzer unbewerteten Objekte in der Datenbank erstellt werden soll), kann nur auf die Bewertungen der Person zugegriffen werden, die das Objekt neu bewertet hat. Wurde das Objekt mit einem „Extremwert“ (sehr gut oder sehr schlecht) bewertet, ergibt sich eine irreführende Vorhersage, weil sich die Vorhersage an diesem Extremwert orientiert. Dieses Problem besteht, bis eine angemessene Anzahl von Nutzern das Objekt bewertet haben (was bei Nischenfilmen oft nicht der Fall ist, vgl. auch praktische Erfahrungen in Kapitel 5.2.). Für eigenschaftsbasierte Systeme hingegen sind die Bewertungen anderer Nutzer uninteressant, da nur die Eigenschaften von Objekten verglichen werden.
    Ein weiteres Problem ergibt sich für den Nutzer kollaborativer Verfahren, der das neue Objekt zum ersten Mal bewertet hat, da ihm diese Bewertung keinen direkten Vorteil bringt. Durch das Bewerten dieses neuen Objekts wird die Qualität der für ihn generierten Empfehlungen nicht verbessert, da es keine anderen Nutzer gibt, die das Objekt bewertet haben und zu denen somit eine Ähnlichkeit der Präferenzen hergestellt werden könnte.[3] Dieses Problem wird auch als *early user*-Problem bezeichnet. Hier müssen Nutzer entsprechend motiviert werden. Nutzer eigenschaftsbasierter Systeme hingegen erwerben mit jedem zusätzlich bewerteten Objekt einen Vorteil, da jedes Objekt mehr Eigenschaften mit sich bringt, die mit den Eigenschaften anderer Objekte verglichen werden können.
  - *Mangelnde Datendichte (K)*: Wenn die Datendichte gering ist, also nur wenige Nutzer dieselben Objekte bewertet haben (sog. *sparsity*-Problem), können auch keine Nutzer mit ähnlichem Geschmack gefunden werden, unabhängig davon, wieviele Bewertungen insgesamt von Nutzern abgegeben wurden. Ein ähnliches Problem ergibt sich für Nutzer mit einem „außergewöhnlichen Geschmack“, die viele Objekte bewertet haben, die von anderen Nutzern kaum bewertet wurden. Insgesamt gesehen eignen sich kollaborative Verfahren am besten für eine relativ kleine statische Menge von Objekten, für die viele Nutzer Bewertungen abgegeben haben.
    Eigenschaftsbasierte Systeme sind auf eine besondere Datendichte nicht angewiesen, für sie ist lediglich die Anzahl der insgesamt vorhandenen Objekte in der Datenbasis interessant.

- *„Gray Sheep“-Problem (K)*: Durch den Vergleich von Nutzerpräferenzen betreiben kollaborative Verfahren im Prinzip eine Art *Clustering*, bei dem Nutzer mit ähnlichem Geschmack in denselben Cluster eingeordnet werden. Probleme ergeben sich für Nutzer mit einem uneinheitlichen Geschmack, die nicht eindeutig einem bestimmten Cluster zugeordnet werden können. Vorhersagen für solche Nutzer sind meistens von schlechter Qualität.
  Eigenschaftsbasierte Verfahren sind diesem Problem gegenüber robust, da nur Eigenschaften von Objekten verglichen werden. Die von einem Nutzer bewerteten Objekte können dabei aus den unterschiedlichsten Bereichen der Domäne stammen.

- *Stabilität vs. Plastizität-Problem (K,E)*: Ist die Präferenz eines Nutzers durch seine Bewertungen erst einmal ermittelt, ist es schwierig diese Präferenz wieder zu ändern. Handelt es sich bei den empfohlenen Objekten bspw. um Restaurants und ein Nutzer beschließt von einem Tag auf den anderen, Vegetarier zu werden, so wird es einige Zeit dauern, bis er auch vegetarische Restaurants empfohlen bekommt, da einzelne gute Bewertungen für vegetarische bzw. schlechte für nicht-vegetarische Restaurants ihr Nutzerprofil nicht verändern. Erst wenn eine gewisse Anzahl von Bewertungen abgegeben wurde, die dem neuen Geschmack entsprechen, kann das System entsprechend reagieren. Dies ist ein Problem, dass kollaborative und eigenschaftsbasierte Systeme teilen.

- *Übersehen von Nachbarn (K)*: Durch Vernachlässigung der Eigenschaften eines Objekts durch kollaborative Verfahren können einige Nachbarn nicht gefunden werden, wenn sie nicht dieselben Objekte bewertet haben. Sind z.B. zwei Nutzer Fans von Filmen eines bestimmten Regisseurs, aber der eine

[3] Siehe [AVERY und ZECKHAUSER 1997].

Nutzer bewertet nur solche Filme dieses Regisseurs, die dieser im Genre "Komödie" gedreht hat, während der andere Nutzer nur solche Filme desselben Regisseurs aus dem Genre "Action" bewertet, so können die Nutzer nicht als solche mit ähnlichem Geschmack identifiziert werden, obwohl es inhaltliche Gemeinsamkeiten gibt.

- *Portfolio-Effekt (E)*: Viele eigenschaftsbasierte Empfehlungssysteme filtern Objekte heraus, die bzgl. der bereits als gut bewerteten eine zu große Ähnlichkeit hinsichtlich der Eigenschaften aufweisen. Wenn diese Objekte jedoch etwas Neues beinhalten (z.B. wenn die Objekte Nachrichten im Bereich Newsgroups sind), dass für die Nutzerin wichtig wäre (neue Fakten oder Thesen), gehen sie als Empfehlungen verloren.

Nachdem die äußeren Faktoren genannt wurden, die die Qualitätsbewertung von Empfehlungen beeinflussen können, soll nun die Bewertung von generierten Empfehlungen intensiv dargestellt werden.

3.2 Qualitätsbewertung von Empfehlungen und deren Abhängigkeiten

Anfangs wurde erwähnt, dass die Aufgaben, die ein Empfehlungssystem aufgrund der bei seinem Design getroffenen Entscheidungen erfüllen soll, die Menge der möglichen Qualitätsmaße einschränken kann. Mittlerweile gibt es eine große Zahl solcher Aufgaben, die nun genannt werden sollen:

Aufgaben eines Empfehlungssystems

Waren es in den Anfängen der Empfehlungssysteme nur wenige Funktionen, die von den Systemen zur Verfügung gestellt wurden, so kamen im Laufe der Jahre immer mehr Anwendungsgebiete hinzu. Oft stellt man auch bei Empfehlungssystemen erst bei der praktischen Benutzung fest, welche Funktionen einem persönlich fehlen oder eine sinnvolle Erweiterung wären. Deshalb erhebt die folgende Liste auch keinen Anspruch auf Vollständigkeit und ist insbesondere auf Aufgaben aus der Sicht der Endnutzer beschränkt, da es in dieser Arbeit vor allem um diese Nutzer geht.

1. *"Annotation in Context" - Generierung von Vorhersagen unter Berücksichtigung von Reihenfolge und Kontext der Objekte*
   Eine der ersten Aufgaben für Empfehlungssysteme, wie sie von dem „Ursystem" *TAPESTRY* eingeführt und später von anderen Systemen wie *GroupLens* übernommen wurde. Voraussetzung für diese Aufgabe ist eine bereits vorhandene Struktur in der Gesamtmenge der zu bewertenden Objekte.
   Beispiel hierfür sind Beiträge in Newsgroups, in denen einzelne Nachrichten eine Diskussion in Form einer größeren Anzahl sich auf die Ursprungsnachricht beziehender Beiträge einleiten können. Ein weiteres Beispiel sind untereinander verlinkte Webseiten.
   Vorhersagen werden bezogen auf die Reihenfolge und den Kontext von Objekten generiert und dienen dazu, „schlechte" Objekte herauszufiltern oder dem Nutzer eine Entscheidungshilfe zu geben, welche Objekte im Kontext betrachtet werden sollten.
   Das Empfehlungssystem hat also vor allem die Aufgabe, erwünschte von unerwünschten Objekten zu trennen. Bezogen auf ein Bewertungsmaß heißt dies, dass, für möglichst viele Objekte eine Vorhersage berechenbar sein muss, um dieser Aufgabe befriedigend nachkommen zu können.

2. *Finden guter Objekte - Vorhersage der genauen Bewertung für ein einzelnes Objekt*
   Gilt als zweite der „ursprünglichen" Aufgaben eines Empfehlungssystems. Für einzelne Objekte und einen Zielnutzer soll dessen Bewertung für diese Objekte vorhergesagt werden. Dabei ist im Gegensatz zu *Annotation in Context* der Kontext unerheblich, vielmehr werden die Objekte voneinander unabhängig betrachtet. Es wird auf Anfrage entweder für ein bestimmtes Objekt eine Vorhersage berechnet und ausgegeben oder es wird eine Vorhersage für alle Objekte in der Datenbank generiert, so dass eine Rangliste der Objekte erstellt werden kann.

3. *Finden guter neuer Objekte - Benachrichtigung bei Auftreten neuer empfehlenswerter Objekte*
   Das Empfehlungssystem benachrichtigt die Nutzerin, sobald neue Objekte der Datenbank hinzugefügt werden, die für die Nutzerin empfehlenswert sind. Hierbei handelt es sich im Prinzip um eine Spezialform der vorherigen Aufgabe, die für kommerzielle Empfehlungssysteme eine Rolle spielt,[4] wo Produkte in die Datenbank eingepflegt werden, sobald sie auf dem Markt erscheinen (z.B. Neuerscheinungen von CDs oder Büchern). Die Nutzerin wird dann meist per E-Mail über die neuen Produkte informiert, die für sie interessant sein könnten.

4. *Finden aller guten Objekte - Auflisten aller empfehlenswerten Objekte unabhängig von der Größe der Datenbank*
   Diese Aufgabe ist in all den Bereichen gefragt, in denen sich eine hohe Falsch-Negativ-Rate äußerst ungünstig auswirkt, wie z.B. bei Rechtsanwälten, die nach Präzedenzfällen suchen. Hier steht vor allem die Betrachtung der Abdeckung aller interessanten Empfehlungen bezogen auf ein bestimmtes Profil im Mittelpunkt.

5. *Empfehlungssequenz - Auflisten einer Sequenz von Objekten, die empfehlenswert sind und bzgl. bestimmter Eigenschaften zueinander passen*
   Wurde bereits zu Anfang des Kapitels erwähnt: Es sollen nicht nur einzelne Objekte empfohlen werden die dem Nutzer gefallen, sondern eine Reihe von Objekten soll in ihrer Gesamtheit einen Nutzer ansprechen. Dies ist z.B. bei personalisierbaren Internetradios wichtig, wo eine Sequenz von Musikstücken Lieder enthalten kann, die einzeln betrachtet für den Nutzer hörenswert sind, ihm hintereinander gespielt jedoch nicht gefallen, da sie beispielsweise völlig unterschiedliche Stimmungen transportieren, der Nutzer jedoch eine Sequenz von Stücken sucht, die seiner augenblicklichen Stimmung entsprechen.

6. *Empfehlungen für Personengruppen - Auflisten von Objekten, die für alle Personen einer Gruppe gleichzeitig empfehlenswert sind*
   Eine von [Hill et al. 1995] im Rahmen der Untersuchungen des *BELLCORE*-Systems angeregte Aufgabe. Hier sollen Empfehlungen nicht nur für eine einzelne Person, sondern für eine Gruppe von Personen generiert werden, so dass die Empfehlungen jedem Mitglied der Gruppe gefallen. Ein Beispiel ist die Empfehlung von Filmen für Paare oder Freundeskreise, die zusammen einen Videoabend veranstalten wollen.

7. *Browsen - Anzeigen empfehlenswerter Objekte, die Ähnlichkeiten zum aktuell ausgewählten Objekt aufweisen*
   Viele Anwender nutzen Empfehlungssysteme zuweilen ohne Anspruch auf eine konkrete Empfehlung. Sie finden es unterhaltsam oder informativ, ihr Nutzerprofil zu verändern, um sich so vom System verschiedene Listen von Empfehlungen generieren zu lassen und diese miteinander zu vergleichen. Auch gibt es Systeme wie bspw. *Amazon* mit seiner „Kunden, die dieses Produkt gekauft haben, haben auch folgende Produkte gekauft“-Funktion, bei denen man durch Anklicken eines Objekts weitere Objekte aufgrund von Ähnlichkeiten oder statistischen Häufungen empfohlen bekommt. Diese Aufgabenstellung ist schon aus dem *Information Retrieval* bekannt, wo mittels Clustering-Verfahren zueinander ähnliche Objekte in einer Gruppe aggregiert werden können. Sogenannte „Scatter-Gather-Verfahren“ eignen sich dabei besonders gut zum Browsen, da sie bei jeder Iteration spezifischere Cluster erzeugen und somit das Browsen vor allem dazu benutzt wird, sich von einem groben Überblick zu interessanten Details vorzuarbeiten.
   Ein Empfehlungsalgorithmus, der mit dieser Aufgabe konfrontiert wird, sollte dabei weniger bzgl. der Genauigkeit seiner Vorhersagen bewertet werden. Vielmehr stehen hier das Interface, die intuitive Benutzung, die Qualität und Art der zu einem Objekt gelieferten Informationen und die Aufbereitung dieser Informationen im Vordergrund.

8. *Finden von vertrauenswürdigen Empfehlungssystemen - Generieren von Empfehlungen, die das Vertrauen von Nutzern in das System steigern*

[4] Siehe [Schafer et al. 1999].

Einige Nutzer sind sehr vorsichtig, was die Tatsache angeht, welchem Empfehlungssystem sie ihr Vertrauen schenken. Dahinter stehen die Befürchtungen, ein Empfehlungssystem könnte falsche Empfehlungen liefern und Nutzer somit zum Kauf von Produkten motivieren, die sich im Nachhinein als teure Fehlinvestitionen herausstellen. Aber auch die - aufgrund steigender Kommerzialisierung der Gesellschaft - nicht unbegründete Furcht, bestimmte Empfehlungen könnten nicht auf gesammelten Informationen, sondern finanziellen Interessen beruhen (Promotion von eigenen Produkten bzw. erfolgte Bezahlung/Bestechung, um gewisse Produkte gehäuft zu empfehlen[5]), treibt Nutzer dazu, Empfehlungssysteme vor dem „richtigen“ Einsatz umfangreich zu testen. Durch das Erstellen verschiedener Nutzerprofile können die Unterschiede und Gemeinsamkeiten der auf den Profilen basierenden Empfehlungen ergründet werden.
Viele Nutzer testen insbesondere, ob die Empfehlungen ihre Lieblingsprodukte oder die von ihnen als besonders schlecht empfundenen Produkte enthalten. Dies birgt jedoch die Gefahr, Systeme zu benachteiligen, die darauf ausgerichtet sind, Empfehlungen zu liefern, die zutreffend und gleichzeitig für den Nutzer überraschend sind.

9. *Verbesserung des eigenen Profils - Schaffung von Anreizen für Nutzerinnen, stetig weitere Bewertungen abzugeben*
   Da Empfehlungssysteme allgemein mit steigender Anzahl von abgegebenen Bewertungen bezogen auf einen Nutzer auch akkuratere Empfehlungen auf Basis dieses Nutzerprofils erzeugen, liegt die Vermutung nahe, dass Nutzer daran interessiert sind, möglichst viele Bewertungen abzugeben, um selbst davon zu profitieren.
   Trotzdem ist dies in der Praxis nicht immer der Fall. Deswegen kann es interessant sein zu untersuchen, ob und aus welchen Gründen Nutzer Bewertungen abgeben, da die Qualität der Vorhersagen und damit der Erfolg eines Empfehlungssystems von diesen Kriterien abhängt.

10. *Selbstausdruck - Bieten von Möglichkeiten zur Selbstdarstellung von Nutzern*
    Unter den Nutzern von Empfehlungssystemen gibt es auch solche, die durch eine hohe Anzahl von abgegebenen Bewertungen aus der allgemeinen Masse herausstechen. Solche „Power-User“ tätigen eine so große Anzahl von Bewertungen jedoch nicht, um das eigene Profil zu verbessern und somit genauere Empfehlungen zu erhalten, sondern um die eigene Person auszudrücken, wie eine Umfrage von [HERLOCKER et al. 2004] ergeben hat. Insbesondere Systeme wie *Amazon*, die nicht nur eine abstrakte Bewertung in numerischer Form, sondern zusätzlich die Möglichkeit angehängter Kommentare geben, sind für solch eine Selbstdarstellung geeignet. Für Empfehlungssysteme bedeutet dies, dass diese den Nutzern eine Möglichkeit bieten sollten, die eigene Persönlichkeit einzubringen. Auch der Grad der Anonymisierung sollte bewertet werden, denn während einige Nutzer gern „im Rampenlicht stehen“, ist für andere die Anonymität motivierend.
    Zieht ein Empfehlungssystem solche Power-User an, so ist dies ein Gewinn für die gesamte Gemeinschaft, da viele Bewertungen die Genauigkeit der Empfehlungen des Systems steigern.

11. *Hilfsbereitschaft - Abgeben von Bewertungen für das Allgemeinwohl*
    Eine Eigenschaft die oft mit Vertretern des vorigen Punktes einhergeht. Einige Nutzer geben Empfehlungen schlicht aus dem Glauben heraus ab, zum Wohl der Gemeinschaft des jeweiligen Empfehlungssystems beizutragen. Trotz allem Idealismus will auch dieses Verhalten motiviert sein. Dafür ist das Erzeugen eines Gemeinschaftsgefühls - das Schaffen einer „community“ - genauso nützlich, wie erweiterte Ausdrucksmöglichkeiten in Form von Kommentaren zu Bewertungen oder sogar Foren, in denen Hilfegesuche artikuliert werden können.

12. *Beeinflussung - Abgeben falscher Bewertungen, um eigene Produkte zu fördern*
    Ein negativer Aspekt im Zusammenhang mit Empfehlungssystemen, der häufig auftritt. Es kann Personen geben, die ein System für egoistische Zwecke missbrauchen. Insbesondere in kommerziellen Empfehlungssystemen sind es solche Personen, die gezielt Bewertungen abgeben, um den Absatz

[5] Vgl. Suchmaschinen, bei denen gegen Bezahlung bestimmte Einträge auf oberen Positionen in der Ergebnisliste platziert werden können.

eigener Produkte oder Produktgruppen zu fördern.
Da insbesondere bei kollaborativen Empfehlungsverfahren damit das ganze Grundkonzept ad absurdum geführt wird, sollte ein System auch danach bewertet werden, wie wirksam es solch einen Missbrauch zu verhindern vermag.
Es ist sicherlich auch eine interessante weiterführende Aufgabe für den Bereich des maschinellen Lernens, solche Personen automatisch zu erkennen und herauszufiltern.

Benutzte Datenbasis

Ist die Aufgabe die ein Empfehlungssystem erfüllen soll erst einmal festgelegt, benötigen alle automatisierten Systeme Daten um die Präferenzen eines Nutzers zu ermitteln. Erst auf Basis des aus diesen Daten erstellten Präferenzmodells können Objekte, die diesem Modell entsprechen empfohlen und somit die festgelegten Aufgaben erfüllt werden.
Da die für einen Empfehlungsalgorithmus gegebene Datenbasis somit als Eingabe des Algorithmus die Ausgabe in Form der Empfehlungen maßgeblich bestimmt, muss diese Datenbasis auch bei der Bewertung von Empfehlungssystemen genau betrachtet werden. Aus welchem Umfeld die Daten stammen ist für die Beurteilung der Qualität eines Empfehlungssystems ebenso von Bedeutung, wie die Art, auf die diese Daten gewonnen oder das Verfahren mittels dessen die Daten zum Präferenzmodell eines Nutzers weiterverarbeitet werden.

1. *Synthetische/Natürliche Daten - Nutzer, Objekte und Bewertungen werden selbst erzeugt/stammen aus einer realen Anwendung*
   Nicht immer kann bei der Qualitätsbeurteilung von Empfehlungsalgorithmen auf natürliche, d.h. unter realer Benutzung eines Systems entstandene Daten zurückgegriffen werden. Dies ist insbesondere dann der Fall, wenn ein neues Empfehlungssystem entwickelt wurde, das zur korrekten Arbeitsweise auf ein Minimum an Daten angewiesen ist. Kollaborative Verfahren sind besonders von dem „cold-start"-Problem betroffen, da sie nur dann Empfehlungen für einen Zielnutzer generieren können, wenn sie genug Bewertungsdaten anderer Nutzer vorliegen haben, die sie mit den Bewertungen des Zielnutzers vergleichen können. Aber auch andere Empfehlungsverfahren bringen ohne eine ausreichende Datenbasis die als Ausgangspunkt genutzt werden kann nur unbefriedigende Ergebnisse. Wird z.B. bei demographischen Algorithmen nicht auf vordefinierte Klassen von Nutzern mit ebenfalls vordefinierten Präferenzen zurückgegriffen, sondern Klassen von Nutzern vielmehr spontan bei Hinzufügen neuer Nutzer mittels Clustering gebildet bzw. aktualisiert, dann entsteht zu Beginn mit wenigen Nutzern eine Situation, in der nur wenige demographische Klassen gebildet werden können. Wenige vorhandene Klassen ermöglichen jedoch nur eine sehr grobe Unterscheidung zwischen verschiedenen Nutzern und somit äußerst unzureichende Empfehlungen. In diesen Fällen kann auf synthetisch generierte Daten zurückgegriffen werden um halbwegs sinnvolle Empfehlungen zu produzieren, bis genügend reale Daten gesammelt worden sind, die dann als Basis genutzt werden können und die synthetischen Daten ersetzen.
   Zuweilen sind reale Daten bei der Evaluierung von Empfehlungssystemen auch gar nicht erwünscht, bspw. wenn die Auswirkung einer ganz bestimmten Datenverteilung auf die Arbeitsweise eines Verfahrens untersucht werden soll. Mittels synthetischer Daten kann jede beliebige Situation simuliert werden, auch solche die in der Praxis so gut wie gar nicht anzutreffen sind. Auch können potenziell neue Empfehlungsverfahren auf diese Weise einem ersten Qualitätstest unterzogen werden, bevor man Zeit und Mühe in eine praktische Anwendung investiert.
   In jedem Fall muss die Verwendung synthetischer Daten bei der Qualitätsbeurteilung berücksichtigt werden, da synthetische Daten die Realität oft nur unzureichend widerspiegeln und somit das Messen einer guten Qualität für einen bestimmten Algorithmus auf synthetischen Daten nicht automatisch eine entsprechend gute Qualität auf realen Daten nach sich zieht. Auch beim Vergleich mehrerer Systeme auf denselben synthetischen Daten ist Vorsicht geboten, da die Daten zufällig oder bewusst so angelegt sein können, dass sie der Arbeitsweise eines bestimmten Empfehlungssystems entgegen kommen. So weisen z.B. [AGGARWAL et al. 1999] bei ihrem graphentheoretischen kollaborativen Ansatz für Empfehlungsverfahren explizit darauf hin, dass die von ihnen beschriebenen

Experimente und die daraus folgende Evaluierung ihres Verfahrens gegenüber anderen Algorithmen „unfair“ seien, da die von ihnen in den Experimenten verwendeten synthetischen Daten dem ihrem Ansatz zugrunde liegenden Modell entsprechen.

2. *Offline/Online Daten - Einsatz ehemaliger realer Nutzer-, Objekt- und Bewertungsdaten unter Laborbedingungen/Nutzung realer Daten zur Laufzeit des Systems*
   Nicht nur synthetische Daten können die Qualitätsbeurteilung von Empfehlungsverfahren „verzerren“, auch reale Daten müssen bzgl. des experimentellen Umfeldes in dem sie zum Einsatz kommen beurteilt werden. Dabei kann es sich bei dem Umfeld um eine reale Anwendungsumgebung mit Nutzern handeln oder um Daten, die zuvor in solch einem Online-Experiment gesammelt wurden und offline, also unter „Laborbedingungen“ zur Qualitätsbeurteilung benutzt werden. Ein Teil dieser Offline-Daten dient dann dazu, das jeweilige Empfehlungssystem die Nutzerpräferenzen erlernen zu lassen, während der andere Teil zu Test- und Vergleichszwecken herangezogen wird, wenn das Empfehlungssystem basierend auf den erlernten Präferenzen Empfehlungen generiert. Anhand der Anzahl und Genauigkeit von Übereinstimmungen zwischen den generierten Empfehlungen und den Daten der Vergleichsmenge kann dann die Qualität des Verfahrens beurteilt werden. Wurden beim Online-Experiment nicht nur die Aktionen eines Nutzers aufgezeichnet, sondern jede Aktion zusätzlich noch mit einem Datums- und Zeitstempel versehen, so kann der genaue zeitliche Ablauf offline nachvollzogen werden. Auch eine Kombination aus Offline- und Online-Tests, wie in dieser Arbeit durchgeführt, ist möglich.
   Solche Offline-Experimente basieren zwar im Gegensatz zu synthetischen Datensätzen auf realen Daten und bieten den großen Vorteil, dass eine Evaluierung unter verschiedenen Gesichtspunkten (wie z.B. eine Parameteroptimierung von Empfehlungsverfahren) automatisiert und somit schnell und einfach durchgeführt werden kann, jedoch bringen solche Offline-Experimente auch einen entscheidenden Nachteil mit sich: Ein Empfehlungsalgorithmus der auf Offline-Daten eine gute Vorhersagequalität liefert, kann in einer Online-Umgebung trotzdem eine wesentlich schlechtere Qualität aufweisen. Nützlichkeits- bzw. wissensbasierte Empfehlungsverfahren sind sensitiv gegenüber Änderungen im Geschmack eines Nutzers. Bei einer solchen Änderung wird das Präferenzmodell den neuen Gegebenheiten angepasst. Wird nun z.B. die Qualität solch eines Systems mittels Offline-Daten beurteilt und werden kurz nach dem Zeitpunkt der Extraktion dieser Offline-Daten online neue Daten in Form von Bewertungen des Nutzers gewonnen die zu einer massiven Veränderung der Nutzerpräferenz führen, so kann sich die Qualität des Empfehlungssystems auf den Offline-Daten und den aktuellen Online-Daten stark voneinander unterscheiden.
   Weiterhin kann durch Offline-Experimente nur die Qualität des Systems bezogen auf die Vorhersagegenauigkeit bestimmt werden, während Nutzer in Online-Experimenten auch andere Qualitätskriterien wie Einfachheit der Bedienung oder Ästhetik des Interfaces bewerten können. Auch praktische Bewertungsmaße wie Nützlichkeit oder die in dieser Arbeit verwendete Interessantheit von Empfehlungen kann nur über Online-Experimente ermittelt werden, die jedoch einen hohen Zeitaufwand mit sich bringen.

3. *Weitere wichtige Dateneigenschaften - Eigenschaften die sich aus dem Einfluss der Domäne, des spezifischen Empfehlungssystems und aufgrund statistischer Gegebenheiten ergeben*
   Neben den beiden zuvor genannten Eigenschaften von Daten für Empfehlungssysteme gibt es weitere Charakteristika, die die Qualitätsbewertung entscheidend beeinflussen können. Diese Eigenschaften werden von [HERLOCKER et al. 2004] in drei Kategorien aufgeteilt die den Einfluss der Art der zu empfehlenden Objekte und des Kontextes in dem sie auftreten (*Domäne*) auf die Daten widerspiegeln. Beeinflusst werden die Daten auch durch (*inhärente*) Eigenschaften des Empfehlungssystems und die statistischen Gegebenheiten auf den Daten selbst (*Eigenschaften des Datensamples*).

   a) *Eigenschaften der Domäne - Art der Objekte und Kontext, in dem das Empfehlungssystem eingesetzt wird:*

      i. *Art der empfohlenen/bewerteten Objekte und der Kontext, in dem empfohlen/bewertet wird*
         Beispiele für mögliche Objekte sind Filme, Musik, Usenet News oder wissenschaftliche

Publikationen. Bzgl. eines bestimmten Objekttyps kann es wiederum verschiedene Kontexte geben. Die Bewertung/Empfehlung kann z.B. online über eine Webseite erfolgen, im Fall von Filmen könnte ein entsprechendes System z.B. auch in Form eines Automaten in Videotheken anzutreffen sein.

ii. *Vom Empfehlungssystem erfüllte Aufgaben (wie zuvor aufgelistet)*
Dieser Punkt bezieht sich auf den vorherigen Abschnitt über die verschiedenen Aufgaben, die ein Nutzer an ein Empfehlungssystem stellen bzw. die dieses System anbieten kann. Je nach Kontext können die Aufgaben sehr unterschiedlich sein und die Art der gesammelten Daten beeinflussen. Für das einfache Finden guter Objekte müssen andere Daten gespeichert werden als für die Aufgabe „Annotation in Context“, wo es zu jedem Objekt Nutzerkommentare und Querverweise zwischen Objekten gibt.

iii. *Neue Empfehlungen vs. Qualität - Generieren unerwarteter oder bekannter Empfehlungen*
In vielen Kontexten geht es vorrangig darum, einem Nutzer Objekte zu empfehlen die für ihn völlig neu sind. Diese besondere Stärke von kollaborativen Empfehlungssystemen ist jedoch nicht immer gewünscht, da das Empfehlen von Objekten die außerhalb der bewussten Wahrnehmung des Nutzers liegen[6] stets mit einem gewissen Risiko verbunden ist. Insbesondere im kommerziellen Kontext kann es vorteilhafter sein, das Hauptaugenmerk auf eine hohe Vohersagequalität zu setzen, so dass dem Nutzer Objekte empfohlen werden, von denen bekannt ist, dass er sie bevorzugt, womit eine höhere Kaufwahrscheinlichkeit gegeben ist.

iv. *Kosten-Nutzen-Verhältnis von wahr/falsch positiven/negativen Vorhersagen*
Insbesondere kostenlose Empfehlungssysteme für Filme haben sich als besonders erfolgreich erwiesen, dies sicherlich auch deswegen, da der Nutzen richtiger Empfehlungen sehr hoch ist. War die Auswahl von Filmen in Videotheken wegen des großen Angebots oft schon schwierig genug, so haben die in letzter Zeit populär gewordenen DVD-Ausleihangebote über das Internet zu einer noch größeren Vielfalt des Angebots geführt. Die Kosten falscher Empfehlungen hingegen sind mit zwei Stunden verschenkter Zeit und je nach Art des Erwerbs (Ausleihen, Kaufen, im Kino ansehen) mehr oder weniger gering, insgesamt also gut zu verkraften. Im theoretisch denkbaren Rahmen von Empfehlungen für Aktienfonds oder Versicherungen könnten sich falsche Empfehlungen jedoch als sehr kostspielig herausstellen.

v. *Granularität der Nutzerpräferenzen - wie fein ist die innerliche Bewertungsskala des Nutzers?*
Empfehlungssysteme geben eine bestimmte Bewertungsskala vor, die wie im Fall von binären Bewertungen (empfehlenswert/nicht empfehlenswert) sehr grob, aber auch beliebig fein sein kann (Systeme mit einer hundertwertigen Skala sind bekannt). Diese Granularität von Bewertungen muss jedoch nicht mit der identisch sein, die ein Nutzer persönlich empfindet. Bei einer vielwertigen durch das Bewertungssystem vorgegebenen Skala beobachtet man oft eine Häufung von so genannten „Extremwerten“, die darauf hinweist, dass Nutzer innerlich einer wesentlich groberen persönlichen Skala folgen.[7]

vi. *Bewertungsfrequenz - die Häufigkeit, mit der Bewertungen abgegeben werden*
Je höher die Bewertungsfrequenz ist, desto mehr Daten stehen zur Generierung vom Empfehlungen zur Verfügung, was wiederum die Genauigkeit der Empfehlungen erhöht.

b) *Inhärente Eigenschaften - Auswirkungen der speziellen Eigenschaften eines benutzten Empfehlungssystems auf die Bewertungsdaten*

i. *Implizite/Explizite Bewertungen - bereits aus der Einleitung bekannt*
Während explizite Bewertungen eine klare Aussage des Nutzers über seine Meinung zu einem Objekt darstellen, werden implizite Bewertungen mittels einer mehr oder weniger

[6] Weil sie ihm völlig unbekannt sind.
[7] Vgl. auch die Untersuchungen von [McNee et al. 2002] oder die Versuchsergebnisse dieser Arbeit in Kapitel 5.2.

sinnvollen Methode aus den Aktionen des Nutzers gewonnen. Da die der benutzten Methodik zugrunde liegenden Annahmen und damit die mit dieser Methodik ermittelten impliziten Bewertungen falsch sein können, lassen sich Empfehlungssysteme mit impliziten und expliziten Bewertungen nur schwer miteinander vergleichen. Implizite Bewertungen haben jedoch den Vorteil, meist eine größere Datendichte (siehe folgende *Eigenschaften des Datensamples*) nach sich zu ziehen, da sie für einen Nutzer keinen erhöhten Arbeitsaufwand bedeuten.

ii. *Bewertungsskala - Intervall und Granularität der Bewertungsskala des Empfehlungssystems*
Die Anzahl der möglichen Bewertungen (Granularität) und das Intervall, aus denen sie stammen. Systeme, die Bewertungsskalen mit niedriger Granularität benutzen (z.B. binär) haben bei der Qualitätsbeurteilung Vorteile gegenüber Systemen mit höherer Granularität.

iii. *Bewertungsdimensionen - ebenfalls bereits bekanntes Kriterium*
Insbesondere nützlichkeits- und wissensbasierte Systeme arbeiten nach dem Prinzip mehrerer Bewertungsdimensionen. Für Musikstücke als Empfehlungsobjekte wären die Dimensionen Originalität, musikalisches Können und Aufnahmequalität denkbar.

iv. *Datums- und Zeitstempel - Speichern jeder Nutzeraktion inkl. der Zeit und des Datums ihres Auftretens*
Einige Systeme speichern für jede Nutzeraktion das aktuelle Datum und die aktuelle Zeit. Dies ist wichtig bei Systemen, wo über die Zeit gesehen Änderungen der Nutzerpräferenzen erwartet werden. Insbesondere im Kontext von Musikstücken findet man bei den meisten Menschen über einen längeren Zeitraum betrachtet Änderungen des Geschmacks. Zeit- und Datumsstempel können einem Empfehlungssystem beim Offline-Lernen und der Bewertung dieses Lernergebnisses zugute kommen, da die meisten Änderungen der Nutzerpräferenz langsam stattfinden. Hat ein Nutzer hingegen im Laufe der Zeit zwei entgegengesetzte Vorlieben entwickelt und ein System wird abwechselnd mit Objekten konfrontiert, die jeweils einer der Vorlieben entsprechen, so kann es je nach System zu wesentlich schlechteren Lernergebnissen kommen („Gray Sheep“-Problem bei kollaborativen Verfahren).

v. *Speichern generierter Empfehlungen - Speichern der vom System generierten Empfehlungen inkl. Zeit und Datum ihres Auftretens*
Dies stellt im Prinzip eine Erweiterung des vorherigen Punktes dar. Werden zusätzlich zu den Nutzeraktionen auch die Aktionen des Empfehlungssystems mit Zeit- und Datumsstempel aufgezeichnet, kann ermittelt werden, ob zu einem früheren Zeitpunkt vom System generierte Empfehlungen die nachfolgenden Bewertungen eines Nutzers beeinflusst haben. Ein Nutzer der bspw. gewalttätige Filme verachtet und mehrere solcher Filme empfohlen bekommt, wird unbewusst eine negative Einstellung dem System gegenüber entwickeln und somit wahrscheinlich nachfolgende Empfehlungen schlechter bewerten, wie es der Fall gewesen wäre, wenn diese ihm vor den Empfehlungen gewalttätiger Filme präsentiert worden wären.

vi. *Vorhandensein demographischer/objektbezogener Daten - Heranziehen zusätzlicher Daten über die Nutzerin oder die Eigenschaften der Objekte zur Empfehlungsgenerierung*
Teilweise aus Abschnitt 2.3 bekannt. Durch zusätzliche Verwendung demographischer und objektbezogener Daten entstehen im Prinzip Hybridsysteme. Allgemein bedeuten zusätzliche Daten bezogen auf die Qualität von Empfehlungen immer einen Vorteil.

vii. *Voreingenommenheit durch spezielle Datenkollektion - alle neuen Nutzer müssen initial dieselben Objekte bewerten*
Viele Empfehlungssysteme wie inhaltsorientierte oder kollaborative Verfahren benötigen von jedem Nutzer initial Bewertungen einiger Objekte, um dessen Präferenzen ermitteln zu können. Werden jedem Nutzer dieselben Objekte zur Bewertung präsentiert, so schafft man eine Situation, in der für diese Objekte im Vergleich zu anderen Objekten wesentlich

mehr Daten vorhanden sind. Bei inhaltsorientierten Verfahren würde dies dazu führen, dass dem Nutzer besonders solche Objekte empfohlen werden, deren Eigenschaften mit denen der initialen Objekte übereinstimmen.

c) *Eigenschaften des Datensamples - Anzahl der Datensätze, Dichte und Verteilung*

i. *Datendichte allgemein*
Gibt an, wie vollständig die vorhandenen Daten sind. Je nach Empfehlungsverfahren ist die Vollständigkeit verschiedener Datentypen wichtig. Kollaborative Verfahren sind z.B. auf eine hohe Dichte von Bewertungen angewiesen, um vernünftige Empfehlungen generieren zu können. Demographische Verfahren benötigen sowohl eine hohe Bewertungsdichte, als auch eine hohe Dichte von demographischen Daten. Bei inhalts-, nützlichkeits- und wissensbezogenen Systemen ist hingegen eine hohe Datendichte im Bezug auf Eigenschaftswerte von Objekten wichtig. Eine geringe Datendichte kann je nach Art des Empfehlungssystems künstlich sowohl durch Hinzufügen/Entfernen bestimmter Nutzer/Objekte/Bewertungen erreicht werden, durch Nutzung impliziter Bewertungen und auch durch Einsatz automatisierter Agenten (siehe [GOOD et al. 1999]), die bzgl. ihrer Aufmerksamkeit nicht wie Menschen auf bestimmte Teilbereiche des Datenraums beschränkt sind.

ii. *Datendichte nutzerbezogen*
Die Datendichte bezogen auf den Zielnutzer ist für alle Arten von Empfehlungssystemen entscheidend. Je größer die persönliche Datendichte ist, desto besser kann die Präferenz eines Nutzers bestimmt werden. Auch hier können die zuvor genannten Mittel zur Steigerung der Dichte zum Einsatz kommen.

iii. *Allgemeine Größe und Verteilung der Daten*
Je mehr Daten vorhanden sind, desto mehr Informationen stehen zur Verfügung, um Empfehlungen zu generieren. Dies muss jedoch stets im Zusammenhang mit der Verteilung dieser Daten gesehen werden. Eine ungleichmäßige Verteilung führt zu Datenlücken („sparsity“). Es gibt jedoch auch Systeme, die sich gerade eine ungleiche Verteilung der Daten zunutze machen oder diese sogar selbst erzeugen, indem Nutzer vor Gebrauch des Systems dieselben Objekte bewerten müssen. Die ungleiche Verteilung kann dann benutzt werden, um Korrelationen zwischen Nutzern/Objekten mit hoher Datendichte herzustellen und diese dann für Vorhersagen im Teilraum mit niedriger Datendichte zu verwenden.
Für einige Verfahren ist auch das Verhältnis zwischen der Anzahl von Nutzern und Objekten wichtig. Gibt es mehr Nutzer als Objekte, so ist für kollaborative Systeme eine Korrelationsberechnung zwischen Nutzern erfolgversprechender. Im entgegengesetzten Fall ist es sinnvoller, die Korrelation zwischen Objekten zu ermitteln.

Wie die vorherigen Auflistungen zeigen, gibt es sehr viele Faktoren, die das Verhalten eines Empfehlungssystems und damit die Qualität der von ihm erstellten Empfehlungen beeinflussen. Noch zahlreicher werden die Einflüsse, wenn man sich vor Augen hält, dass z.B. die gewählten Aufgaben eines Empfehlungssystems durch die verwendete Datenbasis beeinflusst werden und umgekehrt. Daher muss zur Beurteilung der Qualität der generierten Empfehlungen ein den jeweiligen Einflüssen sorgfältig angepasstes Maß gewählt werden. Die größte Auswahl findet man dabei bei den traditionell zur Qualitätsbewertung benutzten hypothetischen Maßen, den Maßen zur Bestimmung der Genauigkeit eines Systems.

Maße für die Genauigkeit (*accuracy*) eines Systems (hypothetische Genauigkeit)

1. *Maße für die Vorhersagegenauigkeit - Differenz zwischen vorhergesagten und tatsächlichen Objektbewertungen*
Diese häufig genutzten Maße zeigen an, wie nahe die durch ein Empfehlungssystem vorhergesagte Bewertung für einen bestimmten Zielnutzer und ein Zielobjekt an der tatsächlich durch den Zielnutzer abgegebenen Bewertung für dieses Objekt liegt. Diese Maße können nur für Empfehlungssysteme angewendet werden, die genaue Bewertungen für Objekte anzeigen (wie bei der Aufgabe *Annotation*

in Context), statt nur eine Einteilung in empfehlenswerte und nicht empfehlenswerte Objekte vorzunehmen (wie die meisten eigenschaftsorientierten oder wissensbasierten Systeme). Dementsprechend ungeeignet sind diese Maße für die Bewertung von Empfehlungssystemen, die hauptsächlich auf Aufgaben wie das *Finden aller guten Objekte* hin konzipiert sind, da dort nicht der Fehler in der genauen Bewertung eines Objekts interessiert, sondern nur die Frage, ob empfehlenswerte Objekte auch tatsächlich am Anfang einer Rangliste erscheinen. Ranglisten können jedoch auch durch Systeme mit genauen Vorhersagewerten erstellt werden, indem für alle nicht bewerteten Objekte bezogen auf einen bestimmten Zielnutzer Bewertungsvorhersagen generiert und die Objekte anschließend bzgl. dieser Vorhersagewerte sortiert werden. In diesem Fall können wiederum Maße für die Vorhersagegenauigkeit Einsatz finden und sind sogar im Vorteil, da nicht die Genauigkeit des Empfehlungssystems bzgl. der Reihenfolge der empfohlenen Objekte, sondern auch die Vorhersagegenauigkeit für jeden einzelnen Rang gemessen werden kann.
Der mittlere absolute Fehler (*Mean Absolute Error*) ist dabei der wohl bekannteste Vertreter dieser Gruppe und eines der meist genutzten Maße insbesondere für kollaborative Empfehlungssysteme (siehe z.B. [BREESE et al. 1998], [HERLOCKER et al. 1999] oder [SHARDANAND und MAES 1995]). Der *MAE* misst die durchschnittliche absolute Abweichung zwischen einer durch das System vorhergesagten Nutzerbewertung p_i und der durch den Nutzer tatsächlich erfolgten Bewertung r_i für ein bestimmtes Objekt. Dabei gibt N die Anzahl der insgesamt vorgenommenen Vorhersagen (für verschiedene Objekte und/oder Nutzer) an:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |p_i - r_i| \tag{3.1}$$

Der *MAE* hat den großen Vorteil, dass er direkt interpretierbar ist und seine statistischen Eigenschaften genauestens untersucht sind, so dass Signifikanztests der Unterschiede zwischen den *MAE*s verschiedener Systeme einfach möglich sind.
Wenig aussagekräftig ist die Anwendung des *MAE* jedoch, wenn die Granularität der Nutzerpräferenz gering ist. Nimmt ein Nutzer bspw. bei einem Empfehlungssystem mit fünfwertiger Skala und den Werten 1 bis 5 innerlich nur eine Aufteilung in gute und schlechte Objekte (mit der Wertung 3.5 als Grenze) vor, so wird der Fehler einer vorhergesagten Bewertung $p_i = 4$ bei einer tatsächlichen Bewertung $r_i = 5$ für diesen Nutzer uninteressant sein.
Der mittlere quadratische Fehler bzw. *Mean Squared Error* ist eine Variation des *MAE*, bei der durch die Quadrierung größere Abweichungen stärker gewichtet werden. Nachteil ist jedoch die Empfindlichkeit gegenüber einzelnen „Ausreißern“: Große Unterschiede zwischen vorhergesagter und tatsächlicher Bewertung, auch wenn sie bezogen auf die Gesamtzahl N der Vorhersagen selten auftreten, können den Wert des *MSE* stark beeinflussen.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (p_i - r_i)^2 \tag{3.2}$$

Die Wurzel aus dem mittleren quadratischen Fehler (*Root Mean Squared Error*) hat dieselben Eigenschaften wie der *MSE*, jedoch mit dem Vorteil, dass dieselbe Maßeinheit wie bei den Bewertungen benutzt wird und somit eine leichtere Interpretierbarkeit der Fehlerwerte gegeben ist.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (p_i - r_i)^2} \tag{3.3}$$

[GOLDBERG et al. 2001] führten den normalisierten mittleren absoluten Fehler (*Normalized Mean Absolute Error*) ein. Dabei wird der *MAE* bzgl. der Größe des Bewertungsintervalls ($\tilde{r}_{max}$ entspricht dabei der maximalen und $\tilde{r}_{min}$ der minimalen möglichen Bewertung für Objekte innerhalb der benutzten Bewertungsskala) normalisiert, um (theoretisch) einen Vergleich der Vorhersagequalität eines

Empfehlungsverfahrens auf verschiedenen Datensätzen mit unterschiedlichen Bewertungsskalen zu ermöglichen.

$$\text{NMAE} = \frac{1}{\tilde{r}_{max} - \tilde{r}_{min}} \cdot \text{MAE} \tag{3.4}$$

Eine weitere Variation ist die separate Anwendung des *MAE* auf Objekte, die von Nutzern mit sogenannten „Extremwerten“ bewertet wurden,[8] sowie auf Objekte ohne solche Extrembewertungen. Diese Bewertungsform wurde von [SHARDANAND und MAES 1995] eingeführt.

2. *Maße für Klassifikationsgenauigkeit - Anzahl der Objekte, die in die richtigen/falschen Bewertungsklassen eingeteilt wurden*
   Diese Gruppe von Qualitätsmaßen ermittelt die Häufigkeit, mit der ein Empfehlungssystem korrekte bzw. falsche Vorhersagen darüber macht, ob ein Objekt empfehlenswert ist oder nicht. Dementsprechend sind diese Maße im Gegensatz zu den vorherigen tolerant gegenüber kleineren Abweichungen in der Vorhersagegenauigkeit, solange dies nicht zu einer falschen Klassifikation führt. Daher eignen sich diese Maße besonders für Empfehlungssysteme, die eine binäre Einteilung der Objekte in „empfehlenswert“/“nicht empfehlenswert“ vornehmen (wie z.B. für Systeme, die der *Finde gute Objekte*-Strategie folgen). Da solche Strategien eine Rangliste von Empfehlungen erzeugen, findet man unter den Qualitätsmaßen entsprechend viele aus dem Bereich des *Information Retrieval* (*IR*), wo Ranglisten von Texten, die bzgl. einer Menge von Suchbegriffen relevant sind und die Verbesserung der Ranglisten aus Sicht von Nutzern eine zentrale Rolle spielen. Somit treten hier auch dieselben Probleme auf wie im *IR*. Beim *Recall*-Maß, das später noch genau beschrieben wird, berechnet man bspw. den Anteil relevanter, d.h. empfehlenswerter Objekte in der vom System präsentierten Rangliste gegenüber den insgesamt in der Objektmenge vorhandenen empfehlenswerten Objekten. Das Problem hierbei ist, die Anzahl der insgesamt in der Objektmenge vorhandenen empfehlenswerten Objekte für einen Nutzer zu bestimmen, wenn dieser - wie in realen Systemen meist der Fall - nicht alle vorhandenen Objekte gesehen und somit bewertet hat. Um mit diesem Problem umzugehen, gibt es im Bereich von Empfehlungssystemen drei Strategien:

   a) *unbewertete Objekte ignorieren*
      Hierbei werden aus der generierten Liste von Empfehlungen vor der Qualitätsbewertung all jene Empfehlungen entfernt, die sich auf Objekte beziehen, die vom Nutzer nicht bewertet wurden. Nachteil an dieser Methode ist eine Verfälschung der Qualität des Empfehlungsverfahrens, da Empfehlungen für Objekte, die nur wenige Nutzer bewertet haben, fast komplett aus der Qualitätsbewertung herausfallen, während Empfehlungen für häufig bewertete Objekte („Mainstream“) unverhältnismäßig hoch gewichtet werden.

   b) *Defaultwerte*
      Unbewertete Objekte werden mit einer Defaultbewertung versehen, die oft leicht negativ ist.[9] Nachteil dieser Methode ist, dass diese Defaultwerte erheblich von der tatsächlichen Bewertung abweichen können.

   c) *Schätzung der Gesamtzahl guter Objekte*
      Eine Methode, die aus dem *Information Retrieval* als *vollständige Relevanzbeurteilung* bekannt ist. Eine Stichprobe in Form einer vollständig bewerteten Liste von Objekten (hier kann die zuvor genannte Methode des Ignorierens unbewerteter Objekte verwendet werden) wird auf die Anzahl der positiv bewerteten Objekte untersucht. Von dem Verhältnis dieser Objekte zur Gesamtzahl der Objekte in der Stichprobe wird auf die Gesamtzahl empfehlenswerter Objekte in der Datenbasis geschlossen. Auch diese Methode ist relativ ungenau, da aussagekräftige Stichproben meist mehr Objekte enthalten müssten, als Nutzer in der Regel bewertet haben. Andere, aussagekräftigere Abschätzungsmethoden aus dem *IR* wie die *objektive Relevanzbeurteilung*

[8] D.h. besonders guten bzw. besonders schlechten Werten.
[9] Vgl. [BREESE et al. 1998].

bei der Experten Objekte als relevant bzw. irrelevant bewerten oder *Pooling*, wo die Relevanzbeurteilungen mehrerer Nutzer zusammengefasst werden, bis alle Objekte abgedeckt sind, sind für Empfehlungssysteme nicht anwendbar, da hier die Relevanz von Objekten im Gegensatz zu Texten bezogen auf eine Menge von Suchbegriffen sehr subjektiv ist.

d) Für Empfehlungssysteme existieren diverse Variationen der ersten und dritten Methode unter Aufsplitten der Bewertungsdaten in Trainings- und Testmengen. In [HERLOCKER et al. 2004] sind diese genauer beschrieben.

Die bekanntesten, aus dem *Information Retrieval* stammenden Maße zur Beurteilung der Klassifikationsgenauigkeit sind *Precision*, *Fallout* und das bereits erwähnte *Recall*-Maß, die bereits 1968 von [CLEVERDON und KEAN 1968] eingeführt und seitdem beibehalten wurden. Beide Maße unterteilen die Gesamtmenge ALL aller Objekte in der Datenbasis in relevante und nicht relevante Objekte, so dass für Empfehlungsalgorithmen mit einer nicht binären Bewertungsskala die Bewertungen vor der Qualitätsbeurteilung in das binäre Format umgewandelt werden müssen. Daraus ergibt sich auch, dass diese Maße zur Qualitätsbeurteilung ungeeignet sind, wenn die Granularität der Nutzerpräferenz nicht binär ist, sich der Nutzer also bspw. für die konkrete Rangordnung innerhalb der Menge relevanter Objekte interessiert.
Die *Precision* gibt dabei bezogen auf eine Empfehlungsliste L das Verhältnis der Menge $R_L \subseteq L$ relevanter Objekte zur Gesamtzahl der Objekte in der Liste und somit die Wahrscheinlichkeit an, dass ein beliebiges Objekt der Liste relevant ist.[10] Deswegen findet die *Precision* Anwendung bei solchen Empfehlungssystemen, wo die *Finde gute Objekte*-Aufgabe erfüllt wird:

$$P = \frac{|R_L|}{|L|} \tag{3.5}$$

Das *Recall*-Maß setzt die Anzahl der relevanten Objekte in der Empfehlungsliste nicht in Beziehung zur Gesamtzahl aller Objekte in der Liste, sondern zur Gesamtzahl relevanter Objekte R_{all} in der gesamten Datenbasis, mit der erwähnten Schwierigkeit der Bestimmung von R_{all}. Dabei gibt der *Recall* im Gegensatz zur *Prediction* die Wahrscheinlichkeit an, dass ein relevantes Objekt aus der Datenbasis in der Empfehlungsliste auftaucht und ist somit für die Qualitätsbewertung all jener Empfehlungssysteme geeignet, bei denen die *Finde alle guten Objekte*-Aufgabe zur Anwendung kommt.

$$R = \frac{|R_L|}{|R_{all}|} \tag{3.6}$$

Ein eher selten benutztes Maß, das als Gegenstück zum *Recall* existiert, ist der *Fallout*, der die Anzahl nicht relevanter Objekte in der Empfehlungsliste im Verhältnis zur Anzahl aller irrelevanten Objekte in der Datenbasis angibt und somit die Fähigkeit des Systems beschreibt, irrelevante Objekte vom Nutzer fernzuhalten. Daher ist dieses Maß besonders für die Qualitätsbewertung von Empfehlungssystemen geeignet, bei denen die Kosten falsch positiver Empfehlungen extrem hoch sind. Wurde zuvor bereits der *Recall* ermittelt, kann zur Berechnung des *Fallouts* auf bereits bekannte Größen zurückgegriffen werden, wobei sich wieder das Problem der Berechnung der Menge R_{all} ergibt.

$$R = \frac{|L| - |R_L|}{|ALL| - |R_{all}|} \text{ mit } R_{all} \subseteq ALL \tag{3.7}$$

Ein von *Precision* und *Recall* abgeleitetes ad hoc-Maß kann Einsatz finden, wenn der Klassifikationsfehler eines Empfehlungssystems bestimmt werden soll, aber nur wenig Daten zur Verfügung stehen, wie z.B. in einem Online-Experiment, wo anhand der bisher generierten Empfehlungen für einen Zielnutzer die Fehlerrate für diesen Nutzer errechnet werden soll. Der Klassifikationsfehler ergibt sich dann als die Anzahl der vom Nutzer als inkorrekt bewerteten Empfehlungen im Verhältnis zu

[10] Die Angabe eines Wahscheinlichkeitswertes macht bei der *Precision* wie auch beim *Recall* einen großen Vorteil aus, da Wahrscheinlichkeitswerte vom Nutzer leichter interpretierbar sind („das Empfehlungssystem empfiehlt mit einer Wahrscheinlichkeit von 70% ein nützliches Objekt“) als etwa der mittlere absolute Fehler („das Empfehlungssystem hat einen *MAE* von 0.78“).

allen bisher generierten Empfehlungen. Da es insbesondere bei längeren Empfehlungslisten vorkommen kann, dass nicht alle Empfehlungen der Liste vom Nutzer bewertet wurden, wird dieses Maß meist insofern abgewandelt, dass nur die bewerteten Empfehlungen der Liste in die Berechnungen einbezogen werden, was die zuvor erwähnten Schwierigkeiten mit sich bringt.

$$E = \frac{|L| - |R_L|}{|L|} \text{ oder } E = \frac{|L_{rated}| - |R_L|}{|L_{rated}|} \tag{3.8}$$

Im *IR* werden *Recall* und *Precision* oft in einem so genannten *RP*-Diagramm aufgetragen. Beim Vergleich zweier Systeme a und b ergibt sich jedoch ein Problem, wenn a bzgl. eines der beiden Maße besser ist als b, b jedoch bezogen auf das andere Maß wiederum a überlegen ist - es stellt sich die Frage, wie *Precision* und *Recall* zueinander gewichtet werden sollen. Untersuchungen haben dabei gezeigt, dass sich *Recall* und *Precision* invers zueinander verhalten und von der Länge der Empfehlungsliste abhängen. Exakte Vergleiche von Systemen bzgl. beider Maße sind somit schwierig. Als ein Versuch der Annäherung hat sich das so genannte *F*-Maß durchgesetzt, dass *Recall* und *Precision* zu einem einzigen Wert zusammenfasst:

$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R} \tag{3.9}$$

β ist dabei ein Parameter, der die relative Gewichtung des *Recall*-Wertes angibt ($\beta = 0$ - nur *Precision* zählt, $\beta = \infty$ - nur *Recall* zählt). Meist wird wie bei [SARWAR et al. 2000b] das F_1-Maß mit $F_1 = \frac{2PR}{P+R}$ benutzt.

Eine Alternative zu den *Precision*- und *Recall*-Maßen ist die so genannte *ROC*-Kurve (*Relative Operating Characteristic* oder *Receiver Operating Characteristic*), die sich aus der Signaltheorie entwickelt hat und die Fähigkeit eines Systems misst, zwischen Signal (relevante Empfehlungen) und Rauschen (nicht relevante Empfehlungen) zu unterscheiden. Von der Annahme ausgehend, dass jedem Objekt vom Empfehlungssystem ein vorhergesagter Relevanzgrad zugewiesen wird, erhält man zwei Verteilungen, die für nicht relevante bzw. relevante Objekte die Wahrscheinlichkeit angeben,

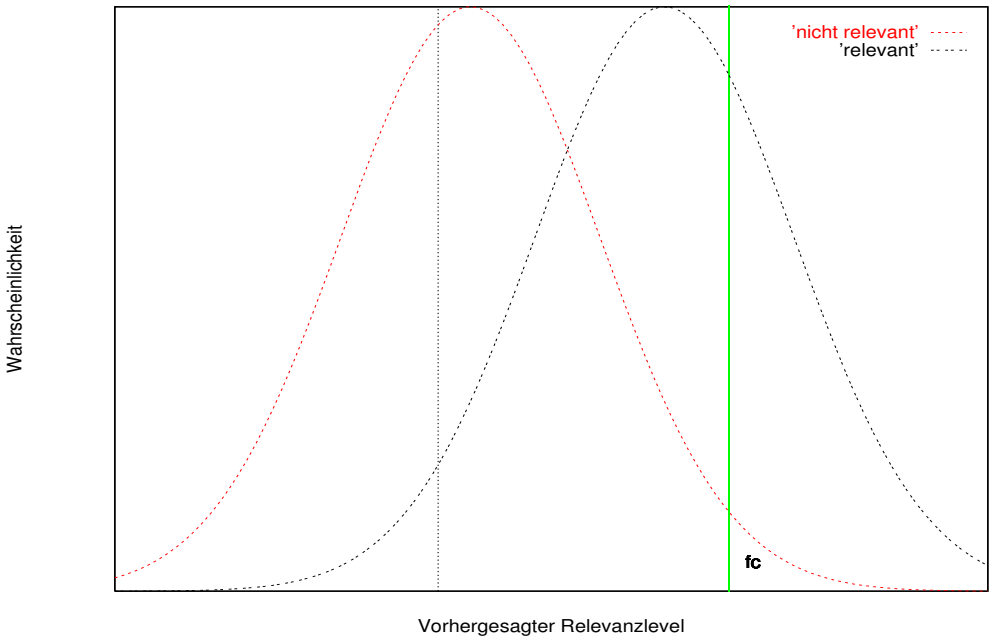


Abbildung 3.1: *ROC*-Kurve: Beispielverteilungen für relevante und irrelevante Beispiele

dass den Objekten ein bestimmter Relevanzgrad zugewiesen wird. Je „weiter“ die Verteilungen dabei auseinander liegen, desto besser ist das System in der Lage, zwischen relevanten und nicht relevanten Objekten zu unterscheiden. Eine weitere Annahme ist, dass Nutzer eine ihnen präsentierte Rangliste von Empfehlungen im allgemeinen von oben nach unten durchgehen, bis ein bestimmtes Abbruchkriterium (z.B. Anzahl gesehener Empfehlungen oder investierte Zeit) erreicht ist. Abb. 3.1 zeigt ein Beispiel für die Verteilung (nicht-)relevanter Objekte und das Abbruchkriterium als grün gefärbte Gerade fc, die die Empfehlungen der Rangliste in gesehene (alle Ränge oberhalb von fc) und ungesehene (alle Ränge unterhalb von fc) aufteilt. Je nach Wahl von fc ergeben sich unterschiedliche *Recall*-Werte (Fläche unterhalb der schwarzen Kurve, rechts von fc), sowie *Fallout*-Werte (Fläche unterhalb der roten Kurve, rechts von fc), die die Prozentzahl relevanter bzw. nicht relevanter Objekte innerhalb der Rangliste angeben, die bis zur Erfüllung des Abbruchkriteriums gesehen wurden. Die *ROC*-Kurve ist dabei nichts anderes, als eine Darstellung des Verhältnisses zwischen *Recall* und *Fallout* für verschiedene Werte von fc. Hat man ein Relevanzkriterium festgelegt und die Menge R_u der relevanten Objekte bzw. $\overline{R}_u$ der nicht relevanten Objekte bzgl. dieses Kriteriums und der Bewertungen eines Nutzers u bestimmt, so lässt sich solch eine *ROC*-Kurve einfach zeichnen, indem nach dem Erstellen einer nach vorhergesagten Bewertungen geordneten Rangliste von Objekten ausgehend von Rang 1 nacheinander jedes Objekt bis zum Ende der Liste betrachtet wird. Ist das Objekt relevant (bezogen auf das gewählte Relevanzkriterium und die tatsächliche, vom Nutzer abgegebene Bewertung), dann wird die *ROC*-Kurve um eine Einheit $(1/|R_u|)$ nach oben verlängert, ist das Objekt nicht relevant (bzgl. derselben genannten Kriterien), so findet eine Verlängerung um eine Einheit $(1/|\overline{R}_u|)$ nach rechts statt. Nicht bewertete Objekte werden dabei ignoriert.

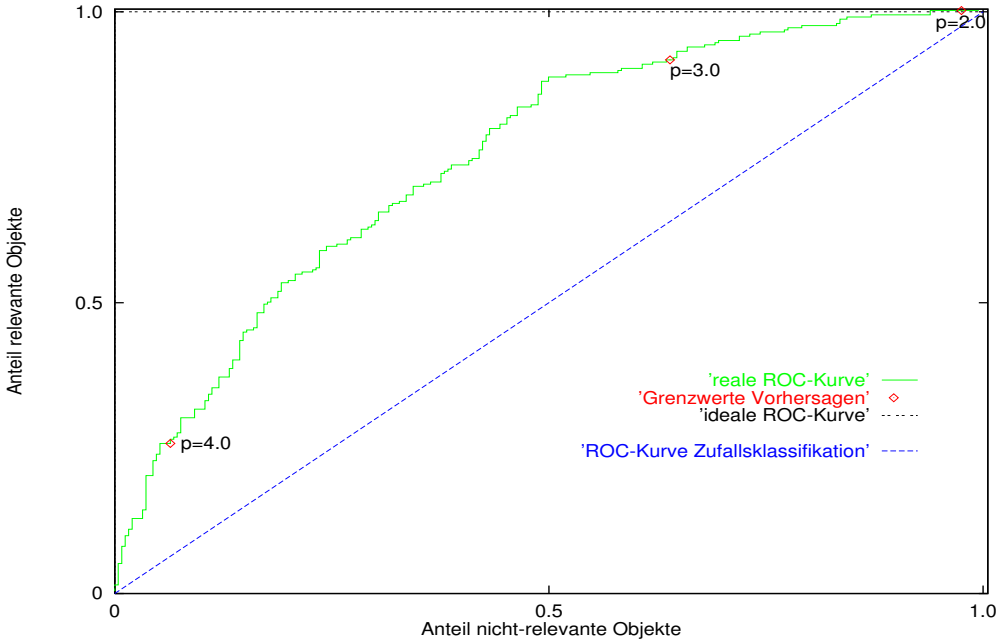


Abbildung 3.2: Beispiel einer *ROC*-Kurve

Abbildung 3.2 zeigt das Beispiel einer nach dieser Vorgehensweise erstellten *ROC-Kurve* anhand der Daten des „Power-Users“ $u =$ „M. Korth“, der $|M_u| = 521$ Filme bewertet hat. Als Klassifikationsgrenze wurde der Wert 3.5 einer fünfstufigen Bewertungsskala gewählt, d.h. alle Filme $m_i \in M_u$ mit $i = 1, \ldots, |M_u|$ und Bewertungen $r(m_i) \geq 3.5$ mit $r(x) \in \{1, 2, 3, 4, 5\}$ wurden als relevant und alle Objekte mit Bewertungen $r(m_i) < 3.5$ als nicht relevant eingestuft. Dasselbe galt für vorhergesagte Bewertungen $p(m_i)$ mit $p(x) \in [0 \ldots 5]$. Die Vorhersage der Bewertungen und damit der

Klassifikationen erfolgte durch den als k-*Nearest-Neighbor*[11] parametrisierten kollaborativen Algorithmus aus Kapitel 5 mit $k = 100$ und unter Anwendung der *All But One*-Methode, d.h. bei 521 Durchgängen wurden jeweils die Bewertungen von 520 Filmen als Trainingsdaten verwendet, um damit die Bewertung für den übrig gebliebenen Film vorherzusagen. Alle 521 Filme wurden dann nach absteigenden vorhergesagten Bewertungen in einer Rangliste geordnet und mit den tatsächlich abgegebenen Bewertungen auf Basis der erwähnten Klassifikation verglichen.
Die sich daraus ergebende *ROC*-Kurve ist in Abb. 3.2 grün gefärbt. Zum Vergleich sieht man in der Abb. auch eine optimale *ROC-Kurve* (schwarz gestrichelt), die erst senkrecht bis 1.0 auf der Y-Achse verläuft (alle tatsächlich als relevant bewerteten Filme kommen in der Rangliste zuerst) und anschließend waagerecht auf der X-Achse nach rechts bis 1.0 (nach den relevanten Filmen folgen hintereinander alle nicht relevanten Filme). Außerdem ist eine *ROC*-Kurve, wie sie erwartungsgemäß ein Verfahren erzeugen würde, welches Filme per Zufall (d.h. abwechselnd) als relevant/nicht relevant klassifiziert, als gestrichelte blaue Linie zu sehen. Die roten Punkte markieren Grenzwerte bzw. Abbruchkriterien bezogen auf die vorhergesagten Bewertungen. Betrachtet man z.B. alle Filme der Rangliste mit einer vorhergesagten Bewertung von $p = 4$ oder höher und bricht die Betrachtung dann ab, so hat man anschließend ca. 20% der relevanten und 6% der irrelevanten Filme gesehen. Die Fläche unterhalb der Kurve (das sog. *Sweat's a measure*) beträgt ca. 0.74719 und ist äquivalent zur Wahrscheinlichkeit, mit der der 100-*Nearest-Neighbour*-Algorithmus korrekt zwischen zwei zufällig gezogenen Filmen (einem aus der Menge der relevanten und einem aus der Menge der irrelevanten Filme) von M. Korth unterscheiden kann.
Großer Vorteil der *ROC*-Kurve ist, dass mit dem *Sweat's a measure* die Gesamtvorhersagequalität des Empfehlungssystems mittels eines einzigen Wertes dargestellt werden kann, der zudem von der Länge der Empfehlungsliste unabhängig ist. Vergleiche zwischen verschiedenen Empfehlungssystemen sind somit einfach möglich und durch die wohlgeformte, hinter dem *ROC*-Maß stehende Entscheidungstheorie statistisch abgesichert. Dafür benötigt man jedoch viele relevante Objekte bezogen auf das Relevanzmaß, um eine aussagekräftige *ROC*-Kurve zu erhalten,[12] insbesondere wenn die Differenz zwischen den *Sweat's a measure*-Werten zweier miteinander verglichener Systeme gering ist. Außerdem hat die Vertauschung der Reihenfolge zweier Objekte (von denen eins relevant und das andere nicht relevant ist) in der vorhergesagten Rangliste den gleichen Effekt, unabhängig davon, ob sie am Anfang oder am Ende der Rangliste auftreten. Daher eignet sich das *ROC*-Maß nicht für die *Finde gute Objekte*-Aufgabe, sondern ist eher für Kontexte wie die *Finde alle guten Objekte*-Aufgabe geeignet, wo es entscheidend ist, alle relevanten Objekte zu sehen, die spezifische Reihenfolge dieser Objekte jedoch nicht ins Gewicht fällt.

3. *Maße für die Ranggenauigkeit - Anzahl der Objekte, die auf den richtigen/falschen Plätzen einer Rangliste stehen*
   Die zuvor vorgestellten Maße für die Klassifikationsgenauigkeit ermöglichen es nur, die Korrektheit der Trennung von Objekten in empfehlenswerte und nicht empfehlenswerte durch das Empfehlungssystem zu beurteilen. Wie erwähnt sind solche Maße ungeeignet, wenn die Granularität der Nutzerpräferenz nicht binär ist. Ist dies der Fall und das Empfehlungssystem arbeitet ebenfalls mit einer nicht binären Bewertungsskala, so kommen Maße für die Ranggenauigkeit zum Einsatz. Solche Maße erkennen bspw. auch, dass in den Top-10 der generierten Empfehlungsliste zwar nur relevante Objekte vorhanden sind, dass beste Objekt jedoch nicht auf Rang 1, sondern Rang 10 eingeordnet wurde.
   Das erste Maß, der *Pearson-Koeffizient*, wird meist innerhalb von Empfehlungssystemen zur Erstellung der Empfehlungen bzw. der Nutzerpräferenz benutzt, kann aber auch zur Qualitätsbewertung eingesetzt werden. Der *Pearson-Koeffizient* ist ein bekanntes Korrelationsmaß,[13] bei dem die Abhängigkeit der Werte einer Variable von den Werten einer zweiten Variable gemessen wird, d.h. die Existenz eines linearen Zusammenhangs zwischen zwei Variablen überprüft wird. Bezogen auf das

[11] *Nearest Neighbor*-Verfahren werden in Kapitel 5 genauer beschrieben.
[12] Vgl. [HANLEY und MCNEIL 1982].
[13] Siehe [RESNICK et al. 1994].

Problem der Ranggenauigkeit eines Empfehlungsverfahrens wird die Korrelation zwischen den nicht binären Bewertungen von Objekten durch einen Nutzer und den für dieselben Objekte vorhergesagten nicht binären Bewertungen durch das Empfehlungssystem bestimmt. Ein Wert von 1 bezeichnet dabei eine perfekte Korrelation zwischen Nutzer und Empfehlungssystem, d.h. das System bewertet die Objekte innerhalb der Empfehlungsliste exakt so, wie es auch der Nutzer getan hat. Ein Wert von 0 hingegen weist darauf hin, dass kein Zusammenhang zwischen den Bewertungen des Nutzers und des Empfehlungssystems besteht, während ein negativer Wert anzeigt, dass Nutzer und System bei der Bewertung der Objekte entgegengesetzter Meinung sind.
Korrelationsmaße haben den Vorteil, dass sie Personen mit wissenschaftlicher Ausbildung ein gutes und leicht zu verstehendes Qualitätsbild des jeweiligen Empfehlungssystems in der kompakten Form eines einzigen Wertes vermitteln.

$$r(u,s) = \frac{\sum_{j \in O(u,s)} (v_{u,j} - \overline{v}_u)(v_{s,j} - \overline{v}_s)}{\sqrt{\sum_{j \in O(u,s)} (v_{u,j} - \overline{v}_u)^2 \cdot \sum_{j \in O(u,s)} (v_{s,j} - \overline{v}_s)^2}} \tag{3.10}$$

Dabei bezeichnet $O(u,s)$ die Menge der Objekte, die der Nutzer u und das System s gemeinsam bewertet haben. $v_{u,j}$ bzw. $v_{s,j}$ entspricht den Bewertungen des Nutzers bzw. des Systems für das aktuell betrachtete Objekt $j \in O(u,s)$, während mit $\overline{v}_u$ bzw. $\overline{v}_s$ der Mittelwert aller Bewertungen des Nutzers bzw. des Systems gekennzeichnet wird.
Spearman's ρ-Maß entspricht dem *Pearson*-Koeffizient bis auf die Tatsache, dass hier nicht Bewertungen v, sondern Ränge r betrachtet werden. Somit wird von konkreten Bewertungen abstrahiert, so dass eine Übereinstimmung von Nutzer und System auch dann gemessen wird, wenn die Objekte unterschiedlich bewertet wurden, die aufgrund der Bewertungen entstehenden Ordnungen der Objekte jedoch übereinstimmen.

$$\rho = \frac{\sum_{j \in O(u,s)} (r_{u,j} - \overline{r}_u)(r_{s,j} - \overline{r}_s)}{\sqrt{\sum_{j \in O(u,s)} (r_{u,j} - \overline{r}_u)^2 \cdot \sum_{j \in O(u,s)} (v_{r,j} - \overline{r}_s)^2}} \tag{3.11}$$

Ein Nachteil des ρ-Maßes ist dann gegeben, wenn ein Nutzer zwei Objekte gleich bewertet hat. In solch einem Fall ist es egal, ob Objekt 1 vor Objekt 2 in der Rangliste steht oder umgekehrt.[14] Das ρ-Maß beachtet jedoch auch in so einem Fall die Reihenfolge der Objekte und weist einem System ein schlechtes Ergebnis zu, wenn die Reihenfolge der gleich bewerteten Objekte in der Empfehlungsliste des Systems und der Liste des Nutzers unterschiedlich ist.
Eine noch weitergehende Abstraktion von konkreten Werten als beim ρ-Maß wird mit Kendall's *Tau*-Maß erreicht, wo Ränge ohne konkrete Werte betrachtet werden. Dazu überprüft man die Rangliste L_s, die durch das Empfehlungssystem s erstellt wird, paarweise auf „konkordante" Paare, d.h. Paare die in der durch die Bewertungen des Nutzers u entstehenden Rangliste L_u von Objekten in derselben Reihenfolge stehen. Die Anzahl dieser konkordanten Paare wird mit C bezeichnet. Die Anzahl D der „diskordanten" Paare, deren Reihenfolge in beiden Listen unterschiedlich ist, wird ebenfalls festgehalten. Schließlich bestimmt man innerhalb der Empfehlungsliste L_s des Systems noch die Anzahl T_s der Paare, die „verknüpft" sind, d.h. bei denen die vorhergesagten Bewertungen identisch sind und entsprechend die Anzahl T_u der verknüpften Paare in der Liste des Nutzers. Das *Tau*-Maß ergibt sich dann als:

$$\text{Tau} = \frac{C - D}{\sqrt{(C + D + T_u)(C + D + T_s)}} \tag{3.12}$$

Beim *Tau*-Maß wird ein Vertauschen der Reihenfolge zweier Objekte in der Empfehlungsliste gegenüber der Liste des Nutzers gleich stark bestraft, egal ob die Objekte auf Rang 1 und 2 oder Rang 999 und 1000 stehen, was nicht der normalen Beurteilung aus Nutzersicht entspricht. Außerdem besteht wie beim ρ-Maß die Gefahr unrechtmäßig schlechter Qualitätswerte bei Objektlisten mit schwacher

[14] Dies wird im *Information Retrieval* auch als „schwache Ordnung" im Gegensatz zu einer „starken Ordnung" bezeichnet, wo alle Objekte unterschiedliche Bewertungen und somit eine eindeutige Reihenfolge haben.

Ordnung.
Ein Rangordnungsmaß das gefahrlos zum Vergleich schwacher Ordnungen benutzt werden kann, ist das normalisierte distanzbasierte Performanzmaß (*Normalized Distance-based Performance Measure*, NDPM). Es handelt sich dabei um einen entscheidungstheoretischen Ansatz, der von [YAO 1995] eingeführt wurde. Dabei wird die Anzahl C^- sich widersprechender Präferenzrelationen in der Rangfolge der vom System generierten Empfehlungsliste und der Liste des Nutzers gezählt. Eine sich widersprechende Präferenzrelation tritt auf, wenn das System vorhersagt, dass der Nutzer Objekt i gegenüber Objekt j bevorzugen wird (weil es höher in der Rangordnung steht), in Wirklichkeit jedoch das Gegenteil der Fall ist. Entsprechend wird eine Übereinstimmung zwischen System und Nutzer als Anzahl C^+ kompatibler Präferenzrelationen ermittelt und zusätzlich die Gesamtzahl C^i von Präferenzrelationen in der Liste des Nutzers, d.h. die Paare von Objekten, die aufgrund unterschiedlicher Bewertungen eine starke Ordnung haben.

$$\text{NDPM} = \frac{2C^- + C^+}{2C^i} \tag{3.13}$$

Die durch den Zähler des Bruchs ausgedrückte Distanz wird zur schlimmstmöglichen Distanz in Beziehung gesetzt, wobei diese Maximaldistanz normalisierend wirkt und somit Vergleiche zwischen verschiedenen Datenbasen möglich macht. Leider wird auch hier das Problem des *Tau*-Maßes nicht behoben, dass keine Gewichtung der Position erfolgt, an der Vertauschungen der Reihenfolge eines Objektpaares auftreten.
Schließlich existiert noch ein Maß zur Bewertung der Rangfolge von Objekten in einer generierten Empfehlungsliste speziell für lange Listen, bei denen ersichtlich ist, dass ein Nutzer sich wahrscheinlich nicht die Zeit nehmen wird, die gesamt Liste zu betrachten. Das *Half-life*-Nützlichkeitsmaß von [BREESE et al. 1998] definiert die Nützlichkeit eines empfohlenen Objekts an Position j der Rangliste als Differenz zwischen seiner Bewertung $r_{u,j}$ durch den Nutzer u und einer Defaultbewertung d, die meist neutral oder leicht negativ ist. Somit hat ein empfohlenes Objekt für den Nutzer umso mehr Nützlichkeit, je höher es vom Nutzer bewertet wurde. Der *Half-life*-Aspekt des Maßes gewichtet die Nützlichkeit zusätzlich anhand des Ranges, an dem das Objekt in der Empfehlungsliste auftaucht. Die Gewichtung nimmt mit steigendem Rang exponentiell ab, wobei der Parameter α die Stärke der Gewichtungsabnahme bestimmt (üblich ist $\alpha = 5$). Die Nützlichkeit der von einem System für den Nutzer u erstellten Rangliste mit n Rängen ergibt sich dann als

$$R_u = \sum_{j=1}^{n} \frac{\max(r_{u,j} - d, 0)}{2^{(j-1)/(\alpha-1)}} \tag{3.14}$$

Ein Empfehlungssystem erzeugt also dann eine optimale Qualität R_u^{max} bzgl. des *Half-life*-Maßes, wenn die Objekte in exakt der Reihenfolge in der Empfehlungsliste auftauchen, wie sie sich aus einer Ordnung der Objekte nach absteigenden vom Nutzer vergebenen Bewertungen für diese Objekte ergibt. Schließlich kann auch die Nützlichkeit der vom System erzeugten Ranglisten bezogen auf alle Nutzer $u \in U$ ermittelt werden als

$$R = 100 \cdot \frac{\sum_{\forall u \in U} R_u}{\sum_{\forall u \in U} R_u^{max}} \tag{3.15}$$

Das *Half-Life*-Maß ist am besten für die *Finde gute Objekte*-Aufgabe geeignet, vorausgesetzt der Zeitfaktor spielt eine kritische Rolle (z.B. ein Empfehlungssystem, das als Automat in einer Videothek installiert ist und von vielen Kunden genutzt wird) oder für die *Finde alle guten Objekte*-Aufgabe, wo Nutzer alle nützlichen Objekte sehen wollen. Allerdings müssen die Parameter d und α weise gewählt sein. Ist α falsch gewählt, so stimmt die Funktion, die die Abnahme der Nützlichkeitsgewichtung beschreibt nicht mit der realen Funktion überein und man erhält ein stark verfälschtes Ergebnis. Einige reale Nützlichkeitsfunktionen lassen sich auch gar nicht mit der *Half-Life*-Funktion ausdrücken. Viele Menschen neigen z.B. bei einer Rangliste dazu, sich die ersten $10 - 20$ Empfehlungen anzusehen (was ausdrückbar ist), aber auch die letzten $10 - 20$ Empfehlungen, während der Mittelteil ignoriert

wird (was sich mit der *Half-Life*-Funktion nicht mehr nachbilden läßt).
Beim Parameter d muss beachtet werden, dass wegen der *max*-Funktion alle Objekte, die mit einer Bewertung $r_{a,j} < d$ bewertet wurden gleich stark in die Qualitätsbewertung eingehen, ungeachtet, ob sie auf Rang 2 oder am Ende der Rangliste auftauchen. Ist d also zu hoch gewählt, so hat dies ebenfalls ein wenig aussagekräftiges Ergebnis zur Folge.

Neben den aufgelisteten traditionellen Maßen haben sich in letzter Zeit auch alternative Maße entwickelt, die sich mehr auf den praktischen Nutzen von generierten Empfehlungen aus Sicht einer Anwenderin konzentrieren. Obwohl noch in den Anfängen begriffen und oft noch nicht genau spezifiziert oder erprobt, sollen die bisherigen Ansätze im Folgenden beschrieben werden.

Alternative Qualitätsmaße (praxisbezogene Maße)

Schon die pure Anzahl der zuvor aufgelisteten Qualitätsmaße bezogen auf die Genauigkeit von Empfehlungssystemen weist auf die Intensität hin, mit der dieses Thema untersucht wurde. Seit das erste automatisierte Empfehlungssystem implementiert wurde, wurden umfangreiche Forschungen und Versuche unternommen, die Genauigkeit stetig zu steigern. Wie bereits erwähnt, scheint die Grenze der Verbesserungsfähigkeit in diesem Bereich erreicht, während Empfehlungssysteme jedoch noch weit davon entfernt sind, perfekt zu sein. Dieses Problem wurde von einigen Forschern erkannt, die angefangen haben, ihre Aufmerksamkeit auf andere Qualitätskriterien zu lenken und entsprechend neue Qualitätsmaße zu entwickeln. Die zwingende Notwendigkeit dieses Umdenkens und der Ernst der Bemühungen werden durch [HERLOCKER et al. 2004] sehr gut zusammengefasst:

> *„There is an emerging understanding that good recommendation accuracy alone does not give users of recommender systems an effective and satisfying experience. Recommender systems must provide not just accuracy, but also* usefulness.“

Ein Empfehlungssystem sollte zwar eine vernünftige Vorhersagegenauigkeit aufweisen, jedoch rechtfertigen geringe Unterschiede bzgl. dieses Qualitätsmaßes keinen extremen Aufwand, solange die Nützlichkeit der empfohlenen Objekte für Systeme mit unterschiedlicher Vohersagegenauigkeit gleich ist. Entsprechende Untersuchungen wurden im Bereich des *Information Retrieval* mit Suchmaschinen vorgenommen (siehe [TURPIN und HERSH 2001]). Für Nutzer sind geringe Differenzen der Vorhersagegenauigkeit unerheblich, da die Granularität ihrer persönlichen Präferenzen meistens ohnehin relativ gering ist (siehe [HILL et al. 1995] und eigene Versuchsergebnisse in Kapitel 5.2). Auch können akzeptable Unterschiede in der von verschiedenen Systemen erstellten Rangordnung empfohlener Objekte durch den Nutzer ausgeglichen werden. Entscheidender ist vielmehr, ob der Nutzer sein Ziel erreicht, was bei der Benutzung eines Empfehlungssystems offensichtlich darin besteht, für ihn nützliche Objekte schnell und ohne großen Aufwand zu finden. Einige (meist innovative) Maße, die die Nützlichkeit generierter Empfehlungen auf verschiedene Art und Weise messen, sollen im Folgenden vorgestellt werden.

1. *Abdeckung - Anzahl der Objekte aus der Datenbank, für die eine Vorhersage generiert werden kann*
   Hierbei handelt es sich um ein Bewertungsmaß, dass schon von verschiedenen Forschern, wie z.B. [GOOD et al. 1999], [HERLOCKER et al. 1999] oder [SARWAR et al. 1998] benutzt wurde. Es beurteilt, für wieviele Objekte der Datenbasis ein Empfehlungssystem Vorhersagen generieren kann und es auch tatsächlich tut. Das Abdeckungsmaß eignet sich insbesondere zur Qualitätsbestimmung von Systemen, die die *Finde alle guten Objekte-* oder *Annotation in Context*-Aufgaben erfüllen, denn wenn ein System für einige Objekte der Datenbasis keine Vorhersagen generieren kann, dann können unter diesen Objekten auch potenziell gute Objekte vorhanden sein, die dem Nutzer so verloren gehen. Auch kann nur dann für Objekte ein Kommentar generiert werden, wenn eine Bewertung vorhanden ist. Allerdings sollte das Abdeckungsmaß stets im Zusammenhang mit der Vorhersagegenauigkeit gemessen werden, da sonst für alle vorhandenen Objekte im Zweifelsfall Defaultvorhersagen generiert werden könnten, die zwar zu einer hohen Abdeckung führen, jedoch für einen Benutzer nicht nützlich sind.
   Das Abdeckungsmaß kann dabei auf verschiedene Art und Weise angewendet werden:

- *Vorhersageabdeckung - Anzahl der Objekte, für die Vorhersagen generiert werden können*
  Hierbei geht es nur um die grundsätzliche Fähigkeit eines Systems, für alle vorhandenen Objekte Vorhersagen zu generieren. Je nachdem, was mit dem Begriff „alle vorhandenen“ gemeint ist, unterscheidet man wiederum zwei Typen:
  - *über alle Objekte*
    Das am häufigsten eingesetzte Maß, bei dem der Datenbasis eine zufällige Menge von Objekten entnommen und der prozentuale Anteil der Objekte an der Gesamtzahl von Objekten des Samples bestimmt wird, für die Vorhersagen generiert werden können.
  - *über bewertete Objekte*
    Wie zuvor, doch werden nur solche Objekte in die Betrachtung mit einbezogen, die der jeweilige Nutzer auch bewertet hat. Diese Variante der Abdeckung ist realistischer, weil sie sich an den realen Bedürfnissen von Nutzern orientiert. Ein Nutzer, der an einer bestimmten Gruppe von Objekten kein Interesse hat, wird solche Objekte auch nicht bewerten. Somit ist es unerheblich für ein System, ob es für diesen Nutzer und solche Objekte Vorhersagen erstellen kann oder nicht.
- *Katalogabdeckung - Anzahl von möglichen Vorhersagen, die tatsächlich generiert werden*
  Diese Variante baut (meistens) auf der Vorhersageabdeckung auf. Wenn ein System in der Lage ist, Vorhersagen für alle Objekte in der Datenbank zu generieren, heißt dies noch nicht, dass auch tatsächlich alle Objekte empfohlen werden. Die Menge der tatsächlich empfohlenen Objekte zu ermitteln kann insbesondere im kommerziellen Bereich wichtig sein, wo ein Katalog von Objekten (daher der Name) vorhanden ist und für niemals empfohlene Objekte auch nur eine sehr geringe Wahrscheinlichkeit besteht, dass sie gekauft werden. Meist berechnet man die Katalogabdeckung, indem man zu einem bestimmten Zeitpunkt für jeden Nutzer die x besten für ihn erstellten Empfehlungen betrachtet und aus diesen Top-x-Mengen von Objekten der einzelnen Nutzer die Vereinigungsmenge erstellt.

2. *Lernrate - Schnelligkeit, mit der ein System seine Vorhersagequalität steigert*
   Viele Empfehlungssysteme berechnen ihre Vorhersagen basierend auf statistischen Maßen. Diese Maße benötigen eine gewisse Anzahl von Daten, bevor zufriedenstellende Empfehlungen generiert werden können, d.h. je mehr Daten vorhanden sind, desto besser ist die Qualität der Vorhersagen.[15] Es gilt zu bestimmen, wie schnell ein Empfehlungssystem in der Lage ist einen Grad zu erreichen, wo „akzeptable“ Empfehlungen erzeugt werden können.
   Da die Lernrate eines Systems nichtlinear und asymptotisch ist (irgendwann ist die maximale Lernfähigkeit erreicht) ist es schwierig, sie auf einfache und kompakte Weise darzustellen. Meist wird deshalb zum Vergleich verschiedener Systeme die Qualität (z.B. Vorhersagegenauigkeit) der Anzahl von vorhandenen Daten gegenübergestellt, auf denen das statistische Maß zur Vorhersagegenerierung beruht (z.B. die Anzahl von Bewertungen für kollaborative Systeme). Für einige Systeme ist die Verwendung der Lernrate als Qualitätsmaß ungeeignet, da sie nicht lernen (z.B. nützlichkeitsbasierte und wissensbasierte Systeme). Ein Beispiel für die praktische Anwendung der Lernrate findet sich bei [SCHEIN et al. 2001] im Zusammenhang mit der Lösung des „cold-start“-Problems bei kollaborativen Verfahren.
   Allgemein können drei verschiedene Lernraten in Augenschein genommen werden:

   - *Gesamtlernrate*
     Hier ist die Lernrate eine Funktion, die abhängig von der Anzahl der Gesamtdaten (Bewertungen, Nutzer oder Objekte) ist.
   - *Lernrate pro Objekt*
     Nur die auf ein Objekt bezogenen Daten (z.B. Bewertungen oder Nutzer, die das Objekt bewertet haben) werden betrachtet.

[15] Besonders kollaborative Verfahren leiden an diesen Startproblemen.

- *Lernrate pro Nutzer*
  Entsprechend findet die Berechnung der Lernrate in diesem Fall bezogen auf die Daten statt, die für einen Nutzer vorhanden sind (z.B. die Anzahl der Objekte, die die Nutzerin eines eigenschaftsorientierten Systems bewertet haben muss, damit dieses akzeptable Vorhersagen generieren kann).

3. *Neuheit (Novelty) - Unbekanntheit der empfohlenen Objekte*
   Ein Empfehlungssystem kann eine hervorragende Vorhersagegenauigkeit, Abdeckung und Lernrate aufweisen und trotzdem kann ein Benutzer des Systems mit den erzeugten Empfehlungen unzufrieden sein, nämlich in dem Fall, in dem ihm Objekte empfohlen werden, die er alle schon kennt. Hier gilt es, andere Maße zu finden, um die Qualität eines Systems zu beurteilen. Die Neuheit einer Empfehlung definiert sich dadurch, dass das empfohlene Objekte einem Nutzer unbekannt ist. Untersuchungen zur Neuigkeit von Objekten liegen aus dem Bereich des *Information Retrieval* vor[16] und wurden unter anderem von [SARWAR et al. 2001] auf den Bereich von Empfehlungssystemen übertragen.
   Dabei ist eine Möglichkeit, die Neuheit von empfohlenen Objekten zu messen bei Systemen gegeben, die Bewertungen/Empfehlungen mit Zeit- und Datumsstempel versehen. Die bewerteten/empfohlenen Objekte bis zu einem bestimmten Zeitpunkt können dann als Trainingsmenge verwendet und damit bestimmt werden, wie viele Objekte der Testmenge dem jeweiligen Nutzer empfohlen werden.

4. *Serendipity - unerwartete und nützliche Empfehlungen*
   Serendipity lässt sich in etwa mit „glücklicher Zufall“ oder „die Gabe, zufällig glückliche und unerwartete Entdeckungen zu machen“ übersetzen. Dieses Qualitätsmaß stellt eine Verschärfung des Neuheitsmaßes dar und muss unbedingt von diesem unterschieden werden. *Serendipity* bezeichnet empfohlene Objekte, die nicht nur neu für den Nutzer sind, sondern die er ohne die Empfehlung des Systems auch niemals entdeckt hätte und die ihm außerordentlichen Nutzen bringen. Die Neuheit einer Empfehlung wäre bspw. schon gegeben, wenn einem Nutzer ein Film seines Lieblingsregisseurs empfohlen wird, den er noch nicht gesehen hat. Für das *Serendipity*-Maß hingegen dürfen solche inhaltlichen Verbindungen nicht bestehen, sondern es muss sich vielmehr um Objekte handeln, die der bewussten Wahrnehmung des Nutzers total entgehen. Auch wenn sich einige Forscher wie [SARWAR et al. 2001] schon mit dem *Serendipity*-Problem befasst haben, so ist dieses Maß noch kaum im Einsatz, insbesondere auch deshalb, weil es ohne direkte Mithilfe des Nutzers nur schwer zu messen ist. Versuche wurden gemacht, indem basierend auf allen Nutzern Listen „offensichtlicher“ Objekte erstellt wurden (z.B. durch statistische Maße) und diese Objekte mit den Empfehlungen für einen Nutzer verglichen wurden. Jedoch müssten zur automatischen Messung der *Serendipity* die gesamten Vorlieben eines Zielnutzers ermittelt werden. Anhand von Objekten, die der Nutzer bewertet hat, die aber aus seinen Vorlieben herausfallen (*Outlier Detection*) müssten die Kriterien solcher Objekte ermittelt werden, die sie von der breiten Masse der Objekte unterscheiden, damit die Frage beantwortet werden kann, unter welchen Umständen ein Nutzer bereit ist, seinen Horizont zu erweitern. Allerdings stehen die Forschungen in diesem Bereich der Qualitätsbewertung von Empfehlungssystemen erst am Anfang.

5. *Vertrauen - Sicherheit eines Systems bzgl. der Korrektheit der von ihm generierten Empfehlung*
   Die von einem Empfehlungssystem generierten Empfehlungen können aus Nutzersicht bzgl. zweier Dimensionen interpretiert werden. Die *Stärke* einer Empfehlung drückt aus, wie sehr der Zielnutzer aus Sicht des Systems das empfohlene Objekt mögen wird (z.B. zeigt eine 5 auf einer fünfwertigen Skala von 1 = „sehr schlecht“ bis 5 = „sehr gut“ eine hohe Stärke an). Die zweite Dimension wird jedoch oft vernachlässigt: Das *Vertrauen* in die Bewertung zeigt an, wie sicher sich das Empfehlungssystem ist, dass die von ihm generierte Bewertung korrekt ist. Stehen bspw. bei einem kollaborativen Empfehlungssystem für ein Objekt nur wenige Bewertungen in der Datenbank zur Verfügung, aus denen eine Vorhersage generiert werden kann, so kann das Vertrauen des Systems in die vorhergesagte Bewertung nur gering sein. Da ein Empfehlungssystem vorwiegend auch entscheidungsunterstützend eingesetzt wird (denn die letztendliche Entscheidung, ob ein empfohlenes

[16] Siehe [BAEZA-YATES und RIBIERO-NETO 1999].

Objekt gekauft/genutzt wird oder nicht liegt immer noch beim Nutzer), sollten Systeme zusätzlich zur reinen Vorhersage das *Vertrauen* in dieselbige anzeigen. Dabei ist auch die Art der Anzeige wichtig, wie [HERLOCKER et al. 2000] in Untersuchungen herausgefunden haben. Eine schlechte Darstellung des *Vertrauenswerts* wirkt weit weniger unterstützend bei der Entscheidungsfindung als gar keine Darstellung, während eine gute Darstellung die Fähigkeit des Nutzers, sich für das richtige Objekt zu entscheiden, enorm steigern kann. Solche Anzeigen sind besonders deshalb wichtig, weil eine mit wenig Vertrauen generierte Vorhersage nicht automatisch bedeutet, dass ein Nutzer diese Empfehlung verwerfen wird. Vielmehr kann es Kontexte und Stimmungen geben, in denen eine Person Neues ausprobieren möchte (und dabei auch gewillt ist, ein Risiko einzugehen), während in anderer Stimmung und/oder anderem Kontext lieber die sichere Empfehlung vorgezogen wird. Wie bei allen bisher vorgestellten Qualitätsmaßen besteht auch hier wieder das Problem, *Vertrauen* als multidimensionales Phänomen zu messen. Je nach Empfehlungssystem und Kontext sind sowohl quantitative (z.B. Anzahl der zu Rate gezogenen Bewertungen eines Objekts durch andere Nutzer), als auch qualitative Ansätze (die Bewertung stammt von einem Nutzer, dessen hohe Korrelation zur eigenen Präferenz bereits in der Vergangenheit ermittelt und durch Feedback bzgl. generierter Empfehlungen bestätigt wurde) zur Messung denkbar.

6. *Evaluierung durch Nutzer - Die Qualitätsbewertung erfolgt durch Nutzerinnen in einer realen Anwendung*
   Einige der alternativen Ideen zur Beurteilung der Qualität eines Empfehlungssystems sind schwer in konkrete Maße zu fassen. Noch schwieriger wird es allerdings, wenn die Gesamtqualität eines Systems (inkl. z.B. des angebotenen Interfaces) gemessen werden soll. In solchen Fällen ist es am einfachsten, das System einem realen Test zu unterziehen und die Benutzer bzgl. der Qualität Bewertungen abgeben zu lassen. [HERLOCKER et al. 2004] unterteilen dabei solch eine praktische Evaluierung in mehrere Bewertungsdimensionen:

   - *Explizite oder implizite Bewertung*
     Nutzer können nach ihrer Bewertung gefragt oder ihr Verhalten interpretiert werden. Untersuchungen von [CLAYPOOL et al. 2001] und [MORITA und SHINODA 1994] haben dabei gezeigt, dass wenn möglich stets explizite und implizite Bewertungen kombiniert werden sollten, da Nutzer bei Empfehlungssystemen mit gleicher Performance (z.B. Vorhersagegenauigkeit) trotzdem eine Präferenz für eines der Systeme entwickeln können. Sammelt man in Versuchen sowohl explizite als auch implizite Bewertungen, so kann man später eine Korrelation zwischen Performance und Präferenz herstellen.
   - *Laboruntersuchungen oder Feldstudien*
     Während Laboruntersuchungen dazu geeignet sind, ganz spezielle Eigenschaften von Systemen zu evaluieren oder Hypothesen zu beweisen/widerlegen, zeigen Feldstudien das Verhalten von Nutzern in ihrem eigenen Kontext. Sollen bspw. die Aufgaben *Selbstausdruck* oder *Finden von vertrauenswürdigen Empfehlungssystemen* untersucht werden, so sind Feldversuche besser geeignet als Laboruntersuchungen.
   - *Ergebnis oder Prozess - Bewertung des Nutzens/Bewertung der Kosten*
     Bei den meisten Qualitätsbewertungen geht es um das Sammeln konkreter Ergebnisse, z.B. der Bewertung der Vorhersagequalität eines Empfehlungssystems durch den Nutzer. Oft wird dabei jedoch die Bewertung des Prozesses vernachlässigt, d.h. die Zeit und Mühen, die ein Nutzer investieren musste, um an die Empfehlungen zu kommen. Vom Standpunkt des Kosten-Nutzen-Prinzips können hohe Prozesskosten die hohe Vorhersagequalität eines Systems relativieren.
   - *Kurzzeit- oder Langzeituntersuchungen*
     Einige Eigenschaften von Systemen können erst nach längerer Laufzeit von Tests untersucht werden, z.B. die Fähigkeit eines Systems mit langfristigen Wandlungen in der Präferenz von Nutzern zurecht zu kommen oder Unzufriedenheit von Nutzern mit einem an sich guten System, dessen schlechtes Interface nach längerer Benutzung demotivierend wirkt.

Diese relativ neuen praxisbezogenen Maße dienen als Grundlage für ein eigenes Qualitätsmaß, das im Folgenden hergeleitet wird.

3.3 Interessantheit von Empfehlungen

> *„For any task, appropriate metrics must be developed that define what counts as successful outcome.“*
>
> [NEWMAN 1997]

> *„Recommender systems must provide not just accuracy, but also* usefulness. [...] *We need new dimensions for analyzing recommender systems that consider the 'nonobviousness' of the recommendation.“*
>
> [HERLOCKER et al. 2004]

Nachdem in diesem Kapitel bereits die Gefahren beim Einsatz rein hypothetischer Qualitätsmaße dargestellt und einige mehr praxisbezogene Alternativen vorgestellt wurden, soll nun den oben angeführten Zitaten folgend ein entsprechendes Maß für die Versuche dieser Arbeit definiert werden. Die Definition beruht dabei auf zwei Überlegungen:

1. *Der größte Nutzen von automatisierten Empfehlungssystemen liegt in der Fähigkeit, unerwartete Empfehlungen zu generieren, die den Horizont einer Nutzerin erweitern.*
   Untersuchungen von [SWEARINGEN und SINHA 2001] haben gezeigt, dass einige Menschen im Zusammenhang mit Empfehlungssystemen durchaus dazu neigen, Empfehlungen für Objekte zu bevorzugen, die ihnen bereits bekannt sind. Diese Menschen, meist neue Nutzer eines Empfehlungssystems, neigen dazu, die Empfehlungssysteme umfangreich zu „testen“, bevor sie ihnen ihr Vertrauen schenken (vgl. auch *Aufgaben von Empfehlungssystemen* in Abschnitt 3.2). Auch gibt es Menschen mit starkem Sicherheitsstreben, die aufgrund schlechter Erfahrungen in der Vergangenheit (z.B. kostspielige Fehlkäufe aufgrund falscher Empfehlungen) Empfehlungen für Objekte bevorzugen, die denen sehr ähnlich sind, die sie bereits kennen und als gut bewertet haben, so dass die „neuen“ Empfehlungen nur eine geringe Gefahr bergen, erneute Enttäuschung hervorzurufen.[17] Diese Verhaltensweisen sind aus subjektiver Sicht der betroffenen Personen verständlich und sollten wegen ihrer positiven Wirkung auf diese Personen akzeptiert und gewürdigt werden. Empfehlungssysteme können so konfiguriert werden, dass sie diesen Anforderungen entsprechen. Diese Methode ist bspw. bei einigen kommerziellen Empfehlungssystemen verbreitet, denn laut [HERLOCKER et al. 2004] ist die erste Regel von Empfehlungssystemen aus Sicht von E-commerce Managern: „Don't make me look stupid!“. Das Empfehlen bekannter Objekte oder von Objekten, die den bekannten sehr ähnlich sind, kann nach Sicht des Autors jedoch nicht Sinn von Empfehlungssystemen sein, da Menschen Objekte, die ihnen bekannt sind und von ihnen geschätzt werden, sowieso im Blickfeld ihrer Aufmerksamkeit behalten und deren Entwicklung besser verfolgen und einschätzen können als automatisierte Empfehlungssysteme. Bezogen auf solche „ähnlichen“ Objekte holen Menschen auch selten Empfehlungen anderer Menschen ein und die persönliche Erfahrung des Autors zeigt für die Fälle, in denen doch Empfehlungen eingeholt werden oder erfolgen, dass „negative Empfehlungen“ (in dem Sinne, dass von bestimmten Objekten abgeraten wird), von solchen Personen meist ignoriert werden („davon möchte ich mich selbst überzeugen“), während positive Empfehlungen bereitwillig angenommen werden („self-fulfilling prophecy“).

[17] Interessanterweise zeigen dieselben Personen nach [HERLOCKER et al. 2004] bei kostenlosen Objekten wie z.B. mp3s oder Publikationen ein gänzlich anderes Verhalten und bevorzugen unbekannte Objekte.

> *„Das Komische am Leben ist: Wenn man darauf besteht, nur das Beste zu bekommen, dann bekommt man es häufig auch.“*
>
> W. Somerset Maugham

Im Gegensatz zu diesem Verhaltensschema ist jedoch in vielen Menschen auch die Lust auf Neues, die Neugier ausgeprägt. Sie ziehen Befriedigung daraus, etwas für sie Neues entdeckt zu haben, dass ihren Gefallen findet und nehmen bei der Suche nach diesem Neuen eine gewisse Anzahl von Fehlschlägen in Kauf. Manche Menschen genießen dabei sogar die „Gefahr“, Fehlschlägen ausgesetzt zu sein. Allgemein gesehen besteht jedoch bei zu vielen Fehlschlägen die Gefahr, dass man letztlich auf seiner Suche demotiviert wird. Hier liegt der eigentliche Sinn von Empfehlungssystemen, da sie durch ihre Kapazität zur Verarbeitung großer Datenmengen die Anzahl der Fehlschläge minimieren können und einer Anwenderin Zeit bei der Suche nach Neuem, dass ihren Gefallen findet, ersparen.
Im Bereich der Filme - um den es hier ja geht - zeigt sich insbesondere, dass für den Erfolg eines Films oft eine gute Idee, die Präsentation von etwas Neuem, dass es so vorher noch nicht gab, wichtiger ist als ausgefeilte Kulissen und Spezialeffekte. Nicht umsonst waren es gerade sog. *Low-Budget*-Filme, die den größten Erfolg hatten und im Gedächtnis der Zuschauer geblieben sind, weil sie diesen etwas gaben, dass sie so noch nicht gesehen hatten. Beispiele für solche Filme aus verschiedenen Genres sind *Beverly Hills Cop*, *Bonnie und Clyde*, *Die Reifeprüfung*, *Dirty Dancing*, *Blair Witch Project*, *Matrix*, *Pulp Fiction*, *Night of the Living Death*, *Toy Story* und *Harold and Maude*. Ihren Erfolg verdanken solche Filme oft der Mundpropaganda, da die meisten Personen sich diese Filme ohne vorherige Empfehlung wohl nicht angesehen hätten. Auch die persönliche Erfahrung des Autors ist, dass solche Filme den größten Eindruck hinterlassen, die man sich ohne entsprechenden Anstoß nie angesehen hätte. Als der Autor eines Abends mit einem Freund ins Kino ging, stellte sich heraus, dass der Film, in den man gehen wollte, ausverkauft war. Der Freund überredete den Autor schließlich als Ersatz in die Action-Komödie *Tödliche Weihnachten* zu gehen. Dazu war wegen einer Abneigung des Autors gegen die mitwirkende Schauspielerin Geena Davis erhebliche Überredungskunst nötig. Eine ähnliche Abneigung gegen den Schauspieler Brad Pitt hätte fast den Genuss des Films *Fight Club* verhindert. Auch den Film *K-Pax* wollte der Autor zuerst lieber meiden und musste zum Kinogang überredet werden. Alle drei Filme gehören heute genauso zu den Lieblingsfilmen des Autors, wie der Film *Uncorked (At Sachem Farm)*, in den sich der Autor beim Fernsehen per Zufall „hineinzappte“ und den er sich aufgrund der Beschreibung in einer Fernsehzeitschrift niemals angesehen hätte. Daher soll das in dieser Arbeit verwendete praxisorientierte Qualitätsmaß das Auftreten unerwarteter Empfehlungen messen, die für eine Zielnutzerin bereichernd wirken, da der Autor darin den größten Nutzen von automatisierten Empfehlungssystemen sieht. Nach diesen Überlegungen alleine könnte auch auf das im vorigen Abschnitt vorgestellte Qualitätsmaß *Serendipity* zurückgegriffen werden, jedoch spielt noch eine weitere Überlegung eine Rolle.

2. *Praxisorientierte Maße sollen hypothetische Maße nicht ersetzen, sondern ergänzen.*
   Auf die Wichtigkeit der Hinwendung zu mehr praxisorientierten Maßen wurde bereits mehrfach hingewiesen. Jedoch wurde auch erwähnt, dass die hypothetischen Qualitätsmaße, mit denen die Genauigkeit eines Empfehlungssystems gemessen wird, weiterhin wichtig sind. Niemand, egal ob im privaten, wissenschaftlichen oder kommerziellen Bereich würde ein neu entwickeltes Empfehlungssystem ohne vorherige Tests für einen praktischen Einsatz freigeben. Hypothetische Maße wie der *MAE* können klären, ob ein Empfehlungsverfahren grundsätzlich funktionsfähig ist und sinnvolle Vorhersagen liefert. Erst nach dieser Klärung sollte untersucht werden, ob sich das Verfahren auch im realen praktischen Einsatz bewährt, die hypothetische Genauigkeit sollte also als notwendige Bedingung für die Nützlichkeit einer Empfehlung dienen, während darauf aufbauend die *Serendipity* die hinreichende Bedingung für die Nützlichkeit repräsentiert.
   So benutzten [SWEARINGEN und SINHA 2001] in einem praktischen Versuch mit Online-Empfehlungssystemen die Kategorie „useful recommendation“, mit der Nutzer empfohlene Bücher bzw. Filme bewerten sollten, die sie noch nicht gelesen bzw. gesehen hatten und die sie aufgrund der gegebenen Beschreibungen interessant fanden. Im Gegensatz dazu konnten Nutzer Empfehlungen auch als

„previously liked recommendations“ klassifizieren. Beide Klassifikationen wurden unter der Oberkategorie „good recommendation“ zusammengefasst. [SWEARINGEN und SINHA 2001] griffen dabei jedoch auf bestehende, meist kommerzielle Empfehlungssysteme zurück, bei denen davon auszugehen ist, dass sie vor ihrem praktischen Einsatz bereits umfangreich auf ihre hypothetische Funktionsfähigkeit getestet wurden. Daher reichte in diesem Fall ein zweistufiges Qualitätsmaß aus. In dieser Arbeit sollen jedoch zwei Empfehlungssysteme selbst implementiert werden. Daher wird die Genauigkeit dieser Implementierungen vor den praktischen Versuchen durch hypothetische Qualitätsmaße ermittelt werden. Ein typisches Problem im maschinellen Lernen ist jedoch, dass ein Verfahren, das auf Trainings- und Testdaten gute Lernergebnisse geliefert hat, bei Anwendung auf eine andere Teilmenge der Domäne wesentlich schlechtere Ergebnisse liefern kann, da Rauschen oder fehlende Daten als Lernergebnis eine Hypothese (in unserem konkreten Fall eine Hypothese über die Präferenz eines Nutzers bezogen auf Filme) bedingen können, die auf die gesamte Domäne (hier Gesamtmenge der benutzten Filme, siehe Kapitel 4) bezogen falsch ist. Dies wird auch als *Overfitting* einer Hypothese bezeichnet.[18] Aus diesem Grund soll die Genauigkeit der Vorhersage eines der in dieser Arbeit implementierten Empfehlungssystems durch den Nutzer im praktischen Versuch bestätigt oder widerlegt werden, was letzten Endes zu einem dreistufigen Qualitätsmaß führt.

Das aus den beiden vorherigen Überlegungen resultierende Qualitätsmaß soll als *Interessantheit* von Empfehlungen bezeichnet werden:

Definition 1. *Sei M eine Menge von Filmen, U eine Menge von Nutzern eines Empfehlungssystems RS für Filme und S eine Skala möglicher Bewertungen für Filme unter Benutzung von RS. Habe ein Nutzer $u \in U$ eine Teilmenge $M_u^{all} \subseteq M$ der Filme in M gesehen und eine Teilmenge $M_u \subset M_u^{all}$ ihm bekannter Filme mit Bewertungen $s_u(m_i^u)$ versehen, wobei $m_i^u \in M_u$ und $s_u(m_i^u) \in S$. Habe RS basierend auf den Trainingsdaten M_u und $S_u = \{s_u(m_i^u) \mid m_i^u \in M_u \wedge s_u(m_i^u) \in S\}$ eine Hypothese H_u erstellt, die die Präferenzen des Nutzers u beschreibt.*
Sei $R_u \subset M$ eine Menge von Empfehlungen für u, wobei für jede Empfehlung $r_i^u \in R_u$, mit $i = 1, \ldots, |R_u|$ gilt $r_i^u \notin M_u$ und das Bewertungssystem RS für jedes r_i^u mittels der Hypothese H_u eine Vorhersage $s_{RS}(r_i^u)$ bzgl. der Bewertung von r_i^u durch den Nutzer u generiert, die seinen Präferenzen entspricht, d.h. $H_u(r_i^u) = s_{RS}(r_i^u)$ mit $s_{RS}(r_i^u) \in S$.
Sei weiterhin $serendipity(x)$ eine boolesche Funktion, die Empfehlungen r_i^u bzgl. der Anforderungen des gleichnamigen Qualitätsmaßes aus Abschnitt 3.2 einordnet, d.h.:

$$serendipity(r_i^u) = \begin{cases} 1 & \text{wenn } r_i^u \notin M_u^{all} \text{ und } r_i^u \text{ ist unerwartet und von großem Nutzen} \\ 0 & \text{sonst} \end{cases}$$

Dann heißt eine Empfehlung $r_i^u \in R_u$ mit zugehöriger vorhergesagter Bewertung $s_{RS}(r_i^u)$ „interessant“, genau dann, wenn die notwendige Bedingung $s_{RS}(r_i^u) \approx s_u(r_i^u)$ und die hinreichende Bedingung $serendipity(r_i^u) = 1$ erfüllt sind.

Wie genau diese Definition dabei für die praktischen Versuche umgesetzt wird, beschreibt Kapitel 4.

[18] Vgl. Definition in [MITCHELL 1997], Seite 67.

4 Beschreibung der Versuchsumgebung

„Tonight, my dear Maxwell, I'm ready to try my experiment on a human!"

Dr. Meirschultz, *Maniac*, 1934

In diesem Kapitel werden die verschiedenen Werkzeuge beschrieben, die - entweder bereits vorliegend oder extra für diese Arbeit erstellt - zur Untersuchung der Interessantheit von automatisch generierten Filmempfehlungen herangezogen wurden. Bei den bereits vorhandenen Hilfsmitteln wurde sowohl auf externe Quellen als auch auf am Lehrstuhl vorhandene Programme und Daten zugegriffen. Für das Erstellen zusätzlicher eigener Werkzeuge wurde aus Gründen der Portabilität und zum Einfügen in die Programmierphilosophie des Lehrstuhls die Programmiersprache *Java* verwendet.
Wie in Abschnitt 1.1 erwähnt, wird in dieser Arbeit ein kollaborativer Algorithmus implementiert, dem als „state of the art"-Verfahren besondere Bedeutung zukommt. Wie bei jedem kollaborativen Verfahren, ist auch hier das „cold-start"-Problem gegenwärtig. Um dieses Problem zu umgehen, wird auf die bereits erwähnte Methode des *seedings* zurückgegriffen, d.h. Bewertungen, die im Rahmen eines anderen kollaborativen Algorithmus vorgenommen wurden, werden hier als Datenbasis für den implementierten Algorithmus nach [Yu et al. 2003] in Kapitel 5 verwendet. Die Wahl fiel dabei auf die sogenannte *MovieLens*-Datenbank.

4.1 Die MovieLens-Datenbank

Die *MovieLens*-Datenbank wurde von der *GroupLens*-Forschungsgruppe[1] des Fachbereichs Informatik der Universität von Minnesota als experimentelle Plattform zum Studium der Bereiche *Empfehlungssysteme*, *kollaboratives Filtern*, *Informationsfilterung* und *Design und Theorie von Gemeinschaften im Internet* erstellt. Hauptziel war dabei die Entwicklung und das Studium von Empfehlungssystemen, die wirklichen Nutzern bei wirklichen Problemen helfen, indem sie diesen Nutzern „gute" Filmempfehlungen liefern. Die *MovieLens*-Datenbank ist sowohl online als interaktives System, als auch in Auszügen als Offline-Datenbank[2] verfügbar.
Die Offline-Datenbank steht für Forschungszwecke kostenlos in zwei Größen zum Download bereit. Die kleine Datenbank umfasst 100000 Bewertungen von 1682 Filmen durch 943 Nutzer, wobei jeder Nutzer mit mindestens 20 Filmbewertungen vertreten ist und diese Bewertungen im Zeitraum vom 19.9.1997 bis 22.4.1998 abgegeben wurden. Die große Datenbank enthält 1000209 Bewertungen von 3900 Filmen durch 6040 *MovieLens*-Nutzer aus dem Jahre 2000. Beide Datenbanken bestehen aus mehreren Dateien im Textformat und beinhalten neben den Filmbewertungen auch einfache demographische Nutzerdaten wie Alter, Geschlecht und Beruf. Die Bewertungen der Filme liegen auf einer ganzzahligen numerischen Skala von 1 bis 5, wobei 1 die schlechteste und 5 die beste Bewertung darstellt.
Als Basis für das Erstellen der *MovieLens*-Datenbank (zum Beheben des *cold-start*-Problems) diente die bekannte *EachMovie*-Datenbank der HP/Compaq-, ehemals DEC-Forschungsgruppe. Ähnlich wie die *MovieLens*-Datenbank waren auch die *EachMovie*-Daten für die HP/Compaq-Gruppe zur Erforschung kollaborativer Filteralgorithmen gedacht. Die *EachMovie*-Daten fanden in dem gleichnamigen Empfehlungssystem Verwendung, das für 18 Monate in den Jahren 1995 bis 1997 als kostenloser öffentlicher Dienst zur Verfügung gestellt wurde. Nach Beendigung dieses Angebots im September 1997 wurden die *EachMovie*-

[1] Die Forschungsgruppe steht unter der Leitung der Professoren John Riedl und Joseph Konstan.
[2] Auf der *GroupLens*-Webseite im Abschnitt „Publicly Available Data".

Daten durch HP/Compaq zum Zwecke der allgemeinen Forschung veröffentlicht[3]. Mit einer Datenfülle von 2811983 Bewertungen von 1628 Filmen durch 72916 Nutzer fanden die *EachMovie*-Daten in zahlreichen Publikationen zum Thema kollaboratives Filtern Verwendung.
Aber auch die auf diesen Daten basierende kleinere Offline-*MovieLens*-Datenbank wurden von vielen Forschern benutzt.[4] Im Rahmen dieser Diplomarbeit wurde für den kollaborativen Empfehlungsansatz der große Offline-Datensatz der *MovieLens*-Datenbank aus mehreren Gründen verwendet.
Er ist weiterhin ohne spezielle Zugangsdaten frei im Netz verfügbar und wurde bereits in zahlreichen Forschungsvorhaben erprobt. Von der beinhalteten Datenmenge ist er einerseits groß genug, um mit ihm kollaborative Empfehlungsalgorithmen betreiben und bewerten zu können, im Vergleich zum *EachMovie*-Datensatz jedoch nicht so umfangreich, dass der in dieser Arbeit verwendete kollaborative Filteralgorithmus an seine Performancegrenzen stößt. Außerdem konnte für schnelle Tests der kleine *MovieLens*-Datensatz Anwendung finden, ohne erst Zeit für das Extrahieren einer kleineren Teilmenge aus dem Komplettdatensatz aufwenden zu müssen. Da der hier verwendete kollaborative Filteralgorithmus ursprünglich auf dem *EachMovie*-Datensatz getestet wurde,[5] ist es von Vorteil, dass die verwendeten *MovieLens*-Daten auf dem *EachMovie*-Datensatz basieren, so dass Vergleichsmöglichkeiten gegeben sind.
Nachteil der *MovieLens*-Datenbank ist jedoch das Fehlen inhaltsorientierter Informationen über die enthaltenen Filme. Da zur Untersuchung der Interessantheit von Filmempfehlungen auch inhaltsorientierte Empfehlungsverfahren Verwendung finden, musste für diese Verfahren ebenfalls eine geeignete Datenquelle gefunden werden.

4.2 Die Internet Movie Database

Die *Internet Movie Database (IMDB)* ist Filmfreunden mit Internetanschluss schon lange ein Begriff. Die *IMDB* wird ständig aktualisiert und erweitert und ist mit über 4 Millionen Filmen die größte Informationsquelle über Filme, die im Internet verfügbar ist. Insbesondere enthält diese Datenbank umfangreiche inhaltliche Informationen über Filme, wie mitwirkende Schauspieler, Regisseure, Handlung, beteiligte Kostümbildner oder Firmen für Spezialeffekte. Auch gibt es Bewertungen von Filmen, die im Gegensatz zu den *MovieLens*- und *EachMovie*-Datenbanken jedoch nicht einzeln für jeden Nutzer gegeben sind. Vielmehr werden nur die Gesamtzahl der Bewertungen, die Durchschnittsbewertung, sowie die Bewertungsverteilung für jeden Film angegeben, so dass diese Daten nicht als Grundlage für kollaborative Verfahren verwendet werden können.
Während die *IMDB* den meisten Internetnutzern nur als Online-Informationsquelle bekannt ist, werden die in der Datenbank enthaltenen Daten in verschiedenen Formaten auch zur Offline-Verwendung bereitgestellt, unter anderem in Form von Textdateien.
Da bereits früher am Lehrstuhl Untersuchungen bezogen auf die *IMDB*-Datenbank stattfanden, wurden die Informationen in den angesprochenen Textdateien bereits in eine *Oracle*-Datenbank eingepflegt. Diese Datenbank und teilweise auch die Textdateien an sich wurden sowohl als Grundlage für die inhaltsorientierten Empfehlungsverfahren als auch zum Erstellen einer Abbildung der Filmtitel der *MovieLens*-Datenbank auf die Titel der *IMDB*-Datenbank benutzt.

4.3 MovieMatcher

Um eine Verbindung zwischen den Datenquellen der kollaborativen und der inhaltsorientierten Empfehlungsverfahren herzustellen, mussten die Filmtitel in der *MovieLens*-Datenbank mit denen in der *IMDB*-Datenbank abgeglichen werden.
Dazu wurde ein Hilfsprogramm in *Java*, der so genannte *MovieMatcher* implementiert. Der *MovieMatcher*

[3] Mittlerweile hat Compaq diese Veröffentlichung eingestellt, auf Anfrage im Rahmen einer Forschungstätigkeit kann man durch Compaq aber einen Nutzernamen und ein Passwort erhalten, welche ein Herunterladen der *EachMovie*-Datenbank ermöglichen.
[4] Z.B. [MUI et al. 2001], [REDDY et al. 2002], [SARWAR et al. 2001] oder [SCHEIN et al. 2001].
[5] Siehe [YU et al. 2003].

liest die *MovieLens*-Datenbank und - um ortsunabhängig zu sein - die *IMDB*-Textdateien für die Originaltitel, die alternativen Titel und die alternativen deutschen Titel von Filmen in der *IMDB* ein und überführt diese Daten in geeignete Datenstrukturen. Um insbesondere in Bezug auf die über vier Millionen Filme in der *IMDB* Speicherplatz zu sparen, wird aus jeder Datei eine Indexdatei erstellt, die jedem Dateieintrag einen eindeutigen Index zuweist. So kann nachfolgend mit den Indizes gearbeitet werden. In einem ersten Schritt wird jeder Filmtitel aus der *MovieLens*-Datenbank in der *IMDB*-Datenbank gesucht. Bei Übereinstimmung werden die Jahreszahlen in beiden Datenbanken verglichen. Da sich gezeigt hat, dass derselbe Film in den beiden Datenbanken oft um ein Jahr bzgl. der Jahreszahl abweicht, werden Filme mit gleichem Titel und gleicher Jahreszahl bzw. mit einer Differenz von eins bezogen auf die Jahreszahl automatisch aufeinander abgebildet. Bei allen übrigen Filmen muss der Nutzer über Angabe des Index die Abbildung manuell vornehmen. Da die *IMDB*-Datenbank von vielen verschiedenen Nutzern erstellt wurde, gibt es auch innerhalb der einzelnen Dateien, die als Index den Originaltitel eines Films benutzen, Abweichungen bzgl. des Originaltitels. Deshalb wiederholt der *MovieMatcher* den Abbildungsvorgang für die alternativen und die deutschen Synchron-Filmtitel, so dass letzten Endes eine Datei mit vier Indizes pro Eintrag entsteht. Über die einzelnen Indexdateien kann so eine Abbildung der einzelnen Titel aufeinander erzeugt werden.
Wegen der teilweise erheblichen Unterschiede in den Titeln der *MovieLens*- und *IMDB*-Datenbank mussten von den 3900 Filmtiteln in der *MovieLens*-Datenbank über 600 Filmtitel per Hand abgeglichen werden. Die aus dieser Erfahrung gewonnenen Abweichungsregeln wurden festgehalten und können wie die Abbildung der Filmtitel aufeinander für spätere Untersuchungen mit ähnlicher Thematik wiederverwendet werden.
Ein weiterer Abgleich ergab sich in Vorbereitung auf die Nutzung inhaltlicher Eigenschaften von Filmen. Obwohl eine ältere Version der *IMDB* bereits als *SQL*-Datenbank am Lehrstuhl existiert, ergab sich aufgrund von Verarbeitungsschwierigkeiten von Sonderzeichen in Filmtiteln, sowie erneut differierender Jahreszahlen[6] die Notwendigkeit, mehrere hundert Filme beider *IMDB*-Datenbanktypen (textuell und *SQL*) per Hand miteinander abzugleichen bzw. nicht vorhandene Filme neu hinzuzufügen oder fehlerhaft eingetragene Filme zu korrigieren. So konnten für den späteren schnellen Zugriff auf inhaltliche Eigenschaften von Filmen die jeweiligen *SQL*-Tabellen-IDs der Filme gespeichert werden.
Die Abbildung der 3900 Filmtitel als Schnittmenge aus der textuell vorliegenden *IMDB*-Datenbank und der *MovieLens*-Datenbank bildet die Basis für die folgenden Untersuchungen und wird in einem weiteren für die Diplomarbeit erstellten Hilfsmittel zur Bewertung der Filme durch die Versuchspersonen benutzt, dem *MovieVoter*.

4.4 MovieVoter

Der *MovieVoter* wurde als graphische Benutzeroberfläche für die Interaktion mit den freiwilligen Versuchspersonen implementiert und unterstützt diese beim Prozess der Bewertung der Interessantheit von Filmempfehlungen. Dieser Prozess lässt sich dabei in drei Schritte unterteilen:

1. Bewertung von Filmen als Eingabe für die Empfehlungsalgorithmen
2. Sichten der Empfehlungen des jeweiligen Empfehlungsalgorithmus
3. Bewerten der Interessantheit der Filmempfehlungen

Das *MovieVoter*-Programm benutzt die vom *MovieMatcher* erstellte Abbildung der Filmtitel als Grundlage, genauer gesagt eine Datei, die aus dieser Abbildung erzeugt wurde. Jeder Eintrag für einen Film m_i in dieser Datei hat folgendes Format:

$$m_i = < \mathrm{MLID}_i, \mathrm{OT}_i, \mathrm{Y}_i, \mathrm{GL}_i, \mathrm{T}_i, \mathrm{ATL}_i > \text{ [7]}$$

[6] Auch die *SQL*-Datenbank wurde aus den erwähnten *IMDB*-Textdateien, in diesem Fall jedoch älteren Datums erstellt. Wegen der bereits erwähnten Fehleranfälligkeiten der Dateien treten diese Unterschiede besonders in verschiedenen Versionen auf. Aufgrund der ebenfalls erwähnten Probleme der *SQL*-Datenbank mit Sonderzeichen konnten leider auch nicht von Anfang an die Daten in dieser Datenbank benutzt werden.

[7] Eine separate Datei mit einer modifizierte Form von Filmtupeln wurde für die Arbeit mit dem eigenschaftsorientierten Algorithmus in Kapitel 6 verwendet, für den inhaltliche Eigenschaften von Filmen aus der erwähnten lehrstuhleigenen *SQL*-Datenbank benutzt

- MLID_i - Eindeutige ID des Films m_i in der *MovieLens*-Datenbank, diese ID wurde nicht durch den *MovieMatcher* erzeugt, sondern ist in der Offline-Version von *MovieLens* bereits vorhanden
- OT_i - Originaltitel des Films m_i in der *IMDB*
- Y_i - Erscheinungsjahr des Films m_i, entnommen aus der *IMDB*
- GL_i - Liste von Genres, denen der Film m_i durch die *MovieLens*-Datenbank zugeordnet wird
- T_i - Typ des Films m_i, wie in der *IMDB* gegeben mit $\text{T}_i \in \{$ *cinema movie*, *TV movie*, *video movie*, *computer game* $\}$, wobei der Großteil der 3900 Filme aus Kinofilmen besteht
- ATL_i - Liste von alternativen Titeln des Films m_i in der *IMDB*, wobei jeder Eintrag $\text{atl}_{ij} \in \text{ATL}_i$ folgendes Format hat:

$$\text{atl}_{ij} = < \text{AT}_{ij}, \text{C}_{ij}, \text{Y}_{ij}, \text{T}_{ij}, \text{AI}_{ij} >$$

Mit:

- AT_{ij} - Alternativer Titel j für den Film m_i
- C_{ij} - Land, in dem der alternative Titel AT_{ij} für den Film m_i vergeben wurde
- Y_{ij} - Jahr, in dem der Film m_i unter dem alternativen Titel AT_{ij} im Land C_{ij} gezeigt wurde
- T_{ij} - Typ des alternativen Titels AT_{ij}, aus demselben Wertebereich wie T_i
- AI_{ij} - Zusätzliche Informationen zum alternativen Titel AT_{ij}, wie z.B. die Angabe, dass es sich bei AT_{ij} um einen Arbeitstitel handelt

Die Datendatei mit dieser Art von Einträgen für alle 3900 benutzten Filme wird vom *MovieVoter* beim Programmstart eingeladen und später dem Nutzer präsentiert. Außerdem wird beim Start jeder Nutzer anhand des eingegebenen Namens identifiziert. Beim Erststart werden zudem weitere Nutzerinformationen erfasst, die aus der E-Mailadresse des Nutzers und einem so genannten *Privacy-Level* bestehen, der angibt, ob bei Bedarf der volle Name der Versuchsperson in der Diplomarbeit angegeben werden darf, nur die Initialen oder lediglich ein Pseudonym. Diese Daten werden in allen Dateien, die zum Austausch mit den Versuchspersonen dienen, wie z.B. Dateien mit Filmbewertungen oder Filmempfehlungen abgespeichert, wobei der Name des jeweiligen Nutzers mit in den Dateinamen aufgenommen wird [8].
Die Bewertungen von Filmen bzw. der Interessantheit von Filmempfehlungen werden vom Programm abgespeichert und können bei einem Neustart wieder eingeladen werden, so dass der Bewertungsprozess je nach verfügbarer Zeit des jeweiligen Nutzers in mehreren Schritten erfolgen kann. Zuallererst muss ein Nutzer dabei natürlich Filme bewerten, damit eine Basis von Informationen gegeben ist, aus der die Empfehlungsalgorithmen die Präferenzen eines Nutzers ermitteln können.

Bewertung von Filmen
Das *MovieVoter*-Programm präsentiert einem Nutzer wie in Abb. 4.1 gezeigt im Hauptfenster eine Liste der 3900 verwendeten Filme aus der *MovieLens*-Datenbank, wobei ein Eintrag der Liste dem oben angegebenen Format für einen Film m_i ohne die Liste ATL_i entspricht. Der Inhalt der Liste ATL_i für einen selektierten Film m_i wird in einer weiteren Liste im Fenster dargestellt. Neben manueller Navigation in der

werden. Diese modifizierten Filmtupel haben die Form $m_i = < \text{MLID}_i, \text{IMDBID}_i, \text{OT}_i, \text{Y}_i, \text{GL}_i, \text{T}_i, \text{ATL}_i >$, wobei IMDBID_i die eindeutige ID des Films in den Tabellen der *SQL*-Datenbank mit den aus der *IMDB* übernommenen inhaltlichen Informationen zu Filmen bezeichnet.

[8] Dadurch kann dieselbe *MovieVoter*-Installation auf einem Rechner von mehreren unterschiedlichen Personen benutzt werden.

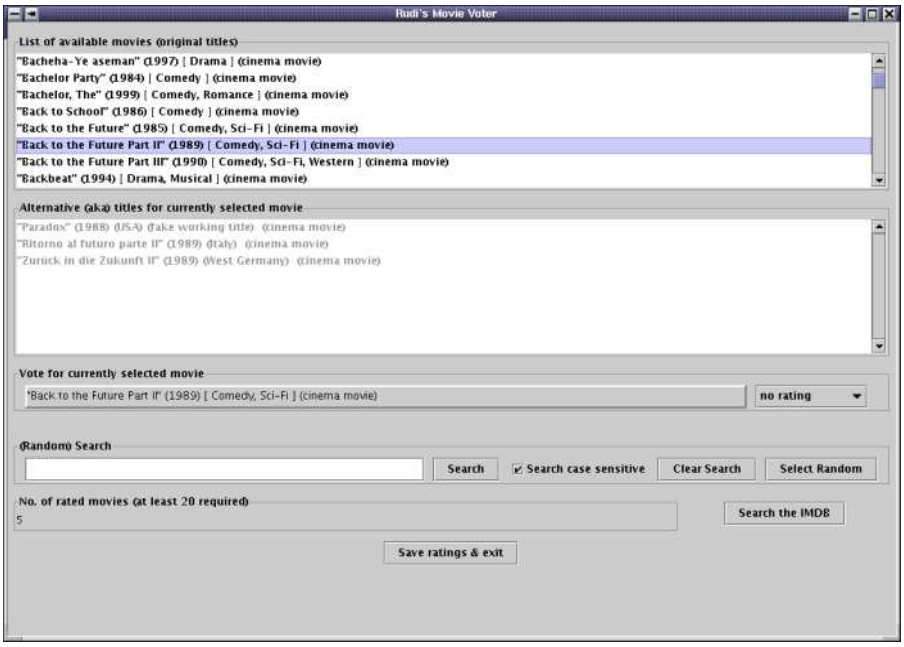


Abbildung 4.1: *MovieVoter*: Bewertung von Filmen

Liste der Original-Filmtitel bietet das Programm die Möglichkeit, Filme per Zufall oder durch Suche mit oder ohne Beachtung der Groß-/Kleinschreibung auszuwählen. Bei bestehender Internetverbindung können zu einem selektierten Filmtitel durch Betätigung eines speziellen Dialogknopfes zusätzliche Informationen in einem Internetbrowser angezeigt werden, wie z.B. in Abb. 4.2 gezeigt. Diese zusätzlichen Informationen wie Handlung eines Films, mitwirkende Schauspieler und Name des Regisseurs werden von der Webseite der Online-Version der *IMDB* bezogen.
Über eine Auswahlbox des Dialogs kann ein Nutzer einem angewählten Film m_i eine nominelle Bewertung r_i^n zuweisen, die intern auf eine reelle Bewertung r_i abgebildet wird[9]. Diese Abbildung wird in Tabelle 4.1 gezeigt.

r_i^n	r_i
excellent	5.0
good	4.0
average	3.0
bad	2.0
disastrous	1.0

Tabelle 4.1: *MovieVoter:* Bewertungsskala für Filme

Eine bereits getätigte Bewertung kann auch wieder gelöscht oder verändert werden. Bei Selektion eines bereits bewerteten Films in der Liste wird in der Auswahlbox die zuvor gewählte Bewertung angezeigt.
Bei Verlassen des Programms werden dem Nutzer eine Liste seiner getätigten Filmbewertungen sowie seine Nutzerdaten angezeigt und diese Daten anschließend in einer Datei abgespeichert. Damit den einzelnen Filmempfehlungsalgorithmen eine ausreichende Datenbasis zur Verfügung steht, um die Präferenzen eines

[9] Die reelle Bewertung entstammt dabei der in der *MovieLens*-Datenbank verwendeten Bewertungsskala.

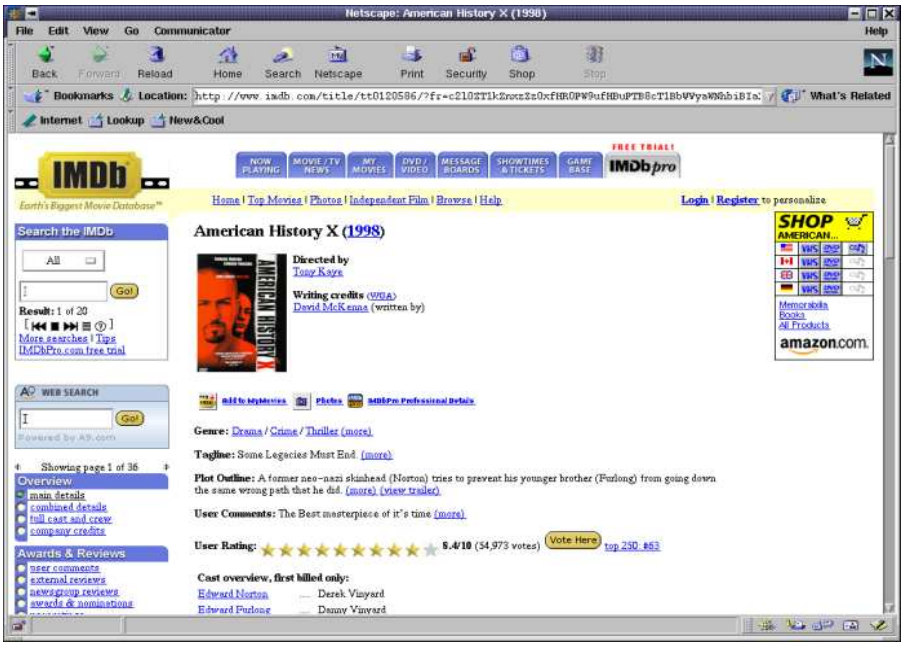


Abbildung 4.2: *MovieVoter*: Aufrufen von *IMDB*-Informationen - Beispiel

Nutzers zu ermitteln, muss jeder Nutzer mindestens 20 der 3900 Filme bewertet haben. Eine entsprechende Meldung wird bei Anzeige der Zusammenfassung ausgegeben. Die abgespeicherte Datei von Filmbewertungen dient als Eingabe für die verschiedenen Algorithmen, die aus den Filmbewertungen eine Datei von Filmempfehlungen erzeugen, die wiederum dem Nutzer zugeschickt wird.

Sichten der Empfehlungen

Eine Datei von Filmempfehlungen für einen bestimmten Nutzer kann ebenfalls mit Hilfe des *MovieVoter*-Programms sichtbar gemacht werden. Eine solche Datei enthält die *MovieLens*-Indizes und vorhergesagten Bewertungen all der Filme, die der Nutzer zuvor nicht bewertet hat[10]. Der Begriff „vorhergesagt“ bezieht sich dabei auf eine durch den jeweiligen Empfehlungsalgorithmus geschätzte reellwertige Bewertung eines Films, wie sie der Nutzer wahrscheinlich vorgenommen hätte. Diese Einschätzung basiert auf den durch die Filmbewertungen des Nutzers repräsentierten Präferenzen.

Die Bewertung orientiert sich dementsprechend auch an der Bewertungsskala von 1 bis 5 für Filme in dieser Versuchsumgebung, es ist jedoch möglich, dass einzelne vorhergesagte Bewertungen die untere Grenze der Skala unter- bzw. die obere Grenze der Skala überschreiten.

Der *MovieVoter* verwendet zur Anzeige der Filmempfehlungen ein Fenster, das zu dem bei der Filmbewertung benutzten annähernd identisch ist, wie Abb. 4.3 zeigt. Jedem Eintrag in der Liste der Original-Filmtitel ist jedoch zusätzlich in eckigen Klammern eine Zahl vorangestellt, die die Position der Filmempfehlung in der Gesamtliste angibt. Die Liste ist nach absteigenden vorhergesagten Bewertungen geordnet[11]. Ein Feld unterhalb der Liste mit alternativen Filmtiteln gibt die vorhergesagte Bewertung für den aktuell selektierten Film an. Such- und Auswahlmöglichkeiten entsprechen denen bei der Bewertung von Filmen. Angezeigte Empfehlungen können im *MovieVoter* auch direkt bzgl. ihrer Interessantheit bewertet werden.

[10] Hatte der Nutzer zuvor beispielsweise 100 von den 3900 Filmen bewertet, so enthält die Datei Indizes und vorhergesagte Bewertungen für 3800 Filme.

[11] Bei gleicher Bewertung werden Filme alphabetisch anhand der Titel und bei gleichen Titeln nach absteigenden Jahreszahlen sortiert.

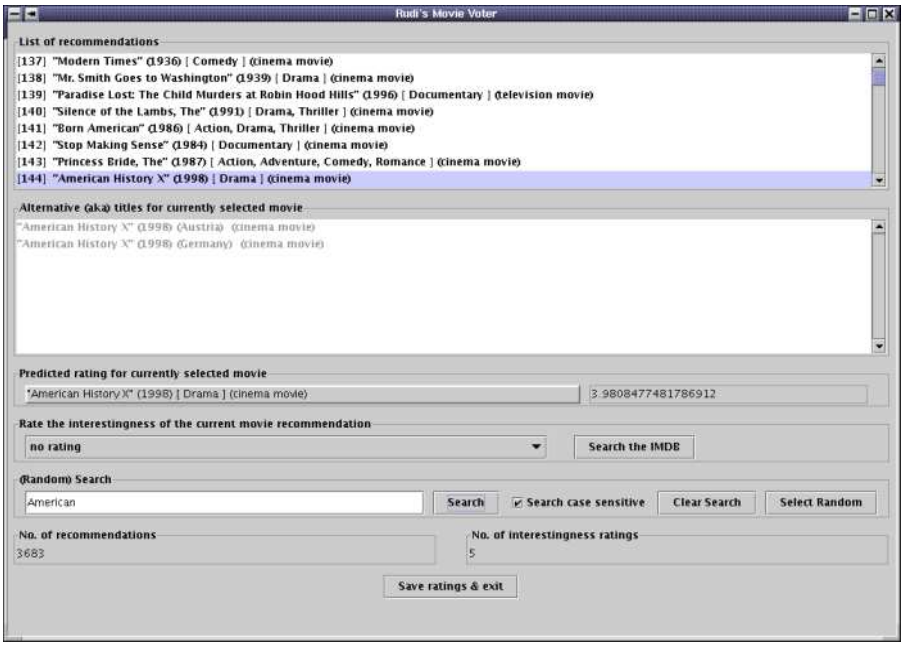


Abbildung 4.3: *MovieVoter*: Sichten und Bewerten von Empfehlungen

Bewerten der Interessantheit
In das Fenster zur Anzeige von Filmempfehlungen ist die Bewertung der Empfehlungen bzgl. ihrer Interessantheit integriert, wobei die Nutzerin die obersten 20 Empfehlungen der Liste bewerten muss. Zur Vereinfachung des Bewertungsprozesses kommt dem wie zuvor bei der Filmbewertung vorhandenen Dialogknopf zum Anzeigen von weiteren, aus der Online-*IMDB* extrahierten Zusatzinformationen zu einem selektierten Film eine besonders wichtige Bedeutung zu. Er gibt der Nutzerin die Wahlfreiheit, zu entscheiden ob und wenn ja wieviele der ersten 20 empfohlenen Filme sie sich ansehen möchte. Anstatt sich erst einen Film ansehen zu müssen, um auf dieser Basis die Interessantheit der Empfehlung dieses Films bewerten zu können, kann sie alternativ auch auf die Informationen der Online-*IMDB* zurückgreifen. Informationen wie z.B. die Handlung eines Films oder Kritiken von Personen, die den entsprechenden Film bereits gesehen haben, versetzen die meisten Menschen in die Lage, abschätzen zu können, ob ihnen ein Film gefallen könnte oder nicht.
Zur letztendlichen Bewertung der Interessantheit von Empfehlungen dient auch hier eine Auswahlbox, wobei die genaue Skala im Folgenden Abschnitt 4.7 genau beschrieben wird.
Bei Verlassen des Programms wird auch hier eine Zusammenfassung der getätigten Interessantheitsbewertungen samt Nutzerdaten angezeigt und die Bewertungen werden in einer separaten Datei abgespeichert.

4.5 YALE

Um den Lernprozess und die Auswertung der verschiedenen Empfehlungsalgorithmen zu vereinfachen, wurde die am Lehrstuhl entwickelte Lernumgebung *YALE*[12] verwendet, die es auf einfache Weise ermöglicht, das Lernen in kleine, in sich geschlossene Schritte - *Operatoren* genannt - aufzuteilen. So kann auf eine Vielzahl bereits vorhandener Operatoren zum Lernen, zur Optimierung und Evaluierung zurückgegriffen werden. Operatoren können beliebig ineinander verschachtelt und der Prozess des Lernens durch eine ebenfalls vorhandene in Abb. 4.4 gezeigte GUI anschaulich verfolgt werden. Durch die Implementierung in der Programmiersprache *Java* ist zudem die Nutzbarkeit des *YALE*-Pakets auf verschiedensten Rechner-

[12] Siehe [RITTHOFF et al. 2001] und [FISCHER et al. 2002].

architekturen gegeben.
Um die Vorteile von *YALE* nutzen zu können, wurden die für diese Arbeit verwendeten Empfehlungsalgorithmen sowohl als „stand alone“-Programme, als auch als Operatoren für *YALE* implementiert. So konnte *YALE* unter anderem zum protokollierten Lernen, zur Parameteroptimierung und zur Evaluierung der Vorhersagegenauigkeit genutzt werden.
Außerdem wurde so die Wiederverwendbarkeit der implementierten Algorithmen, die nachfolgend beschrieben werden, in anderen Untersuchungen gewährleistet.

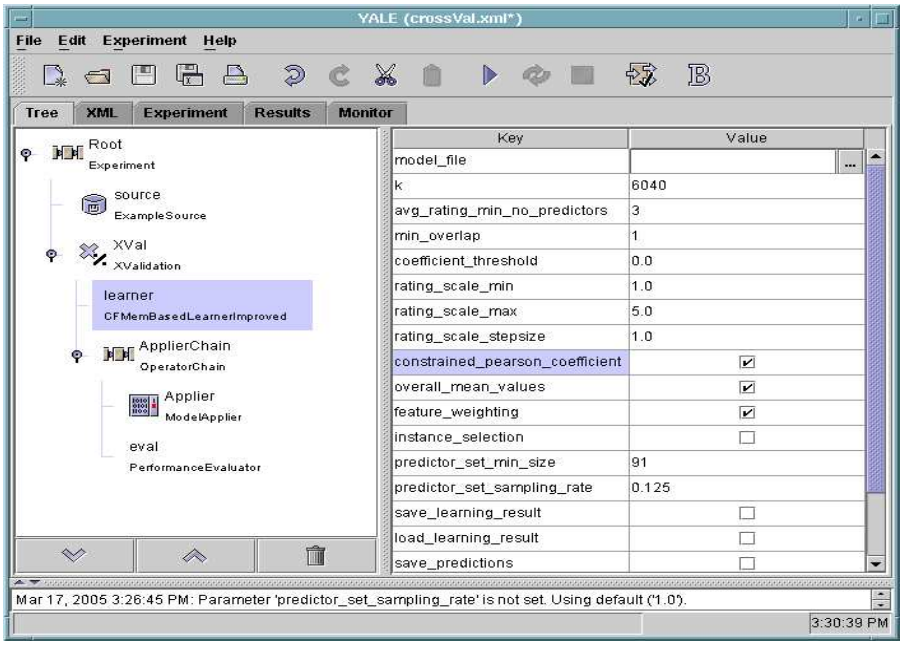


Abbildung 4.4: *YALE*: Evaluierung des kollaborativen Empfehlungsalgorithmus

4.6 Verwendete Typen von Empfehlungssystemen

Im Rahmen dieser Diplomarbeit werden zwei verschiedene Verfahren aus dem Bereich der Empfehlungssysteme implementiert und bzgl. der Interessantheit der von ihnen gelieferten Filmempfehlungen untersucht. Diese Verfahren sind:

1. Ein kollaboratives Empfehlungsverfahren, wie in Kapitel 5 beschrieben, wird besonders intensiv untersucht, da es sich um ein „state-of-the-art“-Verfahren handelt, dass auf besonders gute Vorhersagegenauigkeit hin entwickelt wurde. Sollte sich die angestrebte gute Vorhersagequalität in den „Labortests“ bestätigen, kann untersucht werden, ob sich die praktische Nützlichkeit - in dieser Arbeit gemessen über das eigens definierte Interessantheitsmaß - konform zu der Vorhersagequalität verhält oder nicht.

2. Zur potenziellen Verbesserung des kollaborativen Verfahrens und als Vergleich zu dessen Fähigkeiten und Einschränkungen zieht das zweite, eigenschaftsbasierte Verfahren in Kapitel 6 zusätzlich inhaltliche Kriterien der von den Nutzern bewerteten Filme als Eigenschaften dieser Filme heran. Es stellt somit eine Hybridisierung aus einem kollaborativen und einem eigenschaftsorientierten Verfahren dar und erlaubt Aussagen darüber, ob Hybridsysteme in Hinsicht auf das verwendete praxisorientierte Interessantheitsmaß Verbesserungen gegenüber der alleinigen Nutzung von kollaborativen Verfahren

bewirken können. Zudem handelt es sich bei der Kombination kollaborativ-eigenschaftsorientiert um eine der meist untersuchten bzgl. Hybridsystemen (vgl. auch Kapitel 6.2).

4.7 Bewertungsmaße

In Kapitel 3 wurden umfangreiche Überlegungen bezogen auf die Qualitätsbewertung von Empfehlungssystemen angestellt. Maße zur Ermittlung der hypothetischen Genauigkeit von Verfahren wurden ebenso untersucht, wie praxisorientiertere Alternativen. In Abschnitt 3.3 wurde ein eigenes praxisbezogenes Maß, das Maß der „Interessantheit“ definiert und die Definition auf eine notwendige und eine hinreichende Bedingung aufgebaut. Notwendige Bedingung ist dabei die hypothetische Genauigkeit eines Verfahrens. Diese hypothetische Genauigkeit soll vor dem praktischen Einsatz gemessen und dann von den Versuchspersonen im Rahmen der Interessantheitsbewertung bestätigt oder verneint werden. Hier sollen nun die genauen Bewertungsmaße der Genauigkeit und Interessantheit vorgestellt werden, wie sie in der praktischen Testumgebung zum Einsatz kommen.

Genauigkeit
Zahlreiche Maße zur Bestimmung der Genauigkeit sind bezogen auf Empfehlungssysteme im Einsatz. Dabei ist die Wahl des korrekten Maßes aus dieser umfangreichen Gesamtmenge von zahlreichen Bedingungen abhängig. Eine dieser Bedingungen ist dabei die vom Empfehlungssystem benutzte Bewertungsskala. In dieser Arbeit werden, wie in Abschnitt 4.6 beschrieben, zwei verschiedene Empfehlungssysteme implementiert und unter anderem auf ihre Genauigkeit hin untersucht - ein kollaboratives Verfahren in Kapitel 5 und ein eigenschaftsbasierter Ansatz in Kapitel 6, der zusammen mit dem kollaborativen Verfahren ein Hybridsystem bildet. Das verwendete kollaborative Verfahren nach [YU et al. 2003] wurde in Hinsicht auf eine möglichst genaue Vorhersage der konkreten Bewertung eines Objekts entwickelt. Um die Genauigkeit des Verfahrens bezogen auf dieses Ziel zu messen, bedarf es also eines Maßes zur Messung der Vorhersagegenauigkeit. Der eigenschaftsbasierte Ansatz hingegen dient zur nachträglichen Filterung der vom kollaborativen Verfahren erstellten Empfehlungsliste, um die Interessantheit von Empfehlungen zu verbessern. Er wurde daraufhin optimiert, möglichst viele der von den Nutzern im praktischen Versuch als interessant bewerteten Empfehlungen des kollaborativen Systems bei der Filterung beizubehalten und möglichst viele der als nicht interessant bzw. ungültig bewerteten Empfehlungen herauszufiltern. Für diesen Filterprozess wird ein Maß benötigt, das die Genauigkeit bezogen auf diese binäre Unterteilung von Empfehlungen ermitteln kann, also ein Maß aus der Gruppe Klassifikationsgenauigkeit.
Als Vertreter aus der Gruppe der Maße für die Vorhersagegenauigkeit wird der *MAE*, der mittlere absolute Fehler gewählt. Der *MAE* ist das hypothetische Bewertungsmaß, welches im Bereich der Empfehlungssysteme am häufigsten eingesetzt wird (siehe [SARWAR et al. 2001]). Da in dieser Arbeit der Zusammenhang zwischen hypothetischen Bewertungsmaßen und praxisorientierten Maßen in Form der in Kapitel 3.3 definierten Interessantheit untersucht werden soll, eignet sich der *MAE* wegen der häufigen Benutzung besonders gut, um später allgemeine Aussagen zu treffen. Außerdem lässt sich eine mit dem *MAE* gemessene Genauigkeit aufgrund der Definition des *MAE* s sehr leicht interpretieren.
Zur Bewertung der Klassifikationsgenauigkeit stehen verschiedene Maße wie *Precision*, *Recall* oder das *ROC*-Maß zur Auswahl. Aufgrund besonderer Eigenschaften, die sich während einiger Optimierungstests des eigenschaftsbasierten Algorithmus mit einigen dieser Standardmaße herauskristallisierten und die in Kapitel 6 genau beschrieben werden, wurde auf die Standarmaße für die Klassifikationsgenauigkeit verzichtet und stattdessen ein auf den Standardmaßen basierendes und hinsichtlich den genannten besonderen Eigenschaften modifiziertes eigenes Qualitätsmaß verwendet, dass ebenfalls in Kapitel 6 vorgestellt wird.

Interessantheit
In Kapitel 3.3 wurde der Begriff der „Interessantheit“ von Empfehlungen formal definiert. Hier soll nun diese Definition in eine konkrete Bewertung für die praktischen Versuche umgesetzt werden. Entsprechend der Definition wird eine dreiwertige Skala für die Interessantheit gewählt, die die Erfüllung der notwen-

digen bzw. hinreichenden Bedingung für das Vorhandensein einer interessanten Empfehlung widerspiegelt. Der Anwenderin werden dabei die möglichen Interessantheitsbewertungen durch das implementierte *MovieVoter*-Programm (Abschnitt 4.4) in nominaler, verständlicher Form präsentiert, während diese nominalen Werte für interne Berechnungen und die letztendliche Auswertung des Versuchsergebnisses auf reelle Werte abgebildet werden. Über den *Search the IMDB*-Button des *MovieVoters*, nach dessen Betätigung zur aktuell angewählten Empfehlung r_i^u Informationen wie Handlung, mitwirkende Schauspieler oder Regisseure abgerufen werden können, soll die Nutzerin u in die Lage versetzt werden, sowohl ihre eigene Bewertung $s_u(r_i^u)$, als auch die Unerwartetheit und Nützlichkeit $serendipity(r_i^u)$ der Empfehlung abzuschätzen.

- *I don't agree with the prediction* $\rightarrow 1.0$
  Die notwendige Bedingung für das Vorhandensein einer interessanten Empfehlung ist nicht erfüllt, d.h. entsprechend der Definition in Kapitel 3.3 gilt $s_{RS}(r_i^u) \not\approx s_u(r_i^u)$.
  Bei der Auswertung der Versuche hinsichtlich der Interessantheit werden solchermaßen bewertete Empfehlungen als auch „ungültig" bezeichnet.

- *I agree with the prediction, but don't find it interesting* $\rightarrow 3.0$
  Die notwendige Bedingung für die Existenz einer interessanten Empfehlung ist zwar erfüllt, die hinreichende Bedingung wird jedoch verletzt, d.h. $s_{RS}(r_i^u) \approx s_u(r_i^u)$, aber $serendipity(r_i^u) = 0$.
  In den späteren Versuchsauswertungen wird auf mit diesem Wert versehene Empfehlungen mittels des Begriffs „uninteressant" Bezug genommen.

- *I agree with the prediction and find it interesting* $\rightarrow 5.0$
  Notwendige und hinreichende Bedingung für das Auftreten einer interessanten Empfehlung sind erfüllt: $s_{RS}(r_i^u) \approx s_u(r_i^u) \wedge serendipity(r_i^u) = 1$.
  Empfehlungen mit dieser Bewertung werden später gemäß der Definition des Interessantheitsmaßes als „interessant" bezeichnet.

5 Kollaboratives Empfehlungssystem: IBL

> *„Look, I've really enjoyed our collaboration. I... I feel our intellects and approaches really compliment each other, and I was, you know, hoping you felt the same way."*
>
> Luke, *Joan and Arcadia*, 2003

Menschen sind nach [McNee et al. 2002] grundlegend soziale Geschöpfe. In Interaktion bilden sie untereinander sogenannte "soziale Netzwerke", die im relativ neuen, von dem Anthropologen Radcliffe-Brown in den 30er Jahren des letzten Jahrhunderts ins Leben gerufenen Forschungszweig der Analyse sozialer Netzwerke intensiv untersucht werden.[1] Kollaborative Empfehlungsverfahren basieren auf dem Prinzip sozialer Netzwerke und haben ein natürliches Verhalten von Menschen operationalisiert.
Wenn Menschen im Leben zwischen mehreren Möglichkeiten wählen müssen, ohne eine eigene Erfahrung bzgl. aller Alternativen vorweisen zu können, greifen sie nach [Resnick und Varian 1997] häufig auf direkte Empfehlungen von Freunden und Bekannten zurück, aber auch auf indirekte Empfehlungen wie Mundpropaganda, Empfehlungsschreiben, Restaurantführer oder Buchbesprechungen. Einzige Voraussetzungen für die Rolle eines Empfehlenden sind dabei nach [Terveen und Hill 2001], dass die entsprechende Person/Institution die Wahlmöglichkeiten kennt, die dem Empfehlung Suchenden unbekannt sind, dass sie bei Empfehlungen in der Vergangenheit hilfreich war, dass deren Meinung vom Empfehlung Suchenden geschätzt wird oder sie als anerkannter Experte auf dem betrachteten Gebiet gilt.
Dabei stellen kollaborative Verfahren das soziale Netzwerk meist als Nutzer-Objekt-Matrix dar, in der Matrixeinträge den Bewertungen der Nutzer für die jeweiligen Objekte entsprechen. In Abb. 5.1 ist ein einfaches Beispiel für solch eine Matrix gezeigt. Die Zeilen dieser Matrix entsprechen einzelnen Nutzern bzw.

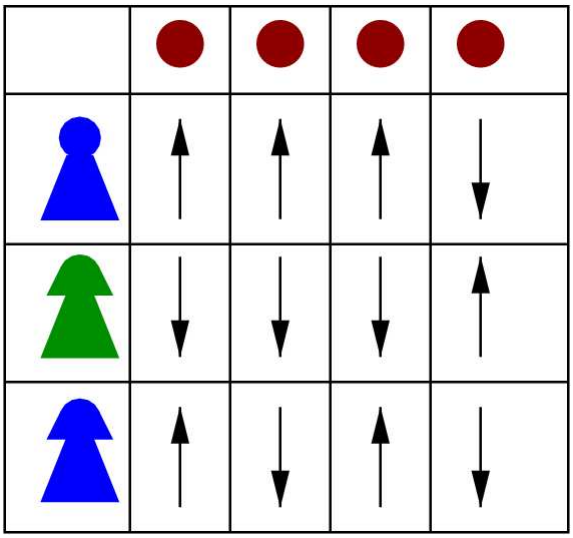


Abbildung 5.1: Nutzer-Objekt-Matrix

Nutzerinnen, die Spalten repräsentieren einzelne Objekte. Die Matrixeinträge bestehen in dem Beispiel entweder aus einem nach oben gerichteten Pfeil der angibt, dass das Objekt von dem Nutzer bzw. der Nutzerin

[1] Siehe [Scott 2000].

als gut bewertet wurde oder einem nach unten gerichteten Pfeil der entsprechend für eine negative Bewertung steht. Der männliche Nutzer in Zeile 1 der Beispielmatrix hat bzgl. seines Geschmacks gar keine Ähnlichkeit zu der grün gefärbten Nutzerin in Zeile 2, vielmehr widersprechen sich ihre Präferenzen. Die andere Nutzerin in Zeile 3 hingegen[2] hat einen sehr ähnlichen Geschmack wie der Nutzer, nur bezogen auf das zweite Objekt von links, ist sie anderer Meinung als er. Die Beispielmatrix ist dabei insofern ideal, dass der Nutzer und die beiden Nutzerinnen alle Objekte in der Matrix kennen und sie somit auch bewerten können. In der Realität ist solch eine Matrix wesentlich größer und weist viele leere Einträge auf, was es den kollaborativen Empfehlungssystemen erschwert, Ähnlichkeiten zwischen Nutzern festzustellen und somit Empfehlungen zu generieren.
Auch ist in realen Anwendungen die Größe solch einer Matrix ständigen Schwankungen unterworfen, wenn Objekte oder Nutzer wegfallen oder hinzukommen. Meistens ist Letzteres der Fall, was insbesondere an der Globalisierung und den Fortschritten in der Kommunikationstechnik liegt, wobei das Internet besondere Erwähnung verdient. Durch diese Faktoren lassen sich heutzutage viel größere soziale Netzwerke bilden, als dies früher möglich gewesen wäre.
Kollaborative Empfehlungsverfahren nutzen dabei die Erfahrung solch großer Gemeinschaften voll aus und ersparen einem Rat Suchenden die Zeit, die normalerweise zum Aufbau eines sozialen Netzwerks nötig wäre. Dadurch ergeben sich auch aus soziologischer Sicht völlig neue Entwicklungen. So fragte man sich in den Anfangstagen der erwähnten Analyse sozialer Netzwerke bspw., ob Gemeinschaften („communities“), wie sie in kleinen ländlichen Dörfern beobachtet wurden, der schon damals zu beobachtenden Stadtflucht und Technisierung zum Opfer fallen würden. Heute wird im Zusammenhang mit kollaborativen Empfehlungsverfahren[3] ganz natürlich von „communities“ und „tribals“ gesprochen;[4] diese Gesellschaftsformen als Ausdruck grundlegender Entwicklungsstufen in der menschlichen Evolution haben sich somit den modernen Zeiten angepasst.
Als logische Folge dieser Entwicklungen scheinen kollaborative Verfahren die idealen Kandidaten für Empfehlungssysteme zu sein und sie finden in Forschung und Praxis auch großen Anklang. Hier soll die Qualität solcher Systeme sowohl mittels hypothetischer Bewertungsmaße, als auch durch das in Abschnitt 3.3 definierte praktische Maß der Interessantheit untersucht werden. Da in den praktischen Versuchen ein einzelnes kollaboratives Verfahren verwendet wurde, folgt eine Begründung für die Auswahl dieses speziellen Algorithmus. Die theoretischen Grundlagen des verwendeten Algorithmus schließen sich an, gefolgt von der Implementierung, inklusive der dabei aufgetretenen Probleme und deren Behebung.
Den Abschluss bilden die Ergebnisse des Versuchs, insbesondere im Bezug auf die Interessantheit von generierten Filmempfehlungen, die weitere Probleme im praktischen Einsatz sichtbar machen.

Auswahlkriterien

Die Wahl eines kollaborativen Empfehlungsalgorithmus fiel auf die von [Yu et al. 2003] beschriebene Implementierung eines kollaborativen Algorithmus mit Eigenschaftsgewichtung und Instanzenselektion.
Diese Wahl wurde aus mehreren Gründen getroffen:

- Es handelt sich um einen „state-of-the-art“-Algorithmus, der in Untersuchungen[5] sehr gute Ergebnisse geliefert hat.
- Die im Algorithmus verwendeten Verbesserungen durch Eigenschaftsgewichtung und Instanzenselektion können auf modulare Weise implementiert und so bei Bedarf ein- oder ausgeschaltet werden. Somit kann der Algorithmus auch als „normales“ kollaboratives Verfahren ohne Optimierungen betrieben und die Ergebnisse der verbesserten Version mit denen der Standardversion verglichen werden.
- Der Algorithmus wurde auf der *EachMovie*-Datenbank und somit auch im Bereich der Empfehlung

[2] Die, um die Ähnlichkeit zum Nutzer zu zeigen ebenfalls blau gefärbt ist.
[3] Und auch in anderen das Internet betreffenden Bereichen, wie z.B. Foren und Online-Spielplattformen.
[4] Wobei die Matrix in Abb. 5.1 eine Mini-community darstellt.
[5] Siehe [Yu et al. 2003].

von Filmen getestet. Da die hier verwendete *MovieLens*-Datenbank aus der *EachMovie*-Datenbank entstanden ist und beide Datenbanken dieselbe Bewertungsskala benutzen, sind Vergleiche zwischen den in [YU et al. 2003] publizierten Ergebnissen und eigenen Ergebnissen im begrenzten Maß möglich.

Theoretische Grundlagen

Der Algorithmus von [YU et al. 2003] bietet eine Lösung der zwei grundlegenden Probleme kollaborativer Filteralgorithmen - der Skalierbarkeit und Genauigkeit - indem Gewichtung von Eigenschaften und Selektion von Instanzen angewendet wird.
Diese zwei Methoden finden ursprünglich in der Klasse sogenannter *IBL*-Algorithmen (*Instance-Based Learning*, instanzenbasiertes Lernen) Verwendung. *IBL*-Verfahren[6] sind Ableger der *Nearest Neighbor*-Methoden[7]. Bei k-*Nearest-Neighbor*-Methoden wird eine neue Instanz genauso klassifiziert, wie die bzgl. eines zuvor gewählten Ähnlichkeits- oder Distanzmaßes ähnlichste/nächste Instanz. Werden $k > 1$ Nachbarn betrachtet, so gilt bezogen auf die Klassifikation das „Mehrheitsurteil“. *Nearest-Neighbor*-Verfahren fassen Daten zusammen, arbeiten nicht inkrementell und haben als Ziel die perfekte Konsistenz mit den Trainingsinstanzen, wobei sie neue Instanzen ignorieren, was sie anfällig für Rauschen macht.[8] Im Gegensatz dazu arbeiten *IBL*-Systeme inkrementell, indem sie alle bzw. alle für eine korrekte Klassifikation benötigten Trainingsbeispiele speichern. Die Ähnlichkeits- bzw. Distanzberechnung ist dabei wegen des inkrementellen Verfahrens stets aktuell. Somit verfolgen *IBL*-Verfahren das Ziel, die Klassifikationsgenauigkeit für alle im Folgenden präsentierten Instanzen zu maximieren. Sie tolerieren damit Rauschen und können zusätzlich auch mit fehlenden Attributwerten der Instanzen umgehen.[9] Die Genauigkeit der von den *IBL*-Verfahren erzeugten Generalisierung der Daten hängt dabei stark von der verwendeten Distanz-/Ähnlichkeitsfunktion ab. Um diese Genauigkeit zu steigern, wird die Distanzfunktion oft mit Gewichtungen der Eigenschaften von Instanzen parametrisiert.
Da zur Berechnung der Distanz zwischen einer neuen Instanz und den gespeicherten Instanzen bei insgesamt n Instanzen ein Aufwand von $O(n^2)$ entsteht, werden für *IBL*-Algorithmen außerdem Methoden verwendet, die eine Auswahl der geeignetsten Instanzen für die Generalisierung erlauben, so dass die Komplexität der Rechenzeit und des Speicherplatzes in Grenzen gehalten werden kann.
Für den benutzten kollaborativen Algorithmus werden die Eigenschaftsgewichtung und Instanzenselektion aus dem *IBL*-Bereich auf das Bewerten von Objekten übertragen. Im Zusammenhang mit der Vorhersage einer Bewertung $v_{A,i}$ für einen Nutzer A und ein Zielobjekt i gelten alle Objektbewertungen $v_{B,j}$ mit $j \neq i$ als Eigenschaften einer Nutzers B, d.h. alle Bewertungen dieses Nutzers für Objekte außer dem Zielobjekt i, wobei als Voraussetzung gilt, dass Nutzer B auch eine Bewertung $v_{B,i}$ für das Zielobjekt i abgegeben hat. Nutzer wiederum werden in diesem Kontext als Instanzen aufgefasst.
Auf Basis dieser Interpretation nutzt der Algorithmus Maße für die Relevanz von Eigenschaften (Objektbewertungen) und Instanzen (Nutzern), die aus Untersuchungen in einem einheitlichen informationstheoretischen Rahmen entstanden sind.

1. *Eigenschaftsgewichtung*
   Für die Gewichtung von Bewertungen werden wechselseitige Abhängigkeiten zwischen jeweils zwei Bewertungen betrachtet. Sind beispielsweise die Bewertungen annähernd gleich verteilt wie in Abb. 5.2, so werden Bewertungen für das Objekt j zur Vorhersage der Bewertung von Objekt i mit einem niedrigen Gewicht versehen. Gibt es jedoch klare Abhängigkeiten zwischen Bewertungen für Objekt i und j, dann werden Bewertungen von Objekt j mit einem hohen Gewicht versehen, wenn sie zur Vorhersage der Bewertung für Objekt i herangezogen werden. Solch eine Abhängigkeit zeigt Abb. 5.3, wo Nutzer, die das Objekt j als schlecht bewertet haben, dem Objekt i meist eine gute Bewertung

[6] Siehe z.B. [AHA et al. 1991].
[7] Vgl. z.B. [COVER und HART 1967], [HART 1968] oder [GATES 1972].
[8] Siehe [AHA et al. 1991].
[9] Siehe ebenfalls [AHA et al. 1991].

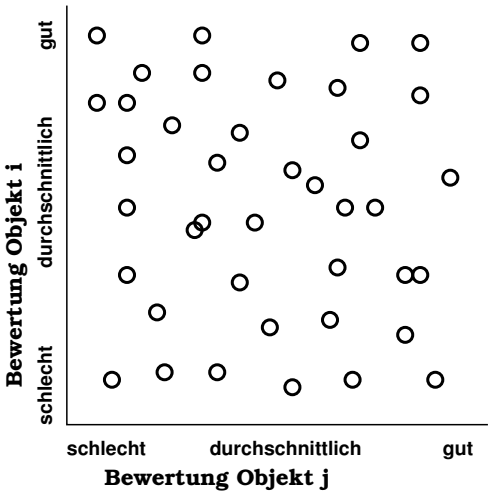


Abbildung 5.2: Unkorrelierte Objektbewertungen

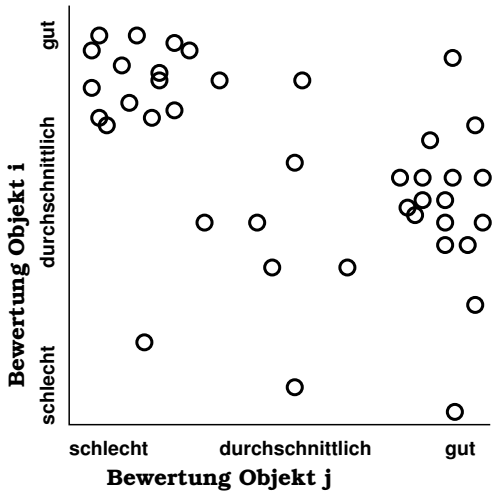


Abbildung 5.3: Korrelierte Objektbewertungen

zugewiesen haben und Nutzer, die Objekt j mit einer guten Bewertung versehen haben, gleichzeitig Objekt i als durchschnittlich eingestuft haben.

Formal kann die Abhängigkeit eines Objekts i von einem Objekt j nach [YU et al. 2003] ausgedrückt werden als:

$$p(|v_{A,i} - v_{B,i}| < e \mid |v_{A,j} - v_{B,j}| < e) \tag{5.1}$$

wobei A und B beliebige Nutzer und e ein Schwellwert für eine große Abhängigkeit zwischen den Bewertungen ist.
Übertragen in einen informationstheoretischen Rahmen kann Shannons Konzept der „mutual information“ benutzt werden, das die statistische Abhängigkeit zwischen zwei Zufallsereignissen X und Y mit Zufallsverteilungen $p(x)$ bzw. $p(y)$ repräsentiert:

$$I(X;Y) = \sum_x \sum_y p(x,y) log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{5.2}$$

Dies ist äquivalent zur Formel:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{5.3}$$

in der $H(X)$ und $H(Y)$ die Entropie von X bzw. Y und $H(X,Y)$ die Joint-Entropie darstellen, die gegeben sind als:

$$H(V_i) = -\sum_{k=1}^{N} p(v_i = k)\ log_2\ p(v_i = k) \tag{5.4}$$

$$H(V_j) = -\sum_{k=1}^{N} p(v_j = k)\ log_2\ p(v_j = k) \tag{5.5}$$

$$H(V_i;V_j) = -\sum_{k=1}^{N} \sum_{l=1}^{N} p(v_j = l, v_i = k)\ log_2\ p(v_j = l, v_i = k) \tag{5.6}$$

k und l sind dabei mögliche Bewertungen, d.h. bezogen auf unsere in Kapitel 4 definierte Versuchsumgebung sind $k, l \in \{1, 2, 3, 4, 5\}$ und $N = 5$.

[YU et al. 2003] haben nun außerdem gezeigt, dass die Definition der mutual information äquivalent zur probabilistischen Definition der wechselseitigen Abhängigkeit ist, wenn die benutzte Bewertungsskala diskret ist. Dies ist in unserer Versuchsumgebung der Fall.
Bevor dieses Wissen konkret zur Vorhersage von Bewertungen genutzt werden kann, bleibt noch das Problem der Abschätzung der Wahrscheinlichkeitsverteilungen $p(\dots)$ zu lösen. Hier kommt erschwerend hinzu, dass die zur Empfehlung benutzten Daten oft große Lücken aufweisen, da die meisten Nutzer nur einen kleinen Teil der zur Verfügung stehenden Objektmenge bewerten. Zur Lösung dieses Problems benutzen [YU et al. 2003] den Bayes'schen Ansatz der m-Abschätzung:

$$p = \frac{r + m \cdot P}{n + m} \tag{5.7}$$

Dabei tritt ein bzgl. der Anzahl von Ausprägungen begrenztes Zufallsereignis in n Fällen r-mal auf. P bezeichnet die a-priori Wahrscheinlichkeit des Ereignisses und m ist eine Konstante, die die Gewichtung von P relativ zu den beobachteten Ausprägungen des Ereignisses angibt. Meist wird der beste Wert für m empirisch ermittelt, bei großen Datenmengen wie z.B. in unseren Experimenten ist diese Vorgehensweise jedoch nicht praktikabel. [YU et al. 2003] verwenden deshalb der Einfachheit halber den Wert $m = \sqrt{n}$.
Damit können die Wahrscheinlichkeiten wie folgt berechnet werden:

$$p(v_i = k) = \frac{r_i^k + \sqrt{n_i} \cdot P(v = k)}{n_i + \sqrt{n_i}} \tag{5.8}$$

$$p(v_j = l) = \frac{r_j^l + \sqrt{n_j} \cdot P(v = k)}{n_j + \sqrt{n_j}} \tag{5.9}$$

$$p(v_j = l, v_i = k) = \frac{r_{i,j}^{k,l} + \sqrt{n_{i,j}} \cdot p(v_i = k) \cdot p(v_j = l)}{n_{i,j} + \sqrt{n_{i,j}}} \tag{5.10}$$

n_i bezeichnet dabei die Anzahl von Nutzern, die Bewertungen für Objekt i abgegeben haben und r_i^k die Anzahl von Nutzern, die Objekt i mit dem Wert k bewertet haben. Die a-priori Wahrscheinlichkeit $P(v = k)$ kann direkt aus der Menge aller vorliegenden Bewertungen unabhängig vom bewerteten Objekt berechnet werden. Entsprechend repräsentiert $n_{i,j}$ die Anzahl von Nutzern die sowohl Objekt i als auch Objekt j bewertet haben und $r_{i,j}^{k,l}$ die Zahl von Nutzern, die Objekt i mit dem Wert k und Objekt j mit dem Wert l bewertet haben. Die a-priori-Wahrscheinlichkeit für 5.10 wird unter der Annahme berechnet, dass Bewertungen von zwei Objekten voneinander unabhängig sind.

2. *Selektion relevanter Instanzen*
   Das standardmäßige Vorgehen kollaborativer Algorithmen bei der Vorhersage von Bewertungen für einen bestimmten Zielnutzer und ein Zielobjekt ist die Berechnung eines Distanzmaßes zwischen dem Zielnutzer und anderen Vergleichsnutzern (Empfehlenden), wobei die für das Distanzmaß verwendeten Daten den Bewertungen beider Nutzer für die Menge von gemeinsam bewerteten Objekten entsprechen. Jene Empfehlenden, deren Distanz zum Zielnutzer einer bestimmten Bedingung, meist einem Schwellwert genügt, werden für die Vorhersage der Bewertung für das Zielobjekt benutzt, indem aus ihren Bewertungen das gewichtete Mittel gebildet wird. Somit ist der rechnerische Aufwand für eine Vorhersage linear zu der Anzahl benutzter Empfehlender.
   Bei stetig wachsenden Systemen läuft man hier schnell in ein Skalierungsproblem. Um die Berechnungs- und damit die Antwortzeiten für den Nutzer erträglich zu halten, ist es deshalb notwendig, die Anzahl der für eine Vorhersage verwendeten Nutzer zu beschränken, ohne dass die Qualität der Vorhersagen unter dieser Einschränkung zu sehr leidet.

a) *Rationalität von Instanzen*
[YU et al. 2003] führen zu diesem Zweck den Begriff der *Rationalität* (rationality) ein, der auf der Idee beruht, dass für die Vorhersage einer Bewertung für einen Zielnutzer a und ein Zielobjekt i nur solche Nutzer b herangezogen werden sollten, deren Bewertungen für Objekte $j \neq i$ genug Informationen dafür liefern, warum Nutzer b das Zielobjekt i mit dem vorliegenden Wert bewertet hat.
Formal ausgedrückt betrachten wir im Folgenden Nutzer u aus der Trainingsmenge $\mathcal{T}_i$, d.h. der Menge von Nutzern, die Bewertungen für das Zielobjekt i abgegeben haben. u wird dabei auch als *Instanz bezogen auf das Zielobjekt* i bezeichnet, während man die Bewertung von u für das Zielobjekt i den *Wert* $v_{u,i}$ *der Instanz* nennt. Als Notation für die Menge der Objekte $j \neq i$, die von Nutzer u bewertet wurden, die sog. *Menge der Instanzeneigenschaften bezogen auf das Zielobjekt* i wählt man $\overline{\mathcal{T}}_{u,i}$ und die Bewertungen $d_{u,i}$ von Nutzer u für diese Menge von Objekten bezeichnet man als *Instanzenbeschreibung bezogen auf das Zielobjekt* i.
Mit diesen Notationen kann man das grundlegende Maß für die hier verwendete Instanzenselektion definieren als:

Definition 1. *Sei eine Instanz $u \in \mathcal{T}_i$, repräsentiert durch ihre Beschreibung $d_{u,i}$ über ihrer Menge von Instanzeneigenschaften $\overline{\mathcal{T}}_{u,i}$, sowie der Wert $v_{u,i}$ der Instanz gegeben.*
Die Rationalität der Instanz u bezogen auf das Zielobjekt i*, bezeichnet mit $R_{u,i}$, ist die Reduktion der Unsicherheit des Instanzenwertes $v_{u,i}$ bei gegebenem Wissen in Form der Beschreibung $d_{u,i}$, die ausgedrückt werden kann als:*

$$\begin{aligned} R_{u,i} &= H(v_i = v_{u,i}) - H(v_i = v_{u,i} | v_{u,\overline{\mathcal{T}}_{u,i}} = d_{u,i}) \\ &= -log_2 p(v_i = v_{u,i}) + log_2 p(v_i = v_{u,i} | v_{u,Fu,i} = d_{u,i}) \end{aligned} \tag{5.11}$$

wobei $H(v_i = v_{u,i})$ die a-priori und $H(v_i = v_{u,i} | v_{u,\overline{\mathcal{T}}_{u,i}} = d_{u,i})$ die a-posteriori Unsicherheit des Instanzenwertes $v_{u,i}$ sind.

b) *Wirksamkeit der Rationalität*
Von der Idee her scheint die Rationalität einer Instanz, wie sie oben definiert ist, geeignet zu sein, um unwichtige von wichtigen Instanzen zu trennen und so eine Skalierbarkeit der Vorhersageberechnungen auch für größere Mengen von Nutzern in der Datenbasis zu gewährleisten.
Doch wie kann die Wirksamkeit dieses Maßes formal begründet werden?
Dazu ist es sinnvoll, das ganze Grundproblem der Vorhersage von Bewertungen für ein bestimmtes Zielobjekt i aus der Sicht des Bayes'schen Lernens zu betrachten.
Der Lerner in unserer kollaborativen Umgebung kann sich auf eine Menge H_i von möglichen Hypothesen mit a-priori Wahrscheinlichkeiten $p(h_i)$ für $h_i \in H_i$ zur Vorhersage der Zielobjekt-Bewertung berufen.
Diese grundlegenden Hypothesen sollen mit konkreten Beobachtungen in Form der Instanzenmenge $\mathcal{T}_i$, mit gegebenen a-priori Wahrscheinlichkeiten $p(\mathcal{T}_i)$ für deren Auftreten, abgeglichen werden.
Wir suchen dann die *Maximum A Posteriori* (*MAP*) Hypothese h_i^{MAP}, d.h. die Hypothese, die am besten zu unseren gegebenen Beobachtungen $\mathcal{T}_i$ passt. Formal ausgedrückt bedeutet dies:

$$h_i^{MAP} = \underset{h_i \in H_i}{arg\,max}\; p(h_i|\mathcal{T}_i) = \underset{h_i \in H_i}{arg\,max} \frac{p(\mathcal{T}_i|h_i)p(h_i)}{p(\mathcal{T}_i)} \tag{5.12}$$

Wenn wir nun wissen, dass h_i^{real} genau die Hypothese ist, nach der der Lerner sucht, dann können wir das Problem der Instanzenselektion als die Suche nach einer optimalen Teilmenge

$\mathcal{S}_i \subseteq \mathcal{T}_i$ interpretieren, die die a-posteriori Wahrscheinlichkeit von h_i^{real} maximiert.

$$\begin{aligned} S_i^{opt} &= \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,max}\; p(h_i^{real}|\mathcal{S}_i) \\ &= \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,min}\; H(\mathcal{S}_i|h_i^{real}) - H(\mathcal{S}_i) + H(h_i^{real}) \\ &= \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,max}\; H(\mathcal{S}_i) - H(\mathcal{S}_i|h_i^{real}) \end{aligned} \tag{5.13}$$

Was die Berechnung der Vorhersage einer Bewertung für ein Zielobjekt i angeht, so benötigen wir für jede Instanz u aus $\mathcal{S}_i$ zum einen den Wert $v_{u,i}$ dieser Instanz bezogen auf das Zielobjekt i und zum zweiten zur Einstufung der Rationalität die Beschreibung $d_{u,i}$ der Instanz bezogen auf das Zielobjekt i. $d_{u,i}$ entspricht dabei nach Definition den Bewertungen des Nutzers für alle Objekte außer dem Objekt i und kann deswegen auch durch den Term $v_{u,\overline{\mathcal{T}}_{u,i}}$ ausgedrückt werden. Somit kann jede Instanz u aus $\mathcal{S}_i$ auch geschrieben werden als $(v_{u,i}, v_{u,\overline{\mathcal{T}}_{u,i}})$.
Wir können außerdem davon ausgehen, dass jede Instanz u unabhängig von den anderen Instanzen gezogen wird und können $\mathcal{S}_i^{opt}$ dann auch schreiben als:

$$\begin{aligned} S_i^{opt} &= \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,max} \sum_{u \in \mathcal{S}_i} \left[H(v_{u,i}, v_{u,\overline{\mathcal{T}}_{u,i}}) - H(v_{u,i}, v_{u,\overline{\mathcal{T}}_{u,i}}|h_i^{real}) \right] \\ &= \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,max} \sum_{u \in \mathcal{S}_i} \left[H(v_{u,i}|v_{u,\overline{\mathcal{T}}_{u,i}}) - H(v_{u,\overline{\mathcal{T}}_{u,i}}) - H(v_{u,i}|h_i^{real}, v_{u,\overline{\mathcal{T}}_{u,i}}) + H(v_{u,\overline{\mathcal{T}}_{u,i}}) \right] \end{aligned}$$

Weiterhin kann davon ausgegangen werden, dass der Wert $v_{u,i}$ einer Instanz unabhängig von seiner Beschreibung $v_{u,\overline{\mathcal{T}}_{u,i}}$ ist, wenn die zugrunde liegende Hypothese - die erst die Verbindung zwischen beiden schafft - fehlt, und wir erhalten dann:

$$= \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,max} \sum_{u \in \mathcal{S}_i} \left[H(v_{u,i}) - H(v_{u,i}|h_i^{real}, v_{u,\overline{\mathcal{T}}_{u,i}}) \right] \tag{5.14}$$

Da h_i^{real} wie angesprochen lediglich die Lücke zwischen Instanzenwert und -beschreibung schließt, kann sie in den Gleichungen weggelassen werden und wir erhalten nach Definition 5.11 der Rationalität abschließend:

$$S_i^{opt} = \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,max} \sum_{u \in \mathcal{S}_i} \left[H(v_{u,i}) - H(v_{u,i}|v_{u,\overline{\mathcal{T}}_{u,i}}) \right] = \underset{\mathcal{S}_i \subseteq \mathcal{T}_i}{arg\,max} \sum_{u \in \mathcal{S}_i} R_{u,i} \tag{5.15}$$

Die vorherigen formalen Betrachtungen zeigen somit, dass die Rationalität einer Instanz allgemein beim Lernen eine große Rolle spielt, denn nach Gleichung 5.15 trägt eine Instanz mit hoher Rationalität auch viel zu einer hohen a-posteriori Wahrscheinlichkeit der Hypothese h_i^{real} bei, während Instanzen mit niedriger Rationalität nur wenig zur a-posteriori Wahrscheinlichkeit beitragen und Instanzen mit negativer Rationalität sogar als Rauschen fungieren, das die Wahrscheinlichkeit absenkt.

c) *Berechnung der Rationalität*
Um die Rationalität wie in 5.11 berechnen zu können, muss eine Abschätzung der a-priori und der a-posteriori Unsicherheit des Instanzenwertes $v_{u,i}$ vorgenommen werden.
Für die a-priori Unsicherheit gibt es zwei häufig benutzte Methoden der Abschätzung, abhängig von der Anzahl der in der Datenbasis vorhandenen Instanzen:

i. *Wenige Instanzen:* In diesem Fall nimmt man eine Gleichverteilung der möglichen Instanzenwerte an, d.h. bei N möglichen Werten für $v_{u,i}$ wählt man $H(v_i = v_{u,i}) = -log_2 \frac{1}{N}$
ii. *Viele Instanzen:* Hier greift man auf einen statistischen Ansatz zurück. Ist z.B. $v_{u,i} = 1$ und 30% aller Nutzer u, die das Zielobjekt i bewertet haben, haben i mit dem Wert 1 bewertet, so ist $H(v_i = v_{u,i}) = -log_2 0,3$

Im Gegensatz zu dieser relativ einfachen Berechnung lässt sich die a-posteriori Wahrscheinlichkeit nicht so einfach abschätzen, da das Zielobjekt i jedes Objekt aus der Menge der insgesamt vorhandenen Objekte sein kann. Im konkreten Fall unserer Versuchsumgebung mit knapp 3900 Filmen und 5 möglichen Bewertungen würde ein naives Bayes'sches Abschätzungsverfahren $3900 \cdot 3900 \cdot 5 \cdot 5$ mögliche Fälle betrachten müssen, was von der Rechenzeit und dem benötigten Hauptspeicher her nicht akzeptabel wäre.

d) *Generelle Rationalität von Instanzen*
Aufgrund der genannten Performance-Probleme verwenden [YU et al. 2003] eine abgeschwächte Definition der Rationalität einer Instanz, ohne Spezifikation der genauen Werte der Bewertungen, vielmehr hängt die Definition nur davon ab, welche Objekte ein Nutzer bewertet hat.[10]

Definition 1. *Sei eine Instanz $u \in \mathcal{T}_i$ mit ihrer Eigenschaftsmenge $\overline{\mathcal{T}}_{u,i}$ und dem Zielobjekt i gegeben. Wenn die a-priori Unsicherheit der Bewertungen auf dem Objekt i durch die Entropie $H(V_i)$ gegeben ist, dann ist die* generelle Rationalität der Instanz u bezogen auf das Zielobjekt *i, bezeichnet mit $R^*_{u,i}$, die Reduktion der Unsicherheit von V_i bei gegebenem Wissen in Form der Bewertungen $V_{\overline{\mathcal{T}}_{u,i}}$ auf der Eigenschaftsmenge $\overline{\mathcal{T}}_{u,i}$:*

$$R^*_{u,i} = H(V_i) - H(V_i|V_{\overline{\mathcal{T}}_{u,i}}) = I(V_i; V_{\overline{\mathcal{T}}_{u,i}}) \tag{5.16}$$

$R^*_{u,i}$ ist eine Approximation von $R_{u,i}$ und hat den Vorteil, dass sie wie die Gewichtung von Eigenschaften in Abschnitt 1 das Maß der *mutual information* beinhaltet, womit Eigenschaftsgewichtung und Instanzenselektion in einem einheitlichen informationstheoretischen Rahmen behandelt werden können.
Geht man bzgl. dieser Definition von der Unabhängigkeit von Instanzeneigenschaften $j \in \overline{\mathcal{T}}_{u,i}$ bei gegebenem V_i aus, kann man, wie [YU et al. 2003] gezeigt haben, die generelle Rationalität einer Instanz wesentlicher einfacher berechnen als:

$$R^*_{u,i} = \sum_{j \in \overline{\mathcal{T}}_{u,i}} I(V_i; V_j) \tag{5.17}$$

Dies zeigt auch, dass zwischen der Relevanz einer Instanz und einer Eigenschaft ein enger Zusammenhang besteht.

e) *Rationalitätsstärke einer Instanz*
Trotz theoretisch begründeter Wirksamkeit und einfacher Berechenbarkeit weist die generelle Rationalität noch die Schwäche auf, dass die Existenz irrelevanter Attribute, die die Genauigkeit bei instanzenbasierten Distanzberechnungen wie hier reduzieren, nicht notwendigerweise zu einem Absinken der generellen Rationalität führen.
Daher sollte nach [YU et al. 2003] bei zwei Instanzen u mit demselben Rationalitätswert diejenige zur Vorhersage bevorzugt werden, die weniger Eigenschaften $j \in \overline{\mathcal{T}}_{u,i}$ enthält, denn jede Instanz kann nach [DOMINGOS 1996] als spezifische Regel betrachtet werden und nach *Occam's razor*[11] sollten kürzere Regeln bevorzugt werden. Außerdem trägt nach Gleichung 5.17

[10] Was sich die Tatsache zunutze macht, dass Nutzer in kollaborativen Umgebungen im allgemeinen unterschiedliche Mengen von Objekten bewerten.
[11] Vgl. [MITCHELL 1997].

jede Eigenschaft j etwas zur Rationalität bei und somit ist bei Instanzen mit einer hohen Anzahl von Eigenschaften auch die Wahrscheinlichkeit höher, dass diese Instanzen mehr irrelevante Eigenschaften enthalten.
[YU et al. 2003] schlagen deshalb eine einfache Heuristik vor, um Instanzen mit einfacher Beschreibung gegenüber solchen mit komplexer Beschreibung zu bevorzugen. Das daraus entstehende Maß der *Rationalitätsstärke einer Instanz* wird dann in diesem Algorithmus letztendlich zur Instanzenselektion benutzt:

$$R^{st}_{u,i} = \frac{1}{|\overline{\mathcal{T}}_{u,i}|} R^{*}_{u,i} \tag{5.18}$$

Bei $|\overline{\mathcal{T}}_{u,i}|$ als Anzahl von Eigenschaften in $\overline{\mathcal{T}}_{u,i}$ kann die Rationalitätsstärke einer Instanz als durchschnittliche Relevanz der Eigenschaften einer Instanz u interpretiert werden, so dass eine Instanz mit vielen relevanten Eigenschaften als relevant bzgl. der Vorhersage einer Bewertung eingestuft wird.

3. *Vorhersage von Bewertungen*
Die durch [YU et al. 2003] eingeführten Verbesserungen der Eigenschaftsgewichtung und Instanzenselektion fließen schließlich in die Berechnung einer Vorhersage $P_{a,i}$ (*Prediction*) für einen Zielnutzer a und ein Zielobjekt i ein, die gegeben ist als Durchschnitt der Bewertungen anderer Nutzer b für das Zielobjekt i, jeweils gewichtet durch die Ähnlichkeit $r(a,b)$ der Nutzer b zum Zielnutzer a:

$$P_{a,i} = \overline{v}_a + k \sum_{b \in N(a,\mathcal{S}_i)} r(a,b)(v_{b,i} - \overline{v}_b) \tag{5.19}$$

Dabei bezeichnen $\overline{v}_a$ und $\overline{v}_b$ die Durchschnittsbewertungen des Zielnutzers a bzw. des aktuellen Vergleichsnutzers b, $v_{b,i}$ die Bewertung des aktuellen Vergleichsnutzers b für das Zielobjekt i und k einen Normalisierungsfaktor, so dass die absoluten Werte der Gewichte $r(a,b)$ in der Summe 1 ergeben.
Die Vergleichsnutzer b werden der Nachbarschaft $N(a,\mathcal{S}_i)$ des Zielnutzers a entnommen, d.h. den Nutzern aus der Menge $\mathcal{S}_i \subseteq \mathcal{T}_i$, die mindestens ein Objekt $j \neq i$ bewertet haben, das auch Zielnutzer a bewertet hat.
$\mathcal{T}_i$ entspricht der Trainingsmenge für das Zielobjekt i und beinhaltet alle Nutzer, die für i eine Bewertung abgegeben haben. Durch die zuvor beschriebene Instanzenselektion werden aus dieser Menge wiederum all jene Nutzer ausgewählt, die sich für die Vorhersage von Bewertungen für das Zielobjekt i besonders gut eignen. Diese ausgewählten Nutzer werden in der Selektionsmenge $\mathcal{S}_i$ zusammengefasst.

Um die Ähnlichkeit/Distanz $r(a,b)$ zwischen dem Zielnutzer a und einem Vergleichsnutzer b zu bestimmen, wird auf den bekannten *Pearson-Koeffizienten* zurückgegriffen, der allerdings in zweierlei Hinsicht modifiziert wird.
Die Eigenschaften des Zielnutzers a und des Vergleichsnutzers b, gegeben durch die Bewertungen $v_{a,j}$ bzw. $v_{b,j}$ für alle gemeinsam bewerteten Objekte j, exklusive dem Zielobjekt i, werden gemäß den Betrachtungen in Abschnitt 1 durch $W_{i,j} = I(V_i;V_j)$, also die *mutual information* zwischen Zielobjekt i und Vergleichsobjekt j gewichtet. Außerdem wird der sogenannte *beschränkte* Pearson-Koeffizient benutzt, der sich dadurch auszeichnet, dass die Eigenschaftswerte $v_{a,j}$ und $v_{b,j}$ der Nutzer a und b nicht in Relation zu ihren Durchschnittsbewertungen $\overline{v}_a$ bzw. $\overline{v}_b$, sondern zum Mittelpunkt v_0 der Bewertungsskala gesetzt werden.[12]
Somit ergibt sich der im Algorithmus benutzte *eigenschaftsgewichtete beschränkte Pearson-Koeffizient* als

[12] In unserer speziellen Versuchsumgebung mit den möglichen Bewertungen $1-5$ entspricht v_0 also dem Wert 3.

$$r(a,b) = \frac{\sum_{j \in O(a,b)} W_{i,j}^2 (v_{a,j} - v_0)(v_{b,j} - v_0)}{\sqrt{\sum_{j \in O(a,b)} W_{i,j}^2 (v_{a,j} - v_0)^2 \cdot \sum_{j \in O(a,b)} W_{i,j}^2 (v_{b,j} - v_0)^2}} \tag{5.20}$$

$O(a,b)$ entspricht dabei der Überlappungsmenge (*Overlap*), d.h. der Menge aller Objekte $j \neq i$, für die sowohl Zielnutzer a, als auch Vergleichsnutzer b eine Bewertung abgegeben haben.
Setzt man in dieser Gleichung $W_{i,j}$ auf 1, so erhält man den normalen beschränkten Pearson-Koeffizienten.

5.1 Implementierung

Der zuvor theoretisch beschriebene kollaborative Empfehlungsalgorithmus von [YU et al. 2003] wurde in der objektorientierten Programmiersprache *Java* als Plugin der in Abschnitt 4.5 beschriebenen Lernumgebung *YALE* implementiert, kann aber auch stand alone betrieben werden.
Der Algorithmus wurde so implementiert, dass sich möglichst viele Eigenschaften, wie die zuvor beschriebene Eigenschaftsgewichtung und Instanzenselektion bei Bedarf an- oder ausschalten lassen, so dass der Algorithmus in unterschiedlichen Konfigurationen einsetzbar und somit ein Vergleich mit anderen kollaborativen Algorithmen möglich ist. So kann der Algorithmus letzten Endes auch so konfiguriert werden, dass er als einfaches *Nearest-Neighbour*-Verfahren betrieben werden kann.
Eine Erweiterung, z.B. die Implementierung eines anderen Distanzmaßes $r(a,b)$ in 5.19 ist auf einfache Weise möglich.
Besonders unter der Umgebung *YALE* bietet sich der Vorteil, mittels der graphischen Benutzeroberfläche die gewünschten Eigenschaften einfach per Mausklick zu (de-)aktivieren, wie schon in Abb. 4.4 gezeigt.

5.1.1 Parameter des Algorithmus

Im Folgenden werden die wichtigsten parametrisierbaren Eigenschaften des Algorithmus vorgestellt, sowie der jeweilige Wert genannt, mit dem der Algorithmus in den praktischen Versuchen betrieben wurde. Wegen der langen Lernzeit des Modells (Eigenschaftsgewichte und Ordnung der Nutzer nach Rationalität) für den großen *MovieLens*-Datensatz und der Tatsache, dass eine der Optimierungen (siehe Abschnitt 5.1.2) nur dann die Geschwindigkeit steigert, wenn alle Vorhersagen für denselben Nutzer generiert werden (was bei einer Kreuzvalidierung über allen Bewertungen des Datensatzes nicht der Fall ist), wurden die meisten der Parameterwerte gemäß den Angaben in der Untersuchung von [YU et al. 2003] gesetzt. Eine vollständige Parameteroptimierung mit integrierter Kreuzvalidierung wäre zeitlich nicht möglich gewesen. Teilweise wurden für einzelne Parameterwerte jedoch Performance-Vergleiche durchgeführt, d.h. der *MAE*-Wert durch eine Kreuzvalidierung mit 10 Durchgängen auf dem großen *MovieLens*-Datensatz gemessen.
Es folgen nun die wichtigsten als Parameter setzbaren Eigenschaften des Algorithmus:

- *constrained pearson coefficient* (boolean) - gibt an, ob in Gleichung 5.20 der Mittelpunkt v_0 der Bewertungsskala benutzt wird oder die mittleren Bewertungen $\overline{v}_a$ bzw. $\overline{v}_b$ der Nutzer a und b.
  Dem Artikel von [YU et al. 2003] folgend, wurde der beschränkte Pearson-Koeffizient auch in den praktischen Versuchen dieser Arbeit benutzt.

- *overall mean values* (boolean) - gilt nur für den Fall, dass *constrained pearson coefficient* auf *false* gesetzt ist. Dann kann durch diesen Parameter festgelegt werden, ob für $\overline{v}_a$ und $\overline{v}_b$ der Mittelwert über allen abgegebenen Bewertungen von a bzw. b berechnet werden soll oder nur über den Bewertungen, die a und b für gemeinsam bewertete Objekte ($O(a,b)$ in Gleichung 5.20) abgegeben haben.
  Wegen Benutzung des beschränkten Pearson-Koeffizienten kommt dieser Parameter in den Versuchen nicht zum Tragen.

- *feature weighting* (boolean) - schaltet den eigenschaftsgewichteten Pearson-Koeffizient ein oder setzt $W_{i,j}$ in Gleichung 5.20 auf 1.
  Die Nutzung der Eigenschaftsgewichtung ist eine Grundlage der Ausführungen von [YU et al. 2003]. In einem Performance-Vergleich zeigte der Algorithmus bei Benutzung der Eigenschaftsgewichtung den besseren *MAE*-Wert von 0.723 gegenüber 0.726 ohne Eigenschaftsgewichtung (s.u.), daher wurde die Eigenschaftsgewichtung benutzt.

- *instance selection* (boolean) - schaltet die Instanzenselektion nach [YU et al. 2003] ein bzw. aus. Bei Nichtnutzung entspricht die Menge $\mathcal{S}_i$ in Gleichung 5.19 der Trainingsmenge $\mathcal{T}_i$.
  Die finale Version des Algorithmus (siehe Abschnitt 5.1.2) machte bzgl. der benötigten Voraussagezeit ein Generieren aller Empfehlungen für einen Nutzer grundsätzlich auch ohne Instanzenselektion möglich. Die Benutzung der Selektion brachte zudem sowohl im Bezug auf den *MAE* (0.741 gegenüber 0.723 ohne Selektion, s.u.), als auch subjektiv gesehen (noch weniger Divergenz der einzelnen Empfehlungslisten, siehe Abschnitt 5.2) schlechtere Ergebnisse. Daher wurde beim endgültigen Generieren der Empfehlungslisten auf die Instanzenselektion verzichtet.

- *predictor set sampling rate* (real) - Bei eingestellter Instanzenselektion wird hier der Samplingfaktor F_s angegeben, d.h. aus der Trainingsmenge $\mathcal{T}_i$ werden nach absteigender Rationalitätsstärke die ersten $F_s \cdot |\mathcal{T}_i| = |\mathcal{S}_i|$ Nutzer für die Vorhersage ausgewählt.
  Wegen der zuvor erwähnten Gründe spielte dieser Parameter in den praktischen Versuchen keine Rolle.

- *predictor set min size* (integer) - Um bei kleinem Samplingfaktor F_s ein zu starkes Absinken der Vorhersagegenauigkeit zu vermeiden, kann hier die minimale Anzahl n_{min} von Nutzern für eine Vorhersage bestimmt werden. Ist durch den eingestellten Samplingfaktor $|\mathcal{S}_i| < n_{min}$ so werden bei nach absteigender Rationalitätsstärke geordneten Nutzern, die nächsten $n_{min} - |\mathcal{S}_i|$ Nutzer aus $\mathcal{T}_i$ zu $\mathcal{S}_i$ hinzugefügt. Ist $|\mathcal{T}_i| < n_{min}$ so werden selbstverständlich nur $|\mathcal{T}_i|$ Nutzer zur Vorhersage benutzt.
  Bei den Versuchen mit eingeschalteter Instanzenselektion wurde das Verhältnis von Sampling-Mindestanzahl zur Gesamtzahl aller Nutzer so gewählt wie bei [YU et al. 2003], d.h. ein Verhältnis von 0.015. Somit wurde die Mindestzahl bei den Versuchen auf 91 gesetzt (gegenüber der Gesamtzahl von 6040 Nutzern). Da die Instanzenselektion wegen schlechterer Ergebnisse jedoch letztendlich nicht benutzt wurde, ist der Wert dieses Parameters unerheblich.

- *min. overlap* (integer) - Durch diesen Parameter ist eine weitere Einschränkung der Nutzer in der Samplingmenge $\mathcal{S}_i$ möglich. Um der kollaborativen Idee zu entsprechen, müssen zwischen dem Zielnutzer und einem zur Empfehlung herangezogenen Vergleichsnutzer Gemeinsamkeiten bzgl. des Geschmacks vorhanden sein. Eine notwendige Voraussetzung dafür ist, dass die Schnittmenge der von beiden Nutzern bewerteten Objekte nicht leer ist. Mit diesem Parameter gibt man an, wie viele Objekte diese Schnittmenge mindestens enthalten muss, damit der Nutzer aus $\mathcal{S}_i$ zur Vorhersage herangezogen wird.
  Wie bei [YU et al. 2003] wurden in den praktischen Versuchen Nutzer in die Trainingsmenge aufgenommen, die mit dem Zielnutzer mindestens einen Film gemeinsam hatten.

- *avg rating - min. no. of predictors* (integer) - Wenn für einen bestimmten Film keine geeigneten Vergleichsnutzer gefunden werden können, um eine Vorhersage zu generieren (Beispiel: Der Zielnutzer hat mit keinem der Vergleichsnutzer, die das Zielobjekt bewertet haben eine Überlappung von gemeinsam bewerteten Filmen), wird als Vorhersagewert auf die mittlere Bewertung über allen Bewertungen für das Zielobjekt zurückgegriffen.
  Dies führt zu Problemen, wenn nur sehr wenige Nutzer überhaupt das Zielobjekt bewertet haben, weil das Objekt gerade erst zur Datenbasis hinzugefügt wurde oder weit außerhalb des „Mainstream"-Geschmacks liegt. Gibt es bspw. nur zwei Nutzer, die das Zielobjekt überhaupt bewertet haben und beide haben dem Objekt die höchste Bewertung zukommen lassen, so wird das Objekt in einer Empfehlungsliste immer auf den vordersten Plätzen auftauchen, womit der Gedanke der personalisierten Empfehlungen zerstört wird.

Dieser Parameter gibt an, wie viele Nutzer ein Objekt mindestens bewertet haben müssen, damit der Mittelwert der Bewertungen des Objekts durch diese Nutzer als Ersatzvorhersagewert benutzt wird. Wird die angegebene Mindestanzahl von Nutzern unterschritten, so wird stattdessen der Mittelwert über allen Bewertungen in der Datenbasis als Ersatzwert herangezogen.
Hier wurde für die praktischen Versuche ein Wert von 3 eingestellt.

- *coefficient threshold* (real) - Hiermit wird einen Schwellenwert angegeben, den der Pearson-Koeffizient $r(a, b)$ aus Gleichung 5.20 überschreiten muss, damit der Vergleichsnutzer b für die Vorhersage benutzt wird.
  Wie bei [YU et al. 2003] wurde auch hier ein Wert von 0 benutzt.

- *k* (integer) - Der Name des Parameters bezieht sich auf das gebräuchliche *k*-Nearest-Neighbour-Verfahren und gibt an, wieviele der gefundenen Nachbarn eines Zielnutzers maximal für die Vorhersage benutzt werden. Wegen der in diesem Algorithmus gegebenen Möglichkeit des Samplings ist eine Verwendung dieses Parameters nur sinnvoll, wenn ein reines *kNN*-Verfahren eingestellt werden soll. Dieser Parameter kann dann die Länge der Liste der Nutzer in der Trainingsmenge begrenzen, ohne dass die Nutzer in der Trainingsmenge nach absteigenden Werten für die Rationalitätsstärke sortiert sind, wie dies beim Sampling der Fall ist.
  Wegen der Erfahrungen in den Tests mit eingeschaltetem Sampling wurde dieser Parameter in den praktischen Versuchen auf die Gesamtzahl der in der jeweils benutzten Datenbasis vorhandenen Nutzer, d.h. auf 6040 für den großen *MovieLens*-Datensatz gesetzt.

- *rating scale min.* (real) - Um eine möglichst breite Einsatzfähigkeit des Algorithmus zu gewährleisten, kann die Bewertungsskala frei bestimmt werden. Dieser Parameter definiert dabei den minimalen Wert auf der Bewertungsskala. Bei nominalen Bewertungsskalen muss eine entsprechende Abbildung generiert werden.
  Entsprechend der in der Versuchsumgebung benutzten Werteskala wurde dieser Wert in praktischen Versuchen auf 1 gesetzt.

- *rating scale max.* (real) - Der entsprechende maximale Wert auf der Bewertungsskala.
  In der konkreten Versuchsumgebung dieser Arbeit liegt dieser Wert bei 5.

- *rating scale stepsize* (real) - Gibt die Schrittweite auf der Bewertungsskala an.
  In der Versuchsumgebung ist die Schrittweite der Bewertungen 1.

5.1.2 Implementierungsprobleme und Optimierungen

Der vorgestellte Algorithmus konnte nicht sofort in einer endgültigen Version implementiert werden, sondern durchlief mehrere Optimierungsschritte, die jeweils von in der Praxis beobachteten Problemen, insbesondere für kollaborative Algorithmen typischen Skalierungsproblemen motiviert waren. Im Folgenden werden die aufgetretenen Probleme und die zu ihrer Lösung angewendeten Optimierungen beschrieben.

Speicherbasiert

Die erste Version des vorgestellten Algorithmus wurde als reiner speicherbasierter kollaborativer Algorithmus implementiert, d.h. die Datenbasis des Algorithmus, gegeben als die in Abschnitt 4.1 vorgestellte große *MovieLens*-Datenbank, bestehend aus Datentupeln der Form (u, o, v) mit u als Nutzernummer, o als Objektnummer (Filmnummer) und v als Bewertung von Objekt o durch Nutzer u, wurde komplett im Speicher gehalten.
Um den dadurch stark in Anspruch genommenen Hauptspeicher nicht weiter zu belasten und den Algorithmus somit auch für Rechner mit wenig Hauptspeicher lauffähig zu halten, wurden alle für die zuvor in Abschnitt 5 vorgestellten Gleichungen benötigten Werte „on-the-flow“ berechnet. Lediglich zwei Optimierungen bezogen auf die Rechenzeit waren in dieser Version enthalten:

- *Hashing* - Hashtabellen, in *Java* als Klasse *HashMap* implementiert, wurden genutzt, um auf Datentupel (u, o, v) in annähernd konstanter Zeit zugreifen zu können. Drei HashMaps wurden für den Zugriff über die Nutzernummer u, die Objektnummer o und die Bewertung v implementiert.
- *optimierter Vergleich von Arrays* - Um die Menge $O(a, b)$ von gemeinsam bewerteten Objekten der Nutzer a und b zu bestimmen, müssen alle Bewertungen der Nutzer bzgl. der enthaltenen Objektnummer miteinander verglichen werden, was bei n möglichen Bewertungen (= n bewertbaren Objekten in der Datenbasis) quadratische Laufzeit $\mathcal{O}(n^2)$ erfordert. Außerdem muss dieser Prozess pro Vorhersage für alle Nutzer in der Trainingsmenge $\mathcal{T}_i$ bzw. in der Samplingmenge $\mathcal{S}_i$ durchgeführt werden, was bei m möglichen Nutzern rechnerischen Aufwand von $\mathcal{O}(m \cdot n^2)$ bedeutet und in unserer Versuchsumgebung mit $n = 3900$ und $m = 6040$ schnell zu sehr zeitaufwändigen Rechnungen führt.
  Aus diesem Grund wurden für diesen Schritt die Bewertungen zweier Nutzer a und b von den HashMaps in zwei Arrays kopiert und diese Arrays nach aufsteigenden Objektnummern o_a und o_b für die Nutzer a und b geordnet. Somit konnte bei der Suche nach einem gemeinsam bewerteten Objekt j im Fall $o_a > o_b$ die Suche abgebrochen werden. Bei Aufwand $\mathcal{O}(n)$ für das Umkopieren, $\mathcal{O}(n \log n)$ für das Sortieren der Arrays mit einem fortgeschrittenen Sortieralgorithmus und Zeit $\mathcal{O}(n \log n)$ für den Vergleich der zwei Arrays ergibt sich somit Laufzeit $\mathcal{O}(m \cdot n)$ im Gegensatz zu $\mathcal{O}(m \cdot n^2)$.

Diese erste Version des Algorithmus konnte zwar durch nicht übermäßigen Hauptspeicherverbrauch auch auf weniger gut ausgestatteten Rechnern betrieben werden, war jedoch trotzdem praktisch nicht einsetzbar, da selbst auf dem kleinen *MovieLens*-Datensatz für die Vorhersage einer einzelnen Bewertung im Schnitt 30 Sekunden benötigt wurden. Geht man davon aus, dass die meisten Probanden in der Versuchsumgebung nur die Mindestanzahl von 20 Filmen bewerten, so müssen zur Erstellung der Empfehlungsliste bei insgesamt 3900 Filmen pro Versuchsperson 3880 Vorhersagen gemacht werden. Da die Versuchsumgebung letztendlich auf dem großen *MovieLens*-Datensatz arbeitet und dieser von der Anzahl der Nutzer, Filme und Bewertungen ca. um den Faktor 10 größer ist, müssen die genannten Vorhersagezeiten zusätzlich (im linearen Fall) mit dem Faktor 10 multipliziert werden. Die erste Version des Algorithmus hätte somit für die Berechnung einer Empfehlungsliste ca. 323,3 Stunden, also fast 2 Wochen benötigt.
Diese, wie auch alle nachfolgenden Angaben beziehen sich dabei von der Hardware her auf einen *Sun*-Rechner vom Typ *Blade 2500* mit 2 *UltraSparc-IIIi*-Prozessoren mit je 1280 MHz und einem Hauptspeicher von 2048 MB.

Modellbasiert

Um den Performance-Problemen der ersten Version zu begegnen, wurde der Algorithmus komplett neu implementiert. Vornehmlich fand dabei eine Hinwendung von einem rein speicherbasierten Ansatz zu einem mehr modellbasierten Ansatz[13] statt, indem von konkreten Bewertungen der Versuchspersonen unabhängige Werte zuvor berechnet und als Modell abgespeichert wurden:

- Die Gewichte $W_{i,j}$ aus Gleichung 5.20 wurden für alle Paare (i, j)[14] von Objekten $i \neq j$ berechnet und als Datei abgespeichert, die vom Algorithmus für Vorhersagen eingeladen werden konnte.
- Da die Definition der Rationalitätsstärke $R^{st}_{u,i}$ eines Nutzers u für die Vorhersage einer Bewertung für ein Objekt i auf den Gewichten $W_{i,j}$ beruht (Gleichungen 5.18 und 5.17), wurden anschließend für alle Objekte i die Werte $R^{st}_{u,i}$ für alle Nutzer u berechnet, die das Objekt i bewertet hatten und jedem Objekt i eine Liste von Tupeln $(u, R^{st}_{u,i})$ zugewiesen, geordnet nach absteigenden Werten $R^{st}_{u,i}$. Auch diese Listen konnten abgespeichert und wieder eingeladen werden.
- Grundsätzlich wurde versucht, doppelte Berechnungen zu vermeiden. Zum einen durch Caching möglichst vieler des öfteren benötigter Werte, zum anderen durch Vermeidung redundanter Berechnungen, wie z.B. wiederholten Schleifendurchläufen mit denselben Werten wie in den Gleichungen 5.4, 5.5 und 5.6 bei separater Berechnung.

[13] Nach der Klassifikation in Abschnitt 2.3 also ein gemischter Typ.
[14] Dabei gilt $W_{i,j} = W_{j,i}$, d.h. für diesen Fall wurde auch nur ein Wert abgespeichert, so dass sich eine Dreiecksmatrix ergab.

Die Berechnung des Modells bestehend aus der Dreiecksmatrix der Gewichte $W_{i,j}$ und den sortierten Listen mit zur Vorhersage für ein Objekt geeigneten Nutzern wurde vor dem Einsatz berechnet, was ca. 5 Tage in Anspruch nahm. Das Modell konnte dann vom Algorithmus eingeladen und auf die Bewertungen von Nutzern angewendet werden. Dadurch ergab sich eine Beschleunigung um den Faktor 10, d.h. zur Berechnung einer Vorhersage wurden nun auf dem großen *MovieLens*-Datensatz im Schnitt 30 Sekunden benötigt, statt vorherigen 300.
Großer Nachteil der Optimierung war die wesentlich erhöhte Belastung des Hauptspeichers. Ein Speicherverbrauch von ca. 1 GB machte nun eine Ausführung auf vielen Rechnern unmöglich. Außerdem führt die Verwendung eines modellbasierten Ansatzes auch zu einer gewissen Ungenauigkeit, da bei der Berechnung der Gewichte $W_{i,j}$ die Bewertungen eines Zielnutzers, für den Vorhersagen gemacht werden sollen nicht einbezogen werden. Je mehr Bewertungen dieser Zielnutzer vorgenommen hat, desto größer ist die entstehende Ungenauigkeit. In einer realen Anwendung außerhalb dieser Versuchsumgebung müssten periodisch Neuberechnungen des Modells unter Einbeziehung neuer Bewertungen vorgenommen werden. Bei einer hohen Frequenz und Anzahl von abgegebenen Bewertungen hätte man so stets mit mehr oder weniger größeren Ungenauigkeiten der Vorhersagen zu kämpfen.
Da bei 30 Sekunden pro Vorhersage für die Erstellung der Empfehlungsliste für einen Nutzer unter den oben gemachten Annahmen immer noch ca. 32,3 Stunden benötigt wurden, war auch diese zweite Version des Algorithmus praktisch nicht wirklich einsetzbar. Aus diesem Grund wurde in einer dritten Version eine weitere Optimierung vorgenommen.

Laufzeitoptimierung durch Hashing

Als Mittel zur weiteren Optimierung wird wie zuvor schon das Caching angewandt, wobei man sich hier eine spezielle Gegebenheit der Versuchsumgebung zunutze macht.
Da hier Empfehlungslisten, d.h. eine große Anzahl von Vorhersagen für denselben Zielnutzer a in Folge berechnet werden, bleibt a in den Ähnlichkeitsgewichten $r(a, b)$ in der Gleichung 5.20 stets gleich. Sei $N^{all}(a)$ die Schnittmenge aller Mengen $N(a, \mathcal{S}_i)$ aus Gleichung 5.19 für alle möglichen Zielobjekte i, die der Zielnutzer nicht bewertet hat. Dann ändert sich diese Menge $N^{all}(a)$ nach einigen Vorhersagen nicht mehr, da bereits alle möglichen Nachbarn b betrachtet wurden.
Werden nun im Rahmen einer einzelnen Vorhersage die Werte $r(a, b)$ für alle betrachteten Nachbarn in einer weiteren HashMap mit Zugriff über die Nummer eines Nachbarn b abgelegt, so wird nach einer Anlaufzeit mit zeitlich kostenintensiven Vorhersagen bei der Berechnung der Vorhersagen nur noch auf gecachte Werte zurückgegriffen, was schließlich zu einer Vohersagezeit im Millisekunden-Bereich führt.
Mit dieser Optimierung konnte die gesamte Empfehlungsliste für einen einzelnen Nutzer mit 20 bewerteten Filmen im Schnitt in 15 Minuten erstellt werden[15]. Mit einem speziell für diesen Zweck geschriebenen Rahmenprogramm, welches das zuvor erstellte Modell einmalig, sowie die Bewertungen von einer Liste von Nutzern im Batchbetrieb einliest, können so die Empfehlungslisten in annehmbarer Zeit auch für eine größere Anzahl von Nutzern erstellt werden.
Während die zusätzliche Belastung des Hauptspeichers bei 6040 Nutzern insgesamt zu vernachlässigen ist, bleibt diese Optimierung doch auf die spezielle Versuchsumgebung beschränkt. Werden für jeden Nutzer nur wenige Vorhersagen berechnet, wie z.B. bei einer Kreuzvalidierung mit bzgl. der Nutzernummer ungeordneten Listen, so kommt der Vorteil des Cachings der Ähnlichkeitsgewichte nicht zum Tragen. Auch in realen Anwendungen, wo Nutzer beispielsweise eine Vorhersage für nur einige wenige spezielle Filme erfragen, ist diese Optimierung nicht anwendbar und kaum ein Nutzer wird bei einer interaktiven Anwendung mit einer Wartezeit von 30 Sekunden für eine Vorhersage zufrieden sein.
Somit zeigt die praktische Umsetzung ein typisches Problem kollaborativer Algorithmen, nämlich die Skalierbarkeit, die in realen Anwendungen des vorgestellten Algorithmus mit einer stetig wachsenden Anzahl von Nutzern, Filmen und Bewertungen noch extremer zu Tage treten dürfte.

[15] Hierbei gilt, dass die benötigte Zeit zur Erstellung der Empfehlungsliste für einen Nutzer von der Zahl der von ihm getätigten Bewertungen abhängt. Bei 900 abgegebenen Bewertungen beträgt die Zeit zur Erstellung der Liste ca. 1 Stunde.

5.2 Versuchsergebnisse

Im Folgenden werden die Ergebnisse der praktischen Versuche mit dem kollaborativen Algorithmus nach [YU et al. 2003] in der in Kapitel 4 beschriebenen Versuchsumgebung vorgestellt. Neben den Ergebnissen der Vorhersagegenauigkeit und der Anzahl generierter Empfehlungen, die von den Versuchspersonen als „interessant“ bewertet wurden, enthält der folgende Abschnitt auch andere Erfahrungen und Ergebnisse, die die Versuche erbracht haben. Einige dieser Erfahrungen werden in Kapitel 6 dann dazu verwendet werden, eine eigenschaftsorientierte Erweiterung des kollaborativen Verfahrens zu entwickeln, dass zu einer Verbesserung der Interessantheit der generierten Empfehlungen führen soll.

Vorhersagegenauigkeit

Die Beurteilung eines Empfehlungsalgorithmus bzgl. der Genauigkeit seiner Vorhersagen ist, wie schon in Kapitel 1.1 erwähnt notwendig, um die grundsätzliche Tauglichkeit eines Verfahrens zu bestimmen.
[HERLOCKER et al. 2004] bezeichnen dabei wie andere Forscher auch einen *MAE*-Wert von 0.73 auf einer Bewertungsskala mit fünf Werten, wie es in der Versuchsumgebung dieser Arbeit der Fall ist, als „magische Grenze“, da „state-of-the-art“-Systeme diese Grenze nur schwerlich unterschreiten. Gemessen an diesem Wert liefert der in den Versuchen benutzte und zuvor beschriebene kollaborative Algorithmus exzellente Ergebnisse, da er mit einem *MAE*-Wert von 0.723 (siehe Tabelle 5.1) diese Grenze unterschreitet. Der

MAE	
ohne Eigenschaftsgewichtung	0.726
mit Eigenschaftsgewichtung	0.723
Eigenschaftsgewichtung und Instanzenselektion, Samplingfaktor $= 0.125$	0.741
k-Nearest-Neighbour, $k = 200$	0.717

Tabelle 5.1: Vorhersagegenauigkeit kollaborativer Algorithmus: *MAE*

MAE-Wert ergibt sich dabei aus einer Kreuzvalidierung des Algorithmus mit 10 Durchgängen, mit den in Abschnitt 5.1.1 angegebenen Parametern auf dem großen *MovieLens*-Datensatz. Die Korrektheit der Berechnungen wurde dabei sowohl durch eine *Java-Test-Suite* als auch durch Nachrechnen eines komplexeren Beispiels per Hand überprüft.
Der positive Einfluss der Eigenschaftsgewichtung nach [YU et al. 2003] ist in diesem Zusammenhang weniger auffällig als bei den Untersuchungen, die [YU et al. 2003] durchführten. Bei Einsatz der Eigenschaftsgewichtung ergibt sich eine Verbesserung um 0.003 ([YU et al. 2003] stellten in ihren Untersuchungen eine Verbesserung von 0.044 fest). Auch die Instanzenselektion mit dem nach [YU et al. 2003] optimalen Samplingfaktor führt im Rahmen der großen *MovieLens*-Datenbank eher zu Verschlechterungen, hier ist eine Erhöhung des *MAE*-Wertes um 0.18 gegenüber dem Einsatz ohne Instanzenselektion zu beobachten.
Insgesamt bringt eine Parametrisierung des implementierten Algorithmus im Sinne eines einfachen k-Nearest-Neighbour-Verfahrens mit $k = 200$ bessere Ergebnisse (Verbesserung um 0.006) als das komplizierte Verfahren nach [YU et al. 2003].
Nach der Diskussion des Einflusses von verwendeten Daten auf das Ergebnis von Qualitätsbewertungen in Kapitel 3 lassen sich aus den Ergebnissen zwei mögliche Schlüsse ziehen:

1. *Hohe Bewertungsdichte für Mainstream-Filme verbessert Gesamt-Performance.*
   Betrachtet man das Verhältnis von Nutzern zu Objekten zu Bewertungen in dem hier verwendeten großen *MovieLens*-Datensatz (6040 Nutzer, 3900 Filme, 1000209 Bewertungen), so liegen mit durchschnittlich 256 Bewertungen pro Film viele Daten vor, um akkurate Vorhersagen für einen Film zu generieren. Natürlich wird es bzgl. der tatsächlichen Verteilung für einige Filme sehr viele („Mainstream“) und für andere Filme sehr wenig Bewertungen („Nischenfilme“) geben. Man kann davon ausgehen, dass das Verhältnis von Mainstream-Filmen gegenüber Nischenfilmen im Datensatz (wie

in der Realität auch) sehr hoch ist.[16] Für diese vielen Mainstream-Filme mit vielen Bewertungen werden somit wahrscheinlich sehr gute Vorhersagen generiert, so dass die wenigen schlechteren Vorhersageergebnisse für Nischenfilme bei der *MAE*-Berechnung nicht zu sehr ins Gewicht fallen.
Außerdem ist nicht ausgeschlossen, dass neben der Datenbereinigung, bei der nur Nutzer in den *MovieLens*-Datensatz aufgenommen wurden, die mindestens 20 Filme bewertet hatten[17], auch andere Bereinigungen vorgenommen wurden. Jedenfalls mögen sich dadurch die extrem guten *MAE*-Werte für die Versuchsumgebung erklären lassen, die die genannte „magische Grenze“ von 0.73 unterschreiten.

2. *Spezielle Eigenschaften der Daten beeinflussen das Ergebnis*.
Das schlechte Ergebnis des ausgefeilten Algorithmus nach [YU et al. 2003] gegenüber einem einfachen *Nearest-Neighbour*-Algorithmus mag ebenfalls durch den speziellen Datensatz erklärbar sein, der hier verwendet wurde. Für die Tests und Evaluierungen seines kollaborativen Algorithmus benutzte Yu den *EachMovie*-Datensatz, den er für die Tests zudem gewissen Datenbereinigungen unterzog. Aufgrund der Tatsache, dass nach [HERLOCKER et al. 2004] schon kleine Unterschiede in den Eigenschaften von Datensätzen zu großen Unterschieden in dem Ergebnis von Qualitätsbewertungen führen können, mag sich auch das vergleichsweise schlechte Abschneiden des Algorithmus nach [YU et al. 2003] gegenüber einem einfachen Algorithmus und die geringe Verbesserung der Fehlerrate bei Einsatz der Eigenschaftsgewichtung bzw. die große Verschlechterung bei Einsatz der Instanzenselektion erklären lassen.

Trotz dieser Überlegungen soll das gute Ergebnis des kollaborativen Algorithmus nicht geschmälert werden. Mag auch der Vergleich mit auf anderen Daten aufbauenden Verfahren aus den genannten Gründen schwierig sein, so ist ein *MAE* von 0.723 im speziellen Kontext dieser Versuchsumgebung sehr gut. Da die Unterteilung der „internen“ Bewertungsskala von Nutzern („Granularität der Nutzerpräferenz“, siehe Kapitel 3) meist sowieso grober ist, als die fünfstufige Unterteilung in der Versuchsumgebung,[18] macht ein durchschnittlicher Fehler von 0.723 oft nicht viel aus. Daher kann aufbauend auf diesem Ergebnis in Augenschein genommen werden, ob die Ergebnisse der Interessantheitsbewertungen ebenfalls so positiv ausfallen.

Interessantheit der Empfehlungen

An den mittels des *MovieVoter*-Programms aus Kapitel 4 durchgeführten Versuchen beteiligten sich anfangs 45 Personen (d.h. sie gaben Bewertungen für mindestens 20 Filme aus der großen *MovieLens*-Datenbank ab).
Die Interessantheitsbewertungen für die auf diesen Filmbewertungen basierenden Empfehlungen, die vom kollaborativen Algorithmus erzeugt wurden, führten 37 dieser 45 Personen durch. Interessantheitsbewertungen für das Hybridsystem aus Kapitel 6 wurden dann noch von 34 Benutzern abgeliefert. Um die Ergebnisse miteinander vergleichen zu können, werden hier entsprechend nur die Interessantheitsbewertungen dieser 34 Versuchspersonen präsentiert.
Da jeder Nutzer die ersten 20 Empfehlungen seiner Empfehlungsliste bewerten musste, ergeben sich somit insgesamt 680 Bewertungen. Der Durchschnittswert von $2.91 \approx 3$ für alle 680 Bewertungen, weist nach der gewählten Skala[19] auf Empfehlungen hin, die zwar korrekt, aber uninteressant sind. Interessant ist dabei ein Blick auf die Verteilung der einzelnen Bewertungen, die in Abb. 5.4 dargestellt ist.
Die Anzahl der Empfehlungen, die von den Nutzern hinsichtlich ihrer konkreten vorhergesagten Bewertungen als ungültig klassifiziert wurden, überwiegt, gefolgt von den interessanten Empfehlungen. Vergleichsweise werden entgegen der Aussage des Durchschnittswerts nur wenige Empfehlungen als korrekt, aber uninteressant bewertet. Somit widerspricht das Ergebnis des praktischen Versuchs den Tests mit dem hypothetischen *MAE*-Qualitätsmaß - viele der generierten Empfehlungen sind bzgl. der konkreten vorhergesagten Bewertung nicht korrekt. Wäre das *MAE*-Ergebnis auf die praktischen Versuche übertragbar, hätte

[16] Darauf lassen auch die Protokolldateien schließen, die bei der Generierung von Empfehlungslisten für die Nutzer angelegt wurden.
[17] Angabe in der *README*-Datei, die dem *MovieLens*-Datensatz beiliegt.
[18] Vgl. auch nachfolgende Versuchsergebnisse.
[19] Vgl. Kapitel 4.

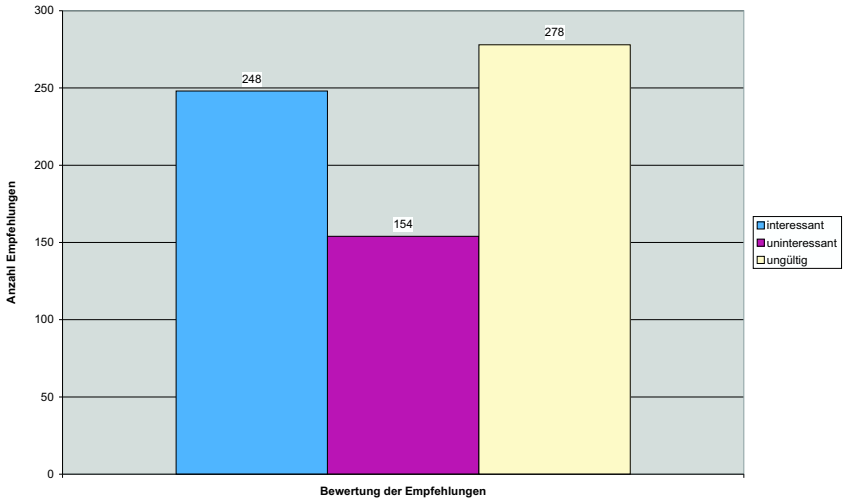


Abbildung 5.4: Interessantheitsbewertungen - kollaboratives System

es besonders aufgrund der Tatsache, dass die Nutzer bzgl. ihrer Präferenz eine geringere Granularität aufweisen als das System[20] eine wesentlich geringere Anzahl von falschen Vorhersagen geben müssen. Dies bestätigt die These aus Kapitel 1.1 sozusagen in verschärfter Form. War die These, dass hypothetische Qualitätsmaße nicht den wirklichen praktischen Nutzen der Empfehlungen in Form der Interessantheit für die Anwender wiedergeben, so ist in den meisten Fällen schon die Grundvoraussetzung für das Interessantheitsmaß, die Vorhersagegenauigkeit, nicht gegeben. Ein Anteil von 36% interessanter, d.h. für die Anwender

Interessantheitsbewertungen - kollaborativ		
Typ	**absolut**	**prozentual**
interessant	248	36
uninteressant	154	23
ungültig	278	41

Tabelle 5.2: Interessantheitsbewertungen kollaborativer Algorithmus - Zusammenfassung

nützlicher Empfehlungen (siehe Tabelle 5.2) entspricht nicht den Erwartungen, die das gute *MAE*-Ergebnis geweckt hat. Es zeigt sich also, dass wie erwartet von rein hypothetischen Bewertungen eines Empfehlungssystems nicht auf den tatsächlichen Nutzen unter realen Bedingungen geschlossen werden kann und somit praktische Versuche, sowie alternative Maße, wie die Interessantheit von Empfehlungen unabdingbar sind.

Vorhersagegenauigkeit vs. Interessantheit

Um den Zusammenhang zwischen Interessantheit und Vorhersagegenauigkeit weiter zu untersuchen, wurden anhand der im praktischen Versuch für den kollaborativen Algorithmus gewonnenen Interessantheitsbewertungen weitere Untersuchungen durchgeführt. Dafür wurde die bereits erwähnte Möglichkeit genutzt, den nach [Yu et al. 2003] implementierten kollaborativen Algorithmus durch spezielle Parametereinstellungen als einfaches k-Nearest-Neighbour-Verfahren laufen zu lassen. Zur Bestimmung der Vorhersagegenauigkeit wurde wieder auf den *MAE* zurückgegriffen. Zu diesem Zweck wurde für verschiedene k jeweils eine Kreuzvalidierung mit 10 Durchgängen auf den Daten der großen *MovieLens*-Datenbank durchgeführt.

[20] Siehe nachfolgende Ergebnisse.

Um die Fähigkeit des jeweiligen k-Nearest-Neighbour-Verfahrens zu bestimmen, interessante Empfehlungen zu produzieren, wurden jeweils Empfehlungslisten für alle 33 Versuchspersonen generiert.[21] Die jeweils ersten 20 Empfehlungen dieser Listen wurden dann mit denen verglichen, die der kollaborative Algorithmus mit Eigenschaftsgewichtung nach [YU et al. 2003] produziert hatte und die von den Nutzern im praktischen Versuch bzgl. der Interessantheit der Empfehlungen bewertet worden waren. Aussagen konnten dabei natürlich nur für solche Empfehlungen des k-Nearest-Neighbour-Algorithmus getroffen werden, die von den Nutzern im praktischen Versuch überhaupt bewertet worden waren.[22] Diese Empfehlungen werden im Folgenden als „bewertete Empfehlungen“ bezeichnet. Somit wurde bei spezifischem k für jeden Nutzer i die „interestingness accuracy“ des Nearest-Neighbour-Verfahrens als Quotient aus den „bewerteten Empfehlungen“, die vom Nutzer im praktischen Versuch als „interessant“ bewertet wurden und den „bewerteten Empfehlungen“ insgesamt berechnet oder formal ausgedrückt:

$$\mathrm{IA}_{k_i} = \frac{|\text{interessant}_{k_i}|}{|\text{bewertet}_{k_i}|}, \text{ für } i = 1, \cdots, m \text{ mit } m = 33 \text{ und } |\text{interessant}_{k_i}| \leq |\text{bewertet}_{k_i}|$$

Daraus konnte dann die allgemeine „interestingness accuracy“ für den k-Nearest-Neighbour-Algorithmus

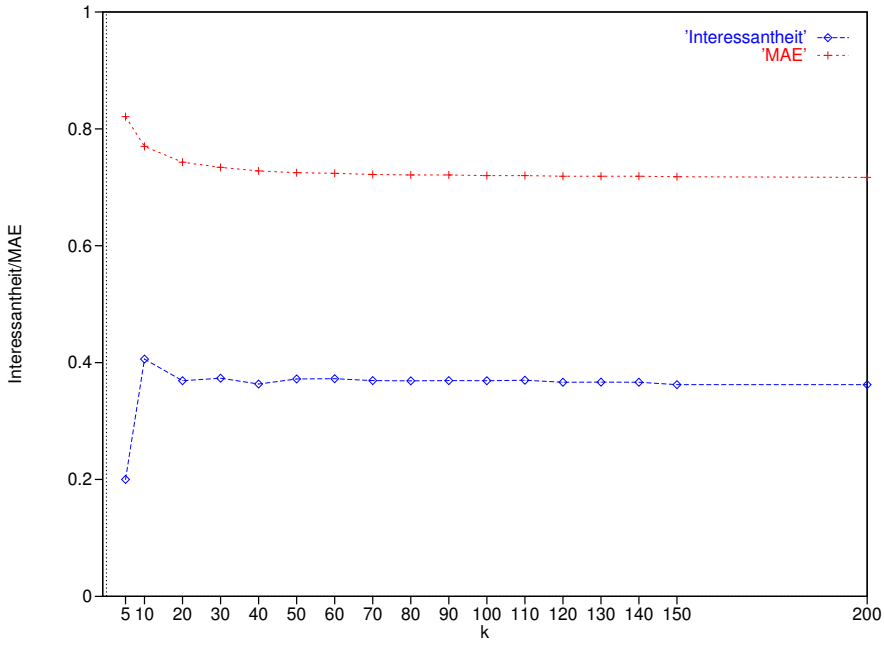


Abbildung 5.5: *Interessantheit* vs. *MAE* für kollaborativen k-Nearest-Neighbor-Algorithmus

und ein bestimmtes k als bzgl. der Nutzeranzahl gemittelte Summe der „interestingness accuracy“-Werte der einzelnen Nutzer berechnet werden. Damit das Ergebnis interpretierbar wird, wurden nur die Werte jener Nutzer in die Berechnung mit einbezogen, bei denen die Anzahl der „bewerteten Empfehlungen“ mindestens bei 5 lag:

$$\mathrm{IA}_k = \frac{\sum_{i=1}^{m} \begin{cases} \mathrm{IA}_{k_i} & \text{wenn } |\text{bewertet}_{k_i}| \geq 5 \\ 0 & \text{sonst} \end{cases}}{|\mathrm{IA}_{k_i}| \text{ mit } |\text{bewertet}_{k_i}| \geq 5}$$

Diese „interestingness accuracy“-Werte für die Ausprägungen des k-Nearest-Neighbour-Algorithmus sind natürlich nicht so aussagekräftig wie Werte, die sich aus Bewertungen aller Top-20-Empfehlungen der generierten Listen durch die Versuchspersonen ergeben würden. Solch eine vollständige Untersuchung schied

[21] Die Interessantheitsbewertungen für den kollaborativen Algorithmus wurden zwar von 37 Personen durchgeführt, jedoch lagen zum Zeitpunkt des Tests leider erst die Interessantheitsbewertungen von 33 Nutzern vor.

[22] Also den Top-20-Empfehlungen der Listen im praktischen Versuch.

jedoch aufgrund des mehrfach erwähnten erheblichen Zeitaufwands solcher praktischen Versuche aus, zumal es wohl keiner Versuchperson zuzumuten gewesen wäre, 17 verschiedene Empfehlungslisten zusätzlich zu bewerten.
Die Gegenüberstellung der berechneten „interestingness accuracy"-Werte und der *MAE*-Werte, wie sie Abb. 5.5 zeigt, ist jedoch trotz der Durchführung unter „Laborbedingungen" sehr interessant, da sie nochmals die These der Arbeit bestätigt, dass sich die reine Vorhersagegenauigkeit eines Empfehlungsverfahrens nicht konform zur Interessantheit der generierten Empfehlungen verhält. Während der MAE mit steigendem k stetig abnimmt[23], die allgemeine Vorhersagegenauigkeit also steigt, wird die maximale Interessantheit der Empfehlungen bereits bei $k = 10$[24] erreicht. Danach zeigen sich nur noch nicht signifikante Schwankungen und für $k \geq 150$[25] stagniert die Entwicklung der Interessantheit schließlich. Hypothetische Genauigkeit eines Verfahrens und praktischer Nutzen können also nicht miteinander gleichgesetzt werden.

Geringe Granularität der Nutzerpräferenzen

Im Abschnitt 1.1 wurde die subjektive Erfahrbarkeit von winzigen Verbesserungen von Empfehlungsverfahren bezogen auf die hypothetische Genauigkeit von Vorhersagen in Frage gestellt. Dass diese kritische Betrachtung ihre Berechtigung hat, wurde während der praktischen Versuche dieser Arbeit bestätigt, indem eine bereits von [HERLOCKER et al. 2004] in der Praxis gemachte Erfahrung nachvollzogen werden konnte.
Lässt man denselben Nutzer zu unterschiedlichen Zeitpunkten dieselbe Menge von Objekten bewerten, so ergeben sich oft unterschiedliche Wertungen für dasselbe Objekt. Dieses Verhalten fiel im Rahmen der hier durchgeführten Versuche eher zufällig auf. So war die erste Version des *MovieVoter*-Programms noch nicht in der Lage, einmal vorgenommene Filmbewertungen bei einem Neustart wieder einzuladen, um den vorhandenen zu einem späteren Zeitpunkt weitere Bewertungen hinzuzufügen. Von mehreren Nutzern auf dieses Manko aufmerksam gemacht, wurde diese Funktionalität einer späteren Version des Programms hinzugefügt. Mehrere Nutzer hatten sich derweil provisorisch beholfen, indem sie der Datei mit den von ihnen bereits getätigten Bewertungen einen neuen Namen zuwiesen und so bei einem Neustart des *MovieVoters* neue Bewertungen in einer zweiten Datei abspeichern konnten. Teilweise entstanden so bei einigen Nutzern bis zu vier verschiedene Dateien mit Filmbewertungen. Bei der Anwendung einer kleinen Routine zum Verschmelzen verschiedener Bewertungsdateien zu einer einzigen Datei kam es oft zum Auftreten mehrfacher Bewertungen desselben Films, da die Nutzer vergessen hatten, dass ein Film von ihnen schon früher gewertet wurde. Auch als ein Nutzer die Bitte nach Durchführung einer Interessantheitsbewertung von Filmempfehlungen missverstand und stattdessen nochmals eine umfangreiche Bewertung von Filmen durchführte, trat diese Situation mehrfacher Bewertungen auf. In einem Großteil der Fälle waren die Werte für mehrfach bewertete Filme nicht identisch. Interessanterweise war der Unterschied zwischen je zwei Bewertungen für dasselbe Objekt jedoch nie größer als 1. Diese maximale Abweichung eine Entdeckung, die so in der Literatur bisher nicht zu finden ist. Sie lässt darauf schließen, dass die Granularität der Präferenzen von Nutzern eher gering und ihre „interne" Bewertungsskala somit eher der Kategorie *fuzzy* zuzuordnen ist, d.h. dass sie Filme subjektiv als „in etwa" gut/mittelmäßig/schlecht bewerten. Wenn Nutzer also dazu neigen, ein solch unscharfes Bewertungsmaß anzulegen, dann scheinen sie kaum in der Lage zu sein, Unterschiede in vorhergesagten Bewertungen in der Größenordnung von bspw. 0.01 subjektiv wahrzunehmen. Dann haben auch unter teilweise großem Aufwand durchgeführte minimale Verbesserungen der Vorhersagegenauigkeit wenig Sinn.
Außerdem unterstreicht dieses Ergebnis der geringen Granularität von Nutzerpräferenzen, wie zuvor erwähnt, den erheblichen Unterschied zwischen theoretisch ermitteltem *MAE* und tatsächlicher Vorhersagegenauigkeit aus Sicht der Nutzer.

[23] Je größer das k, desto langsamer nimmt der MAE-Wert ab, die Tendenz zur Abnahme bleibt jedoch bestehen. Dabei sinkt der *MAE* insgesamt von 0.821 auf 0.717.
[24] Dies gilt für die 25 der 33 Nutzer, bei denen in den generierten Empfehlungslisten mindestens 5 zuvor bewertete Empfehlungen auftauchen. Da dies immerhin 76% der Gesamtnutzer sind, ist der berechnete Wert aussagekräftig genug.
[25] Versuche für $k \in \{250, 300, 400, 500, 1000, 2000, 3000, 4000, 5000, 6040\}$ ergaben denselben IA_k-Wert wie für $k = 150$.

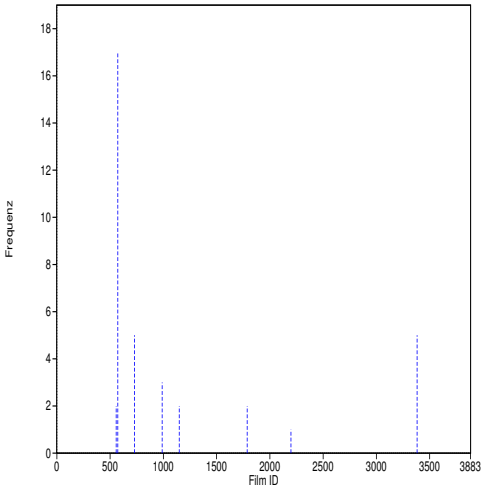


Abbildung 5.6: Verteilung Film IDs Rang 1, kollaborativer Algorithmus

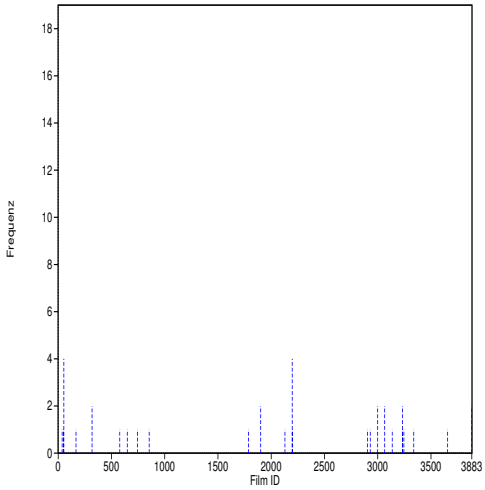


Abbildung 5.7: Verteilung Film IDs Rang 20, kollaborativer Algorithmus

Ähnliche Empfehlungslisten

Ein Problem des kollaborativen Algorithmus in den Versuchen war das Produzieren ähnlicher Empfehlungslisten für die verschiedenen Nutzer. Diese Ähnlichkeit bezog sich vor allem auf die ersten 20 Ränge, die ja gerade von den Nutzern bzgl. ihrer Interessantheit bewertet werden sollten. So traten meistens dieselben Filme in den Top-20 auf.
Bei diesen Filmen handelt es sich um „Nischenfilme“, die nur von wenigen Nutzern in der *MovieLens*-Datenbasis bewertet worden waren, wobei diese Nutzern den sogenannten „Power-Usern“ zugerechnet werden können, d.h. den Nutzern, die extrem viele Filme bewertet haben. Somit haben die meisten der Testpersonen in den Versuchen dieser Arbeit zumindest einige Filme mit diesen Nutzern gemeinsam, so dass die Bewertungen dieser Power-User bei der Vorhersage für die Nischenfilme zum Tragen kommen. Haben die Power-User diese Filme zudem extrem gut bewertet, wie es hier der Fall ist, so ergeben sich für diese Filme immer vorhergesagte Bewertungen, die zu einem Platz ganz oben in den Empfehlungslisten führen. Obwohl gerade die Fähigkeit genreübergreifende Empfehlungen zu generieren den großen Vorteil kollaborativer Verfahren darstellt, empfanden viele Versuchspersonen diesen Vorteil eher als Nachteil, da sie der Meinung waren, ihre persönlichen Präferenzen würden zu wenig in den obersten Rängen widergespiegelt.[26] Diese ungleiche Verteilung zeigt sich in den Abbildungen 5.6 und 5.7. Während die Verteilung der Film-IDs auf Rang 20 über allen 37 Nutzern[27] einer optimalen (Gleich-)Verteilung[28] halbwegs nahe kommt, sieht dies bei Rang 1 ganz anders aus. Hier ist es vor allem ein Film, der unverhältnismäßig oft auf Rang 1 landet. Dies wäre verständlich, wenn die meisten Testnutzer einen ähnlichen Geschmack hätten, jedoch zeigen sich bei Durchsicht der von den Nutzern bewerteten Filme sehr unterschiedliche Tendenzen bzgl. des Geschmacks. Dabei besteht bzgl. der ungleichen Verteilung von Film-IDs die Tendenz, dass das Ungleichgewicht umso größer ist, je niedriger der Rang ist. Dies zeigt auch Abb. 5.9, wo die Standardabweichungen der Film-IDs über den ersten 20 Rängen und allen Nutzern dargestellt sind. Von einigen „Ausreißern“ abgesehen, zeigt sich der grundsätzliche Trend, dass die Abweichung umso geringer ausfällt, je niedriger der betrachtete Rang ist. Auch die Verteilung der Film-IDs über den ersten 20 Rängen und allen

[26] Dies konnte entsprechenden Kommentaren der Nutzer entnommen werden.
[27] Empfehlungslisten für beide in dieser Arbeit verwendeten Verfahren wurden nur für diejenigen Nutzer erstellt, die eine Interessantheitsbewertung für die Empfehlungen des kollaborativen Algorithmus durchgeführt hatten.
[28] Optimal wäre bei Testnutzern mit sehr unterschiedlichen Geschmäckern eine daraus folgende sehr geringe Ähnlichkeit der Empfehlungslisten, bei denen jeder Nutzer andere Filme in den Top-20 empfohlen bekommt, so dass m unterschiedliche Film-IDs mit einer Frequenz von jeweils 1 eine Gleichverteilung bilden, wobei m die Gesamtzahl der Testnutzer bezeichnet.

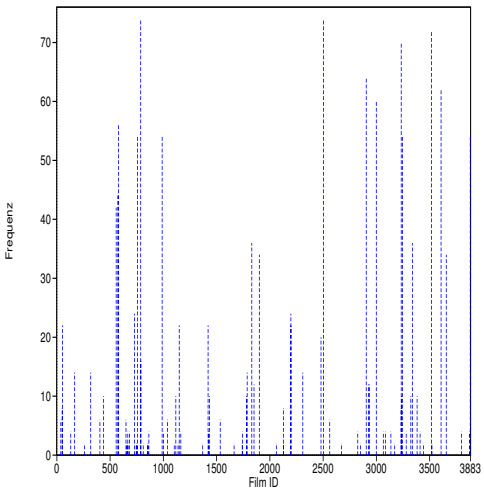


Abbildung 5.8: Verteilung Film IDs Rang 1-20, kollaborativer Algorithmus

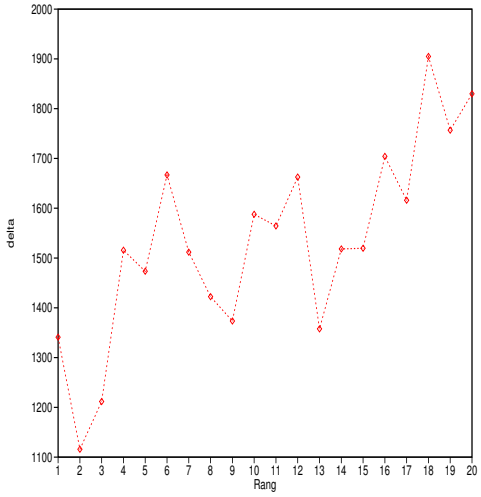


Abbildung 5.9: Standardabweichung Film IDs, kollaborativer Algorithmus

Nutzern, wie in Abb. 5.8 gezeigt, weist auf die Ähnlichkeit der Empfehlungslisten in den ersten 20 Rängen hin.

Spärliche Daten

Ein weiterer Nachteil der Nischenfilme und der damit verbundenen wenig vorhandenen Bewertungen ist, dass für einige dieser Filme keine Vorhersagen nach dem kollaborativen Prinzip getroffen werden können. Wenn nur wenige Nutzer diese Nischenfilme bewertet haben und es sich bei diesen Nutzern nicht um Power-User handelt bzw. der Geschmack dieser Nutzer von dem des Zielnutzers abweicht, kann es vorkommen, dass keine Übereinstimmungen zwischen Ziel- und Vergleichsnutzer aus der Datenbasis gefunden werden können. Somit muss zur Vorhersage auf alternative Werte zurückgegriffen werden. Der kollaborative Algorithmus wurde so implementiert, dass er in diesem Fall den Mittelwert aus den Bewertungen aller Nutzer, die den entsprechenden Film bewertet haben, als Vorhersagewert benutzt. So sinnvoll diese Methode der Generierung eines Ersatzvorhersagewertes bei theoretischer Betrachtung erscheint, so zeigten doch die praktischen Versuche erst die negativen Folgen in einer realen Anwendung, so dass dieser typische und bekannte Nachteil kollaborativer Algorithmen greifbar wurde. Dieses schon zuvor in Abschnitt 5.1 beschriebene Problem führte dazu, dass einige Nischenfilme stets mit demselben Ersatzvorhersagewert auf den obersten Rängen der Empfehlungslisten landeten, weil die wenigen Nutzer, die diese Filme bewertet hatten, sehr gute Bewertungen abgegeben hatten. Dies wiederum erzeugte in Verbindung mit dem im vorigen Abschnitt beschriebenen Problem Empfehlungslisten, denen fast jeglicher Anspruch auf Personalisierung abgesprochen werden musste. Um dieses Problem in Grenzen zu halten, wurde nachträglich der zuvor beschriebene Parameter *avg rating - min. no. of predictors* eingeführt und damit die endgültigen Empfehlungslisten für die praktischen Versuche erzeugt.

All die genannten Ergebnisse und Erfahrungen der praktischen Versuche weisen zum einen auf erhebliche Schwächen von rein hypothetischen Qualitätsbewertungen von Empfehlungsverfahren hin und zeigen ausserdem, dass kollaborative Verfahren alleine keine zufriedenstellenden praktischen Ergebnisse liefern. In Abschnitt 1.1 wurde die These aufgestellt, dass Hybridsysteme die Interessantheit von Empfehlungen verbessern können. Ein solches Hybridsystem wurde im Rahmen der gegebenen Arbeit entwickelt und soll im Folgenden vorgestellt werden.

6 Eigenschaftsbasierte Erweiterung: Verbessern der Interessantheit

> *„This is the most effective memory enhancement drug on the market. It'll improve your short-term memory. It'll improve your long-term memory. And most of all, it'll improve your short-term memory."*
>
> Jackie Rogers Jr., *Saturday Night Life*, 1975

Nachdem im vorherigen Kapitel gezeigt wurde, dass ein kollaborativer *state-of-the-art*-Algorithmus nicht ausreichend ist, um hinreichend interessante Filmempfehlungen zu produzieren, soll hier nun eine weitere These aus Abschnitt 1.1 untersucht werden - dass Hybridsysteme die Interessantheit von Empfehlungen verbessern können.
Wie bereits dort angesprochen, haben kollaborative Empfehlungsverfahren den Vorteil, dass kein inhaltlicher Zusammenhang zwischen den von ihnen empfohlenen Objekten und den vom jeweiligen Zielnutzer als gut bewerteten Objekten bestehen muss. Somit können außergewöhnliche Empfehlungen generiert werden, die nach Definition der Interessantheit in Abschnitt 3.3 das Potenzial haben, interessant zu sein. Die meisten kollaborativen Verfahren basieren dabei von ihrer Funktionsweise her auf instanzenbasiertem Lernen (*IBL*), insbesondere den sog. k-*Nearest-Neighbor*-Verfahren, wo zu einer Instanz, d.h. einem Zielnutzer, die k nächsten Nachbarn (Vergleichsnutzer) gesucht werden, die bzgl. eines definierten Distanzmaßes den geringsten Abstand zu dem betrachteten Zielnutzer aufweisen. Diese k Nachbarn werden dann zur Vorhersage der Bewertung für ein Zielobjekt bezogen auf den Zielnutzer herangezogen. Auch das in Kapitel 5 untersuchte kollaborative Verfahren arbeitet nach diesem Prinzip. Leider haben solche *IBL*-Verfahren jedoch einen entscheidenden Nachteil, die Qualität der Vorhersage hängt sehr stark von der verwendeten Distanzfunktion ab, insbesondere davon, welche Eigenschaften der Instanzen (d.h. welche Objektbewertungen der Vergleichsnutzer) in die Berechnungen eingehen.[1] Die Gefahr, redundante oder als Rauschen fungierende Eigenschaften zu stark in die Berechnungen einfließen und damit die Vorhersagen unbrauchbar werden zu lassen, ist sehr hoch.[2] Die Versuchsergebnisse in Kapitel 5 zeigen dieses Problem. Der Großteil der Versuchsteilnehmer sagte zwar aus, dass ihm die meisten der empfohlenen Filme gänzlich unbekannt seien (Stärke der kollaborativen Verfahren) und die als gültig aber uninteressant klassifizierten Empfehlungen machen auch den geringsten Anteil der Bewertungen aus, jedoch überwiegen Empfehlungen, die von den Anwendern als ungültig klassifiziert wurden (außergewöhnlich und unbekannt, aber nicht nützlich - Rauschen). Um diesem Problem zu begegnen, stehen einem drei Möglichkeiten zur Verfügung. Man kann den kollaborativen Algorithmus so verbessern, dass die Zahl der ungültigen Empfehlungen reduziert wird. Dies ist der Grund, warum viele *IBL*-Algorithmen Methoden der Eigenschaftsgewichtung benutzen.[3] Auch der in Kapitel 5 verwendete kollaborative Algorithmus nach [Yu et al. 2003] greift auf diese Art der Verbesserung zurück. Manchmal fällt diese Verbesserung allerdings nicht so stark aus, wie erhofft (siehe Abschnitt 5.2). Die zweite Möglichkeit ungültige Empfehlungen zu reduzieren wäre, ein ganz anderes Empfehlungsverfahren, wie z.B. ein eigenschaftsorientiertes Verfahren zu benutzen. Da eigenschaftsorientierte Verfahren die inhaltlichen Eigenschaften von bewerteten und unbekannten Objekten vergleichen, sind bei entsprechender Wahl der Eigenschaften (solche, die in der Menge der vom Zielnutzer gut bewerteten Objekte häufig dieselben Werte aufweisen) keine ungültigen Empfehlungen zu befürchten. Dafür können solche Systeme aber auch keine außergewöhnlichen Empfehlungen erzeugen.[4] Die dritte Möglichkeit, dem

[1] Vgl. [Yu et al. 2003].
[2] Siehe [Wettschereck et al. 1997].
[3] Siehe [Wettschereck et al. 1997].
[4] Siehe z.B. [Burke 2002].

Problem zu begegnen, ist ein Hybridsystem, wie es hier benutzt wird.

Auswahlkriterien

Das benutzte Hybridsystem ist als Aufsatz für das kollaborative Empfehlungssystem aus Kapitel 5 konzipiert, der die vom kollaborativen Verfahren erstellten Empfehlungen bzgl. inhaltlicher Eigenschaften wie Genre oder mitwirkende Schauspieler filtert und potenziell ungültige Empfehlungen (weil Rauschen) aus den Empfehlungslisten entfernt. Bezogen auf die verschiedenen Typen von Hybridsystemen wurde hier die Entscheidung zugunsten eines sogenannten Kaskaden-Hybridsystems getroffen. Bei Kaskadensystemen erzeugt ein bestimmtes Empfehlungssystem zuerst eigenständig eine Liste von Empfehlungen. Ein weiteres Empfehlungssystem modifiziert dann die bestehende Liste von Empfehlungen nach bestimmten Kriterien, so dass sich eine endgültige Empfehlungsliste ergibt. Es findet also keine völlige Neuerstellung von Empfehlungen statt, sondern lediglich eine Verfeinerung. Dies ist insbesondere in der hier vorliegenden Situation sinnvoll, da das kollaborative Empfehlungssystem aus Kapitel 5 nicht grundsätzlich ungültige Empfehlungen generiert. Es soll lediglich die Anzahl der interessanten Empfehlungen gesteigert und die Zahl der ungültigen Empfehlungen durch Herausfiltern vermindert werden. Außerdem hat ein Kaskadensystem den Vorteil, dass es als Aufsatz des bestehenden Empfehlungssystems einfacher zu implementieren ist, als dies im Fall einer völligen Neukonzeption der Fall wäre.
Der Grund, das Kaskadensystem so zu implementieren, dass die erste Empfehlungsliste vom kollaborativen und nicht vom eigenschaftsbasierten Verfahren erzeugt wird ist einfach: Würde das eigenschaftsbasierte System die grundlegenden Empfehlungen generieren, so wären diese aufgrund der genannten Eigenschaften solcher Systeme keine außergewöhnlichen und somit potenziell interessanten Empfehlungen. Um nun zu verhindern, dass die zweite Stufe der Kaskade, das eigenschaftsbasierte System, nachträglich alle interessanten Empfehlungen aus der vom kollaborativen Verfahren erzeugten Liste herausfiltert, wurden die Erfahrungen und Daten, die im Rahmen des praktischen Versuchs mit dem kollaborativen Verfahren aus dem vorigen Kapitel gewonnen wurden, genutzt. Die Versuchsteilnehmer beklagten sich in diesen Versuchen, sie würden keinerlei inhaltliche Übereinstimmung der kollaborativen Empfehlungen zu ihren bewerteten Filmen erkennen, was darauf hinweist, dass Empfehlungen ohne jeglichen inhaltlichen Bezug zu den von den Nutzern als gut bewerteten Objekten entfernt werden müssen. Gleichzeitig betonten die Nutzerinnen immer wieder, wie gänzlich unbekannt ihnen die meisten Empfehlungen - auch die von ihnen als interessant klassifizierten - seien. Die Grenze für das Kriterium „inhaltliche Verbindung“ darf also nicht zu hoch gesetzt werden. Daher wurde die inhaltliche Filterstufe so konzipiert, dass zu einer Empfehlung der ähnlichste vom Zielnutzer bewertete Film gesucht wird. Die minimale Grenze (Parameter *minimal similarity threshold*, s.u.), die dieser Film dabei bzgl. der Ähnlichkeit überschreiten muss, wurde auf 0 gesetzt, d.h. es muss ein bewerteter und damit bekannter Film vorhanden sein, der zumindest eine minimale Ähnlichkeit zur aktuell betrachteten Empfehlung hat (gibt es keinen Film mit einer Ähnlichkeit größer 0, so wird die Empfehlung als uninteressant eingestuft, da es sich höchstwahrscheinlich um eine ungültige Empfehlung handelt). Gleichzeitig wird die minimale Grenze aber auch nicht höher als 0 gesetzt, d.h. auch kleine Ähnlichkeiten (wie z.B. ein gleiches Genre unter vielen) reicht aus, um eine inhaltliche Verbindung herzustellen. Die Parameteroptimierung (s.u.) bestätigte dieses Vorgehen, da alle Werte für *minimal similarity threshold* größer als 0 schlechtere Ergebnisse erzielten. Zusätzlich muss eine Empfehlung aber ein weiteres Kriterium erfüllen, um nicht als ungültig herausgefiltert zu werden. Die vom Nutzer abgegebene Bewertung für den ähnlichsten Film und die vorhergesagte Bewertung für die Empfehlung müssen annähernd gleich sein. „Annähernd“ bezieht sich auf die in Abschnitt 5.2 gemachte Erfahrung mit der geringen subjektiven Granularität der Nutzerpräferenzen, die dazu führt, dass Objekte innerlich nur „in etwa“ als gut, mittelmäßig oder schlecht bewertet werden, wobei die Toleranz einen Punkt auf der fünfstufigen Bewertungsskala ausmacht. Auch dies wurde in der Parameteroptimierung bestätigt, da alle Toleranzwerte (Parameter *prediction difference*, s.u.) größer 1 ebenfalls schlechtere Ergebnisse brachten. Dass auch die Bewertungen von Empfehlung und Film ähnlich sein müssen, begründet sich in mehreren Annahmen, die zur Vereinfachung getroffen wurden. Sind sich die Bewertungen von Film und Empfehlung nicht ähnlich (Differenz größer 1), so gibt es zwei mögliche Fälle: Entweder die Empfehlung wurde wesentlich besser bewertet als der Film, dann haben wir es wahrscheinlich mit dem Problem von Nischenfilmen zu tun, die allen Nutzern

empfohlen werden (siehe Abschnitt 5.2) und somit höchstwahrscheinlich wieder mit ungültigen Empfehlungen. Diese Annahme birgt natürlich ein gewisses Risiko, dass eine tatsächlich interessante Empfehlung herausgefiltert werden könnte. Wir nehmen diese Gefahr jedoch für das Verringern der Zahl der ungültigen Empfehlungen in Kauf. Im zweiten angesprochenen Fall wurde der Film wesentlich besser bewertet als die Empfehlung, d.h. wir gehen davon aus, dass der empfohlene Film auch vom Nutzer schlecht bewertet würde und damit keinen Nutzen für ihn hat. Auch hier besteht die Gefahr, eine interessante Empfehlung zu verlieren, nämlich dann, wenn die vorhergesagte Bewertung falsch ist und die tatsächliche Bewertung besser wäre. Bei der Generierung der Vorhersagen durch den kollaborativen Algorithmus hat sich jedoch gezeigt, dass die vorhergesagten Bewertungen in den meisten Fällen besser waren als die Durchschnittsbewertung für das jeweilige Zielobjekt. Das Restrisiko gehen wir ebenfalls zugunsten einer verminderten Zahl ungültiger Empfehlungen ein. Außerdem bestätigt auch hier die Parameteroptimierung die Überlegungen, denn es wurde ein boolescher Parameter (*use min prediction difference*, s.u.) eingeführt, der es erlaubt zu wählen, ob eine Empfehlung als ungültig herausgefiltert wird, wenn sich die Bewertungen von Empfehlung und bekanntem Film ähnlich sind oder dann, wenn sie sich wesentlich unterscheiden. Auch hier ergab die Optimierung bessere Werte für den Fall, dass sich die Bewertungen ähnlich sind.
Nachdem nun die Gründe für die Wahl des speziellen Hybridsystems ausführlich erläutert wurden, soll die Struktur des restlichen Kapitels derjenigen von Kapitel 5 folgen. Zuerst werden die Kriterien genannt, die zur Wahl der hier speziell benutzten, inhaltlichen Repräsentationsform für Filme geführt haben. Anschließend werden die theoretischen Grundlagen hinter dem gewählten Hybridansatz erläutert, bevor die Implementierung beschrieben wird. Schließlich erfolgt eine Darstellung der Ergebnisse, die sich aus der praktischen Anwendung des Hybridsystems auf die Versuchsumgebung ergeben haben. Insbesondere die Sicht auf die Entwicklung der Interessantheit der Empfehlungen wird benutzt, um beurteilen zu können, ob die erwartete Verbesserung der generierten Filmempfehlungen erreicht wurde.

Weitere Auswahlkriterien - inhaltliche Repräsentation von Filmen

Nach der Wahl des Hybridsystems musste noch für die eigenschaftsbasierte Erweiterung, die - wie der Name schon sagt - mit inhaltlichen Eigenschaften von Filmen arbeitet, eine geeignete Repräsentationsform der Filme bzgl. dieser Eigenschaften gefunden werden. Dazu wurde auf das aus dem *Information Retrieval* bekannte *Vektorraum-Modell* (*VRM*) zurückgegriffen. Dieses wurde während des experimentellen *SMART*-Retrieval-Projekts von [SALTON 1971] entwickelt und später von [RAGHAVAN und WONG 1986] überarbeitet. Dieses Modell wurde deshalb gewählt, weil es sich dabei um ein relativ einfaches Modell handelt, dass zudem sehr anschaulich ist und als eines der populärsten Retrievalmodelle umfangreich erprobt und dafür bekannt ist, sehr gute Ergebnisse zu liefern. Das Übertragen des Modells vom Textretrieval- in den Kontext von Empfehlungssystemen ist zudem sehr einfach.
Die heuristische *TFIDF*-Gewichtung, die ebenfalls im Rahmen des *SMART*-Projekts entwickelt[5] und später bei Anwendung in den experimentellen Systemen *Inquery*[6] und *OKAPI*[7] verfeinert wurde, ermöglicht zusätzlich - übertragen auf den Kontext dieser Arbeit - eine Gewichtung einzelner Filmeigenschaften wie z.B. „Regisseur“ oder „Schauspieler“ anhand der Vorkommenshäufigkeit konkreter Werte dieser Eigenschaften in der betrachteten Menge von Filmen und somit eine höhere Genauigkeit beim Vergleich verschiedener Filme anhand inhaltlicher Gesichtspunkte.
Mithilfe dieser Vektorraum-Repräsentation können Filme bzgl. ihrer Eigenschaften auf einfache Weise miteinander verglichen werden. In Verbindung mit den in den praktischen Versuchen mit dem kollaborativen Algorithmus gewonnenen Erfahrungen werden dann aus den Empfehlungslisten des kollaborativen Ansatzes Filme herausgefiltert, von denen aufgrund der genannten Erfahrungen angenommen wird, dass sie ungültig sind.
Im Folgenden werden nun zuerst die theoretischen Grundlagen des genannten Vektorraum-Modells und der *TFIDF*-Gewichtung im Kontext der Gegebenheiten der Versuchsumgebung dieser Arbeit beschrieben, bevor der Prozess des Herausfilterns potenziell ungültiger Empfehlungen algorithmisch erläutert wird.

[5] Siehe [SALTON und BUCKLEY 1987].
[6] Vgl. [CALLAN et al. 1992].
[7] Siehe [ROBERTSON et al. 1992].

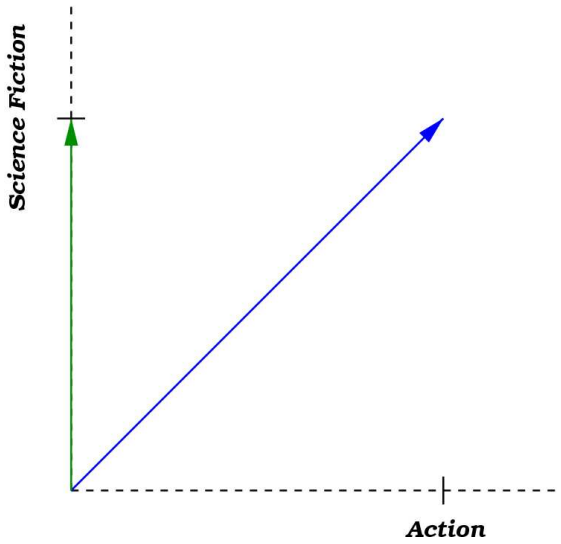


Abbildung 6.1: Vektormodell - Beispiel Genres

Theoretische Grundlagen

Beim Vektorraum-Modell werden inhaltliche Eigenschaften der jeweils betrachteten Objekte - hier Filme - als Punkte in einem Vektorraum repräsentiert, wobei dieser Vektorraum durch die verschiedenen Eigenschaften aufgespannt wird. Zwei Filme werden dann als zueinander ähnlich angesehen, wenn sich die Vektoren der beiden Filme bezogen auf eine gewählte Metrik ähneln.
In der für diese Arbeit implementierten eigenschaftsorientierten Erweiterung des kollaborativen Empfehlungsalgorithmus wurden für den Vergleich von Filmen die sechs Eigenschaften *Genre*, *Regisseur(in)*, *Produzent(in)*, *Produktionsfirma*, *Schauspieler(in)* und *Ursprungsland* herangezogen. Eine nahe liegende Möglichkeit der Transformation von Filmen in Vektoren wäre hierbei gewesen, die sechs erwähnten Eigenschaften als den Vektorraum aufspannende Terme zu interpretieren, spezifischen Werten dieser Eigenschaften eindeutige IDs zuzuweisen und diese IDs als Koordinaten des Films in dem Vektorraum anzusehen. Würde bspw. die Schauspielerin Catherine Hepburn die ID 23 erhalten, so würden Filme, in denen Catherine Hepburn mitwirkt, als Wert für die Eigenschaft *Schauspieler(in)* diese ID und damit die Koordinate 23 für die Dimension *Schauspieler(in)* des Vektorraums zugewiesen bekommen. Diese Art der Interpretation wurde jedoch nicht gewählt, da ihr Problem die Unfähigkeit ist, mit mehreren möglichen Werten für eine Eigenschaft umzugehen. Da jedoch in den meisten Filmen mehrere Schauspieler mitwirken und auch die anderen Eigenschaften mehrfach belegt sein können, musste eine andere Möglichkeit gefunden werden, Filme in Vektoren für das *VRM* umzuwandeln. Außerdem sollte dem Algorithmus die Fähigkeit gegeben werden, die Eigenschaften unterschiedlich zu gewichten um auszudrücken, dass gewisse Eigenschaften für die Ähnlichkeit von Filmen eine hervorgehobene Rolle spielen. Daher wurden die erwähnten sechs Eigenschaften von Filmen beim Vergleich zweier Filme separat betrachtet (d.h. für jede Eigenschaft ein eigener Vektorraum benutzt) und Werte dieser Eigenschaften, wie in der Anwendung des Vektorraum-Modells im Textretrieval meist üblich, als den jeweiligen Eigenschaftsvektorraum aufspannende Terme interpretiert, womit sich eine binäre Werteskala („vorhanden“/“nicht vorhanden“) ergibt. Bezogen auf das vorherige Beispiel würde die Schauspielerin Hepburn nun bspw. als 1. Koordinate des Vektorraums *Schauspieler(in)* interpretiert. Einem Film mit Catherine Hepburn würde dementsprechend als Wert der 1. Koordinate eine 1 zugewiesen, einem Film ohne diese Schauspielerin der Wert 0.
Abbildung 6.1 zeigt ein Beispiel für diese eigenschaftsbasierte Repräsentation mit dem Vektormodell. Dabei wird vereinfachend angenommen, dass es nur die zwei Genres *Action* und *Science Fiction* gibt. Die gestrichelten Linien geben die Vektoren an, die den Vektorraum *Genre* aufspannen, diese Vektoren entspre-

chen gerade den genannten Genres. Der grüne und der blaue Vektor wiederum sind Repräsentationen zweier Filme bzgl. der *Genre*-Eigenschaft. Während der durch die Farbe blau dargestellte Film sowohl dem Genre *Action* als auch dem Genre *Science Fiction* zuzuordnen ist, handelt es sich bei dem durch die Farbe grün repräsentierten Film um einen, der nur dem *Science Fiction*-Genre zugehörig ist. Sei nun M_u die Menge aller Filme aus der großen *MovieLens*-Datenbank, die der Nutzer u im Rahmen der praktischen Experimente dieser Arbeit für den kollaborativen Algorithmus aus Kapitel 5 bewertet hat. Sei weiterhin A_u die Menge aller unterschiedlichen Schauspieler(innen), die in der Gesamtmenge M_u der durch u bewerteten Filme mitwirken. Dann wird ein Film $m_i^u \in M_u$, mit $i = 1, \ldots, |M_u|$ für den Vektorraum *Schauspieler(in)* durch einen Vektor $\vec{a}^{(u,i)}$ repräsentiert, mit $a_j^{(u,i)} \in \{0, 1\}$ und $j = 1, \ldots, |A_u|$. Entsprechend seien $\vec{g}^{(u,i)}$, $\vec{d}^{(u,i)}$, $\vec{p}^{(u,i)}$, $\vec{pc}^{(u,i)}$ und $\vec{c}^{(u,i)}$ die Vektoren des Films m_i^u für die Filmeigenschaften/Vektorräume *Genre*, *Regisseur(in)*, *Produzent(in)*, *Produktionsfirma* und *Herkunftsland*. Dann wird der Film m_i^u durch das Vektortupel $m_i^u = (\vec{g}^{(u,i)}, \vec{d}^{(u,i)}, \vec{p}^{(u,i)}, \vec{pc}^{(u,i)}, \vec{a}^{(u,i)}, \vec{c}^{(u,i)})$ repräsentiert.
Nun muss noch ein Vergleichsmaß für zwei Vektoren $\vec{v}_k$ und $\vec{v}_l$ eines Vektorraums definiert werden. Hierbei wird für den eigenschaftsbasierten Algorithmus auf das bekannte Cosinusmaß zurückgegriffen:

$$\cos(\vec{v}_k, \vec{v}_l) = \frac{\sum \vec{v}_k \vec{v}_l}{\sum \vec{v}_k^2 \cdot \sum \vec{v}_l^2} \tag{6.1}$$

Seien nun w_g, w_d, w_p, w_{pc}, w_a und w_c Gewichte für die Filmeigenschaften *Genre*, *Regisseur(in)*, *Produzent(in)*, *Produktionsfirma*, *Schauspieler(in)* und *Herkunftsland*, R_u die Menge von vorhergesagten Filmempfehlungen, die vom kollaborativen Algorithmus basierend auf den Bewertungen von Nutzer u für die Filme M_u generiert wurde und $r_k^u \in R_u$ mit $k = 1, \ldots, |R_u|$ einer der Filme, die dem Nutzer u empfohlen wurden. Dann wird die Ähnlichkeit zwischen dem bewerteten Film und dem empfohlenen Film berechnet als:

$$\text{sim}(r_k^u, m_i^u) = \frac{1}{6}\left(s_g^{(u,k,i)} + s_d^{(u,k,i)} + s_p^{(u,k,i)} + s_{pc}^{(u,k,i)} + s_a^{(u,k,i)} + s_c^{(u,k,i)}\right) \tag{6.2}$$

mit

$$\begin{aligned}
s_g^{(u,k,i)} &= w_g \cdot \cos\left(\vec{g}^{(u,k)}, \vec{g}^{(u,i)}\right) \\
s_d^{(u,k,i)} &= w_d \cdot \cos\left(\vec{d}^{(u,k)}, \vec{d}^{(u,i)}\right) \\
s_p^{(u,k,i)} &= w_p \cdot \cos\left(\vec{p}^{(u,k)}, \vec{p}^{(u,i)}\right) \\
s_{pc}^{(u,k,i)} &= w_{pc} \cdot \cos\left(\vec{pc}^{(u,k)}, \vec{pc}^{(u,i)}\right) \\
s_a^{(u,k,i)} &= w_a \cdot \cos\left(\vec{a}^{(u,k)}, \vec{a}^{(u,i)}\right) \\
s_c^{(u,k,i)} &= w_c \cdot \cos\left(\vec{c}^{(u,k)}, \vec{c}^{(u,i)}\right)
\end{aligned}$$

Die Ähnlichkeit zwischen zwei Filmen ergibt sich also als durch die sechs Filmeigenschaften gemittelte Summe aus den gewichteten, durch das Cosinusmaß berechneten Ähnlichkeiten der Filme bzgl. der einzelnen Filmeigenschaften.
Zusätzlich kann nun noch mittels der angesprochenen *TFIDF*-Heuristik eine Gewichtung der einzelnen Eigenschaftswerte, wie z.B. des Genres *Action* vorgenommen werden. Sei bspw. der von Nutzer u bewertete Film $m_i^u \in M_u$ mit $i = 1, \ldots, |M_u|$ und dem Vektor $\vec{g}^{(u,i)}$ für die Filmeigenschaft *Genre* gegeben. Es soll nun die Gewichtung $w_{g_j^{(u,i)}}$ für die Eigenschaft $g_j^{(u,i)}$ des Genrevektors $\vec{g}^{(u,i)}$ mit $j = 1, \ldots, |G_u|$ und $g_j^{(u,i)} \in \{0, 1\}$ berechnet werden. Diese ergibt sich aus dem Produkt der sog. *normalisierten Termfrequenz* ntf und der *inversen Dokumentfrequenz* idf, die der Gewichtung auch ihren Namen (*(n)tf-idf*) gegeben haben:

$$w_{g_j^{(u,i)}} = ntf_j \cdot idf_j \tag{6.3}$$

Die inverse Dokumentfrequenz bzw. in unserem Fall die inverse Filmfrequenz gewichtet eine Eigenschaft umso höher, je seltener sie in der Menge M_u der vom Nutzer u bewerteten Filme vorkommt. Dahinter steckt der Gedanke, dass Filme mit seltener vorkommenden Eigenschaften die Aufmerksamkeit eines Nutzers mehr erregen, d.h. eine höhere Wichtigkeit haben, als Eigenschaften, die in fast allen Filmen vorkommen,

die ein Nutzer kennt.

$$idf_j = \frac{log\frac{|M_u|}{n_j}}{|M_u| + 1} \tag{6.4}$$

Dabei bezeichnet n_j die Anzahl der Filme in M_u, in denen die Eigenschaft j (das spezifische Genre) vorkommt.
Die normalisierte Termfrequenz schließlich gewichtet Terme (Filmeigenschaften) entsprechend ihrer Vorkommenshäufigkeit im aktuell betrachteten Film m_i^u. Dabei wird die Frequenz bzgl. der Gesamtzahl der Eigenschaften im aktuellen Film normalisiert. D.h. in diesem speziellen Fall findet eine Normalisierung bzgl. der Anzahl der unterschiedlichen Genres, die dem Film zugeordnet sind, statt.

$$ntf_j = \frac{tf_{ij}}{tf_{ij} + 0.5 + 1.5 \cdot \frac{l_i}{al}} \tag{6.5}$$

tf_{ij} ist dabei die Vorkommenshäufigkeit der Eigenschaft (des Genres) j im Film m_i^u. Da wir mit binären Eigenschaftswerten arbeiten, ist $tf_{ij} \in \{0, 1\}$. l_i gibt die Länge, d.h. die Anzahl der Eigenschaften (Genres), die dem Film m_i^u zugeordnet sind an, während al die durchschnittliche Länge, also die durchschnittliche Anzahl der Eigenschaften (Genres), die einem Film aus M_u zugeordnet sind, bezeichnet. Entsprechend dem Beispiel können auch die Gewichtungen für einzelne Produktionsfirmen, Länder, usw. berechnet werden. Bei Benutzung der *TFIDF*-Gewichtung folgt die Ähnlichkeitsberechnung von zwei Filmen demselben oben angegebenen Prinzip, nur das statt des Wertes 1 der binären Werteskala (Eigenschaft kommt vor) die errechneten *TFIDF*-Gewichte benutzt werden (1· *TFIDF*-Gewicht = *TFIDF*-Gewicht).
Nachdem die Grundlagen für einen Vergleich zweier Filme bzgl. inhaltlicher Eigenschaften gelegt wurden, soll nun der genaue algorithmische Ablauf des Herausfilterns ungültiger Empfehlungen aus der durch den kollaborativen Algorithmus generierten Empfehlungsliste dargestellt werden, der auf den Überlegungen am Anfang des Kapitels beruht:

- *Für alle Nutzer $u \in U$:*
  - *Setze sim_{min}*
  - *Lies die Filmbewertungen M_u, sowie die Empfehlungsliste R_u des kollaborativen Algorithmus für Nutzer u ein.*
  - *Für jede Filmempfehlung $r_k^u \in R_u$ der Empfehlungsliste:*
    - $\text{sim}_{max} = 0$
    - $m_{sim} = \emptyset$
    - *Für jeden vom Nutzer bewerteten Film $m_i^u \in M_u$:*
      - *Berechne sim(r_k^u, m_i^u)*
      - *Wenn sim$(r_k^u, m_i^u) > sim_{min}$ AND sim$(r_k^u, m_i^u) > sim_{max}$, setze sim_{max} = sim(r_k^u, m_i^u) und $m_{sim} = m_i^u$*
    - *Wenn $m_{sim} = \emptyset$, entferne r_k^u aus R_u und gehe zu* MARKE
    - *Wenn $abs(rating(r_k^u) - rating(m_{sim})) > 1$, dann entferne r_k^u aus R_u*
    - *MARKE*

6.1 Implementierung

Nachdem die theoretischen Grundlagen für die eigenschaftsorientierte Erweiterung erläutert wurden, kann nun deren konkrete Implementierung beschrieben werden. Analog zur Struktur von Kapitel 5 werden dabei zuerst die wichtigsten Parameter des eigenschaftbasierten Algorithmus aufgelistet und die anhand einer Parameteroptimierung ermittelten Werte, die auch in der praktischen Anwendung des Algorithmus benutzt wurden, genannt. Danach erfolgt eine Darstellung der Probleme, die bei der Implementierung auftraten und den zur Lösung dieser Probleme erstellten Optimierungen.

6.1.1 Parameter des Algorithmus

Das Herausfiltern potenziell uninteressanter Empfehlungen kann durch diverse Parameter des eigenschaftsorientierten Algorithmus beeinflusst werden. Im Gegensatz zum kollaborativen Algorithmus aus Kapitel 5 in Verbindung mit dem großen *MovieLens*-Datensatz konnte hier eine angemessene Optimierung der Parameter durch ein *Brute-Force*-Verfahren vorgenommen werden, wobei auf die Interessantheitsbewertungen der Versuchspersonen bzgl. der vom kollaborativen Algorithmus generierten Empfehlungen zurückgegriffen wurde. Bei der Optimierung galt es, eine Kombination von Parameterwerten zu finden, die es dem Algorithmus ermöglicht, möglichst viele der von den Nutzern als interessant bewertete Empfehlungen unangetastet zu lassen, während gleichzeitig möglichst viele der als ungültig klassifizierten Empfehlungen herausgefiltert werden sollen. Die möglichen Werte wurden dabei für jeden Parameter fest vorgegeben, das *Brute-Force*-Verfahren bestand darin, Instanzen des Algorithmus für alle möglichen Kombinationen der Parameterwerte zu erzeugen und deren Qualität bzgl. des gewählten Optimierungsmaßes zu ermitteln. Die Kombination von Parameterwerten mit dem besten Qualitätsergebnis wurde dann für die endgültige Anwendung des Algorithmus benutzt. Die vorgegebenen möglichen Parameterwerte werden dabei weiter unten bei den Beschreibungen der Parameter aufgeführt.
Vor der Optimierung mussten allerdings einige Tücken in Form der Wahl des genauen Optimierungsmaßes überwunden werden. Grundsätzlich neigt der eigenschaftsorientierte Algorithmus bei Benutzung solcher Standardmaße wie *accuracy*, *Precision* oder *Recall* dazu, entweder alle interessanten Empfehlungen zu übernehmen, dafür aber viele ungültige und uninteressante Empfehlungen ebenfalls als interessant einzustufen oder die ungültigen und uninteressanten Empfehlungen herauszufiltern, dafür aber kaum interessante Empfehlungen in die gefilterte Empfehlungsliste zu übernehmen. Daher wurde bei der Optimierung die Rate der korrekt als (un)interessant[8] klassifizierten Empfehlungen („true positive“ bzw. „true negative“) betrachtet. Um große Werte der beiden Raten stärker zu gewichten und kleine Werte stärker zu bestrafen, wurden sie vor der Mittelung zum Quadrat genommen. Außerdem sollte dem Beibehalten interessanter Bewertungen ein größeres Gewicht zugesprochen werden. Sei U die Menge aller Nutzer, deren Empfehlungslisten betrachtet werden und R_u die Liste von Empfehlungen für Nutzer $u \in U$. Dann wurde die Vorhersagequalität PQ_u des eigenschaftsorientierten Algorithmus für Nutzer u berechnet als:

$$PQ_u = \frac{1}{2} \cdot \left((IA_u \cdot 1.5)^2 + UA_u^2 \right) \tag{6.6}$$

mit der Vorhersagegenauigkeit IA_u für interessante und UA_u für uninteressante Empfehlungen, definiert als

$$\begin{aligned} IA_u &= \tfrac{TP}{TP+FP} \\ UA_u &= \tfrac{TN}{TN+FN} \end{aligned}$$

wobei TP der Anzahl korrekt als interessant klassifizierter („true positive“) und FP der Anzahl nicht korrekt als interessant klassifizierter („false positive“) Empfehlungen entspricht und entsprechend TN für „true negative“ und FN für „false negative“ bzgl. der Uninteressantheit von Empfehlungen stehen.
Die Gesamtvorhersagequalität für alle Nutzer U wurde dann als gemittelte Summe der einzelnen Vorhersagegenauigkeiten berechnet:

$$PQ_{all} = \frac{1}{|U|} \cdot \sum_{\forall u \in U} PQ_u \tag{6.7}$$

Die anhand dieses Maßes vorgenommene Parameteroptimierung basierte dabei auf den Daten von $|U| = 33$ Nutzern, die zu dieser Zeit ihre Interessantheitsbewertungen für die Empfehlungen des kollaborativen Algorithmus bereits abgeschlossen hatten. Für die Optimierung und das anschließende Testen wurden dabei dieselben Daten benutzt, so dass die Gefahr eines *Overfittings* besteht. Da hier jedoch kein eigentliches Modell gelernt wurde,[9] handelt es sich nur um ein Parameteroverfitting.

[8] Wegen der binären Klassifikation werden sowohl ungültige, als auch uninteressante Empfehlungen als „uninteressant“ bezeichnet.
[9] Die Filmbewertungen der Nutzer werden zwar in die Vektorraum-Repräsentation überführt, dafür ist aber kein Lernen erforderlich.

Die einzelnen Parameter des eigenschaftsorientierten Algorithmus, sowie die möglichen Werte für die Optimierung und die letztendlich aufgrund des Optimierungsergebnisses gewählten Parameterwerte ergeben sich dann wie folgt:

- *minimal similarity threshold* (real) (mögliche Werte für Optimierung: $0, 0.5, 0.7, 1$) - Gibt den Wert an,[10] der bei der Ähnlichkeitsberechnung von Filmen überschritten werden muss, damit beide Filme als ähnlich angesehen werden. Dieser Wert wurde auf 0 gesetzt, da Werte größer 0 bei der Parameteroptimierung schlechtere Ergebnisse erbrachten.

- *use min prediction difference* (boolean) (mögliche Werte für Optimierung: *true, false*) - Ermöglicht die wahlweise Einstellung, dass Empfehlungen dann als interessant angesehen werden, wenn die Differenz zwischen der Bewertung der Empfehlung und der Bewertung des zur Empfehlung ähnlichsten Films einen bestimmten Wert nicht unter- (= *true*) oder überschreitet (= *false*). Die Parameteroptimierung zeigte, dass das Setzen dieses Wertes auf *true* schlechtere Ergebnisse bringt. Dies bestätigte die anhand der Ergebnisse aus Kapitel 5.2 erstellte Hypothese am Anfang des Kapitels, dass empfohlene Filme nicht nur bzgl. ihrer Eigenschaften ähnlich zu den initial bewerteten Filmen eines Nutzers sein müssen, um akzeptiert zu werden, sondern auch ihre Bewertungen nicht zu sehr voneinander abweichen dürfen.

- *prediction difference* (real) (mögliche Werte für die Optimierung: $0.0 - 4.5$, Schrittweite 0.5) - Hier kann nun die genaue Differenz zwischen der tatsächlichen Bewertung eines Films durch den Nutzer und der vorhergesagten Bewertung für einen anderen Film angegeben werden, die zusammen mit dem vorherigen Parameter bestimmt, ob eine Empfehlung beibehalten oder verworfen wird. Auch hier bestätigte die Parameteroptimierung die erstellten Hypothesen. Der optimale Wert von 1.0 entspricht genau der maximalen Abweichung, die durch die in den Versuchen erfahrene „fuzzy“-Bewertung von Nutzern (dieselben Filme werden bei mehrmaliger Bewertung oft unterschiedlich bewertet, die Abweichung beträgt angesichts der fünfwertigen Skala jedoch nie mehr als 1) definiert wurde.

- *use TFIDF* (boolean) (mögliche Werte für die Optimierung: *true, false*) - mit diesem Parameter kann die oben beschriebene *TFIDF*-Gewichtung ein- (= *true*) bzw. ausgeschaltet (= *false*) werden. Bei eingeschalteter Gewichtung ergaben sich schlechtere Werte, so dass im endgültigen Algorithmus ohne gearbeitet wurde.

Es folgen Parameter für die Gewichtung der einzelnen Filmeigenschaften. Alle diese Gewichtungen wurden durch den *real*-Datentyp ausgedrückt, die Parameteroptimierung wurde auf die möglichen Werte $0.1, 0.5, 0.7$ und 1.0 beschränkt, da bei einer feineren Skalierung der Zeitaufwand der Optimierung trotz Aufteilung auf mehrere Rechner zu groß und somit die für die Nutzer verbleibende Bewertungszeit für die Empfehlungslisten zu kurz gewesen wäre.

- *genre weight* - Die Gewichtung w_g für das Ergebnis der Ähnlichkeitsberechnung für Empfehlung und Film bzgl. der ihnen zugeordneten Genres. Die Parameteroptimierung erbrachte hier als optimalen Wert 1.0, d.h. dem Genre von Filmen wird bei der Ähnlichkeitsberechnung großes Gewicht eingeräumt.

- *director weight* - Die entsprechende Ähnlichkeitsgewichtung w_d für die Filmeigenschaft *Regisseur(in)*. Auch diese Eigenschaft ist für die Ähnlichkeitsberechnung wichtig. Werte kleiner als 1.0 brachten hier schlechtere Ergebnisse.

- *producer weight* - Wie zuvor, jedoch für die Filmeigenschaft *Produzent(in)*. Die Produzentin scheint (über alle Nutzer gesehen) nicht so entscheidend für die Ähnlichkeit von Filmen zu sein. Ein Gewichtswert $w_p = 0.7$ erzeugte hierbei optimale Werte.

- *production company weight* - Die Gewichtung w_{pc} der Produktionsfirma scheint absolut nicht entscheidend für die Ähnlichkeit zu sein, da hier der niedrigste Wert 0.1 die besten Ergebnisse brachte.

[10] sim_{min} aus der Ablaufbeschreibung des Algorithmus oben.

- *actor weight* - Wie erwartet, sind die in Filmen mitwirkenden Schauspieler sehr wichtig für die Ähnlichkeitsberechnung, was sich in einem optimalen Wert von 1.0 widerspiegelt.

- *country* - Das Herkunftsland eines Films scheint für die meisten Nutzer kaum eine Rolle zu spielen. Werte von $w_c > 0.1$ brachten hier durchweg schlechtere Ergebnisse ein.

6.1.2 Implementierungsprobleme und Optimierungen

Auch der eigenschaftsbasierte Algorithmus hatte in der Praxis mit Performance-Problemen zu kämpfen, die hier durch die Abfrage der lehrstuhleigenen *Oracle*-Datenbank mit *IMDB*-Filmdaten hervorgerufen wurde. Um für einen Zielfilm aus der Empfehlungsliste eine Vorhersage bzgl. der Interessantheit zu generieren, muss beim eigenschaftsorientierten Ansatz dieser empfohlene Film mit jedem Film verglichen werden, den der Zielnutzer zuvor bewertet hat. Für den inhaltlichen Vergleich müssen für jeden dieser Vergleichsfilme die Einträge für die Genre, Regisseure, Schauspieler, Produzenten, produzierenden Firmen und die Ursprungsländer mittels *SQL*-Anfragen aus der Datenbank extrahiert werden. Diese Anfragen über ein *Java*-Interface zu stellen kostet dabei Zeit. Bei einem Datenbank-Server *E250* mit 2 *UltraSparc II*-Prozessoren und 400 MHz CPU-Leistung, sowie 1664 MB Hauptspeicher und einer *Oracle 8i Enterprise Edition* mit der Release-Nr. 8.1.6.0.0 und - für den eigenschaftsorientierten Empfehlungsalgorithmus - einem Rechner mit der in Kapitel 5.1.2 definierten Leistung, betrug in Tests die Zeitdauer für die Generierung einer Interessantheitsvorhersage bei 100 vom Zielnutzer bewerteten Filmen ungefähr 1060 Sekunden, also ca. 17.7 Minuten. Bei Nutzern mit mehr Bewertungen (ein „Power-User“ unter den Versuchspersonen hatte über 900 Filme bewertet) würde sich die benötigte Zeit entsprechend erhöhen.
Selbst wenn man die Daten für die vom Nutzer bewerteten Filme nach einmaliger Datenbankextraktion cachen würde, was sinnvoll wäre, so müssten noch die entsprechenden Daten für jeden einzelnen Film aus der Empfehlungsliste abgefragt werden, damit ein Vergleich durchgeführt werden könnte.
Insbesondere sind bei Datenbankabfragen viele einfache Abfragen zeitkritischer als eine sehr komplexe Abfrage, für die große Datenbanktabellen durchsucht werden müssen. Durch Benutzung des Java-Interfaces entsteht zudem ein weiterer Zeitfaktor, der bei jeder der einzelnen Anfragen von neuem anfällt. Gehen wir nun exemplarisch für die Abschätzung der Zeitdauer bei der Interessantheitsvorhersage für alle Filme einer Empfehlungsliste vom zuvor genannten Beispiel aus, so müssten Interessantheitsbewertungen für $3900 - 100 = 3800$ Filme vorhergesagt werden. Bei einer durchschnittlichen Abfragezeit für die Filmdaten von $\frac{1060}{100} = 10.6$ Sekunden läge die Vorhersagedauer damit bei $10.6 \cdot 3800 = 40280$ Sekunden, d.h. ca. 11.2 Stunden.

Laufzeitoptimierung durch Caching und Hashing

Aufgrund der genannten Probleme wurden die Daten für alle 3900 Filme aus der Versuchsumgebung einmalig aus der *Oracle*-Datenbank extrahiert, in eine interne Datenstruktur überführt und in einer Datei gespeichert. Das Caching der Daten durch Einlesen dieser Datei nimmt dabei nur ca. 0.5 Sekunden in Anspruch und muss für die Generierung von Vorhersagen für die Empfehlungslisten mehrerer Nutzer nur einmalig durchgeführt werden.
Die interne Datenstruktur arbeitet dabei mit Hashing. So kann über Angabe der ID eines Films schnell auf eine Struktur bestehend aus Listen von Werten für die einzelnen Filmeigenschaften wie Genre oder Schauspieler zugegriffen werden. Die Werte der Eigenschaften entsprechen dabei wiederum IDs , wie sie in den entsprechenden Datenbanktabellen vergeben wurden. Für einen Vergleich von Filmen sind diese IDs ausreichend und praktischer, sowie sicherer zu vergleichen als Zeichenketten.
Mithilfe dieser Optimierung senkte sich für das oben angegebene Beispiel mit 100 bewerteten Filmen die Dauer für eine Interessantheitsvorhersage von 1060 Sekunden auf knapp 1 Sekunde, wobei die nur einmal benötigte Ladezeit von 0.5 Sekunden für die Eigenschaftsdaten aller Filme der Datenbank in diesem Wert enthalten ist. Für die Vorhersage der Interessantheit für alle 3800 Empfehlungen benötigte die verbesserte Version des Algorithmus ca. 579 Sekunden, d.h. fast 10 Minuten und somit eine annehmbare Zeit für die Versuchsumgebung. Wie schon beim kollaborativen Algorithmus wird auch hier das Caching und die damit

gesteigerte Vorhersagegeschwindigkeit durch einen erhöhten Hauptspeicherverbrauch erkauft, der jedoch bei weitem nicht so hoch ist wie im kollaborativen Fall.

6.2 Versuchsergebnisse

Es folgen die Ergebnisse, die mit dem vorgestellten eigenschaftsbasierten Filteraufsatz für den kollaborativen Algorithmus erzielt wurden. Dabei steht natürlich das Ergebnis der Interessantheitsbewertungen der Nutzer für die Empfehlungslisten des Hybridsystems im Vordergrund. Es wird aber auch untersucht, ob die in den Algorithmus eingeflossenen Überlegungen zu Anfang dieses Kapitels die Divergenz der Empfehlungslisten für die verschiedenen Nutzer erhöht haben. Abschließend folgen einige Anmerkungen zu den hier gemachten Erfahrungen mit praktischen Versuchen im allgemeinen.

Klassifikationsgenauigkeit für Interessantheitsbewertungen

Die mit dem in Abschnitt 6.1.1 definierten Bewertungsmaß ermittelte optimale (hypothetische) Vorhersagequalität im Sinne einer Klassifikation der Empfehlungen in interessante und nicht interessante (d.h. uninteressante und ungültige) Empfehlungen ist schlecht. Als Werte für die Klassifikationsgenauigkeit für alle Nutzer[11] ergeben sich $IA_{all} = 0.62$ und $UA_{all} = 0.47$. Damit ist die Genauigkeit nur minimal höher als man es bei einem Algorithmus erwarten würde, der Empfehlungen zufällig als interessant bzw. uninteressant klassifiziert.
Dieses schlechte Ergebnis hat mehrere Gründe. Zum einen schwankt die Vorhersagequalität zwischen einzelnen Nutzern sehr stark. Dies liegt daran, dass die Parameteroptimierung nur für alle Nutzer insgesamt ausgeführt wurde.[12] Tests, bei denen für einige Nutzer separate Parameteroptimierungen ausgeführt wurden, ergaben teilweise völlig andere optimale Parameterwerte und wesentlich bessere Vorhersagequalitäten. Außerdem wurde die Menge der möglichen Werte bei der Parameteroptimierung - ebenfalls aus Zeitgründen - beschränkt. Eine feinere Skalierung insbesondere bei den Gewichtswerten für die einzelnen Filmeigenschaften hätte nach Ansicht des Autors ebenfalls Verbesserungen bringen können. Ein weiterer Grund für das schlechte Ergebnis sind Datenlücken. Die benutzte *Oracle*-Datenbank mit den Daten der *IMDB* weist für viele Filme nur unvollständige Informationen auf. Da diese Datenbank mit einer älteren Offline-(Text-)Version der *IMDB* erstellt wurde, kann nicht bestimmt werden, ob schon die Offline-Version diese Datenlücken aufwies oder nicht alle Daten der Offline-Version ein gepflegt werden konnten. Für viele Filme fehlten in der *Oracle*-Datenbank selbst so grundlegende Informationen wie mitwirkende Schauspieler oder Regisseur. Oft konnten Ähnlichkeitsvergleiche nur anhand von Genre-Informationen durchgeführt werden. Auf diesem Fehlen von Daten könnte auch das schlechte Abschneiden der *TFIDF*-Gewichtung bei der Optimierung beruhen, da diese Gewichtungsform erst bei einer größeren Menge von Daten gute Ergebnisse bringt. Hypothese ist hier deshalb, dass der eigenschaftsorientierte Algorithmus mit vollständigen Daten und auf einzelne Nutzer hin optimierte Parameter bessere Ergebnisse bringen würde.

Interessantheit der Empfehlungen

Trotz der genannten eher mäßigen Vorhersageergebnisse des verbesserten Algorithmus für die Interessantheitsklassifizierung ist die letztendliche Verbesserung der Interessantheit der Empfehlungen in den praktischen Tests sehr stark.[13] Mit einer Durchschnittsbewertung von ca. 3.24 im Gegensatz zu ca. 2.91 beim kollaborativen Algorithmus hat sich eine Verbesserung um 0.33 Punkte ergeben. Stärker noch fällt die Verbesserung ins Gewicht, wenn man sich die Verteilung der einzelnen Bewertungen in Abb. 6.2 anschaut.

[11] D.h. die 33 Nutzer, deren Daten auch für die Parameteroptimierung verwendet wurden.
[12] Der Zeitaufwand für eine Einzeloptimierung bezogen auf alle Nutzer wäre selbst bei einer Verteilung auf mehrere Rechner zu zeitintensiv gewesen.
[13] Was wiederum dafür spricht, dass hypothetische Qualitätsmaße praktische Tests nicht ersetzen können.

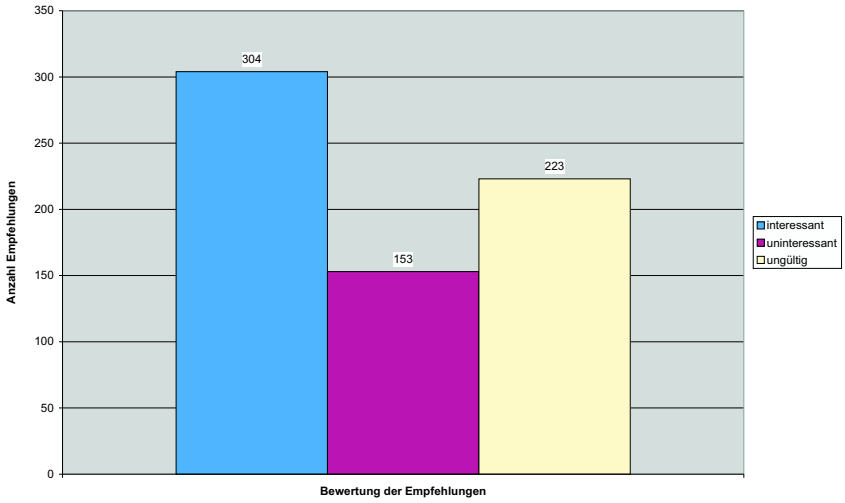


Abbildung 6.2: Interessantheitsbewertungen - Hybridsystem

Im Gegensatz zum kollaborativen Algorithmus überwiegt jetzt die Anzahl der interessanten Empfehlungen deutlich. Der prozentuale Anteil der interessanten Empfehlungen (Tabelle 6.1) hat gegenüber dem kollaborativen Algorithmus um 8% von 36% auf 44% zugenommen, während der Anteil der ungültigen Empfehlungen von 41% auf 33%, d.h. um 8% gesunken ist. Der Anteil der gültigen, aber uninteressanten Empfehlungen ist dagegen mit 23% gleich geblieben.

Interessantheitsbewertungen						
	absolut			**prozentual**		
Typ	**kollaborativ**	**Hybrid**	**Differenz**	**kollaborativ**	**Hybrid**	**Differenz**
interessant	248	304	+56	36	44	+8
uninteressant	154	153	−1	23	23	+/ − 0
ungültig	278	223	−55	41	33	−8

Tabelle 6.1: Interessantheitsbewertungen kollaborativ & Hybrid - Zusammenfassung

Der direkte Vergleich der Bewertungen für beide Algorithmen in Abb. 6.3 zeigt die Trendwende besondersdeutlich. Das Verhältnis zwischen den interessanten Bewertungen (ganz links) und den ungültigen Bewertungen (ganz rechts) hat sich umgekehrt, während sich bei den gültigen aber uninteressanten Bewertungen (Mitte) kaum eine Änderung ergeben hat. Die Interessantheitsbewertungen der einzelnen Nutzer zeigen bis auf ganz wenige Ausnahmen bei allen Nutzern eine Verbesserung. Bei einigen Personen ist diese Verbesserung nur minimal (dies sind insbesondere diejenigen, bei denen die hypothetische Klassifikationsgenauigkeit entsprechend schlecht ausfällt), bei anderen dagegen sehr stark.
Dass bei fast allen Empfehlungslisten eine Verbesserung der nutzerbewerteten Interessantheit stattgefunden hat, muss insbesondere deswegen stark gewichtet werden, weil der eigenschaftsorientierte Filteraufsatz wegen der erwähnten, relativ schlechten Klassifikationsgenauigkeit für die als interessant klassifizierten Empfehlungen des kollaborativen Algorithmus nicht automatisch alle diese interessanten Empfehlungen in die neuen Empfehlungslisten übernommen hat. Vielmehr finden sich unter den Top-20-Positionen der neuen Empfehlungslisten auch einige Empfehlungen, die von den Nutzerinnen während der Testphase mit dem ursprünglichen kollaborativen Empfehlungsalgorithmus als uninteressant oder sogar ungültig bewertet wurden, während einige interessante Empfehlungen leider herausgefiltert wurden. Somit ist selbst eine anscheinend nur minimale Verbesserung der Interessantheit um eine zusätzlich als interessant bewertete

Empfehlung letzten Endes mehr als nur minimal, da sich unter den als interessant klassifizierten Empfehlungen des zweiten Bewertungsdurchgangs mehr als eine neu hinzugekommene Empfehlung befindet. Auch die persönlichen Reaktionen der Nutzer waren entsprechend positiv. Viele gaben an, im Gegensatz zum kollaborativen Verfahren einen Bezug zu ihren abgegebenen Filmbewertungen für das Erstellen der Nutzerpräferenz zu sehen, ohne dass die neuen empfohlenen Filme offensichtliche Empfehlungen („inside the box“-Problem von rein eigenschaftsorientierten Verfahren) dargestellt hätten.[14]
Die eigenschaftsorientierte Erweiterung als Implementierung der im Zusammenhang mit den praktischen Versuchen zum kollaborativen Algorithmus gemachten Erfahrungen aus Kapitel 5.2 hat somit die Richtigkeit der aus diesen Erfahrungen gezogenen Schlüsse bestätigt. Die Fähigkeit kollaborativer Verfahren, Nischenfilme zu empfehlen, darf nicht ohne eine gewisse vorhandene Bindung der Eigenschaften dieser Nischenfilme zu denen der Filme eingesetzt werden, die die entsprechenden Nutzerinnen zuvor hoch bewertet haben, da sonst die Gefahr groß ist, dass es sich bei den Empfehlungen um ungültige Empfehlungen handelt, weil die nach der in Kapitel 3.3 erstellten Definition des Interessantheitsmaßes notwendige Bedingung dieses Qualitätsmaßes, die Vorhersagegenauigkeit, nicht erfüllt ist. Dies bestätigt zum einen den Sinn dieser dreistufigen Interessantheitsdefinition und zum anderen die bereits mehrfach erwähnte Meinung, dass die praxisorientierten Qualitätsmaße die hypothetischen Maße nicht ersetzen, sondern als notwendige, auf sie aufbauende Ergänzungen fungieren sollen, um eine aussagekräftige Qualitätsbewertung von Empfehlungssystemen durchführen zu können.
Zusammengefasst zeigt sich das Hybridverfahren dem rein kollaborativen Ansatz gegenüber also als weit überlegen. Die Hybridisierung sollte in diesem Zusammenhang insbesondere deswegen weiterverfolgt werden, da das Ergebnis der Parameteroptimierung durch Beheben der genannten Probleme (Optimierung global statt nutzerbezogen, Datenlücken) noch verbesserungsfähig ist.

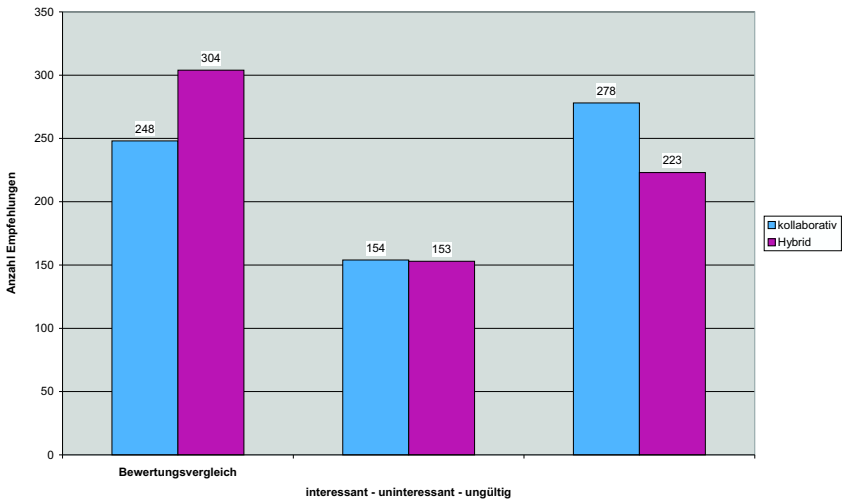


Abbildung 6.3: Interessantheitsbewertungen - Vergleich kollaborativer Algorithmus und Hybridsystem

Divergenz der Empfehlungslisten

In Kapitel 5.2 wurde als ein weiteres Ergebnis der praktischen Versuche mit dem kollaborativen Algorithmus das Problem ähnlicher Empfehlungslisten für die verschiedenen Versuchspersonen genannt. Diese Situation lässt sich durch Einsatz des Hybridsystems ebenfalls verbessern. Die Abb. 6.4 und 6.5 zeigen den Unterschied zwischen den Verteilungen der Film-IDs der auf Rang 1 platzierten Empfehlungen für

[14] Auch dies bestätigt die Wichtigkeit praktischer Versuche, da solche Anmerkungen von Anwendern nicht durch Tests unter Laborbedingungen gewonnen werden können.

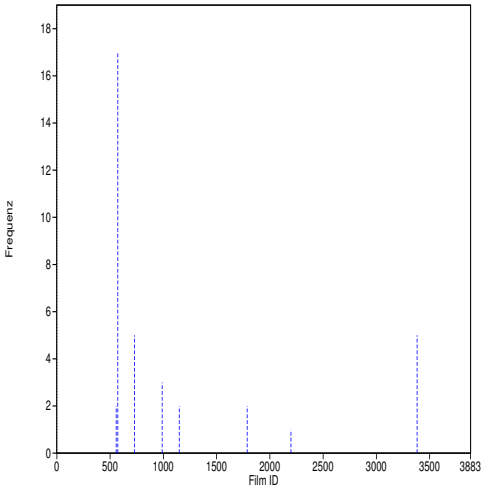


Abbildung 6.4: Verteilung Film IDs Rang 1, kollaborativer Algorithmus

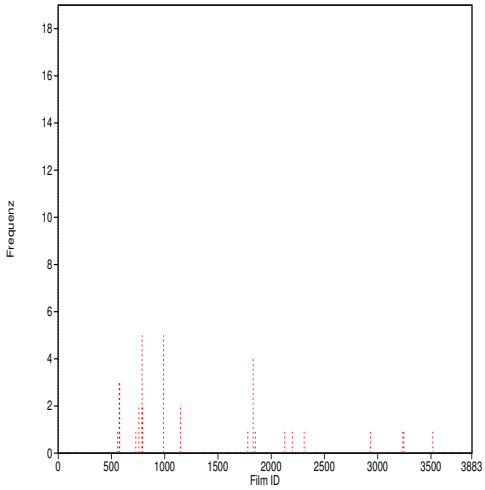


Abbildung 6.5: Verteilung Film IDs Rang 1, Hybridsystem

den kollaborativen Algorithmus und das Hybridsystem bezogen auf alle Nutzer. Beim Hybridsystem ist die Frequenz einzelner Film-IDs wesentlich geringer als beim kollaborativen Ansatz, dafür finden sich mehr verschiedene Filme auf dem ersten Rang. Dasselbe Bild zeigt sich bei Betrachtung des letzten bewerteten Rangs (Abb. 6.6 und 6.7). Auch die Verteilung der Film-IDs über alle 20 Ränge und Nutzer in den Abbildungen 6.8 und 6.9 verdeutlicht die Verbesserung durch das Hybridsystem. Der kollaborative Algorithmus ist durch den eigenschaftsorientierten Aufsatz wesentlich besser in der Lage, personalisierte Empfehlungslisten zu erzeugen, als durch alleinigen Einsatz.

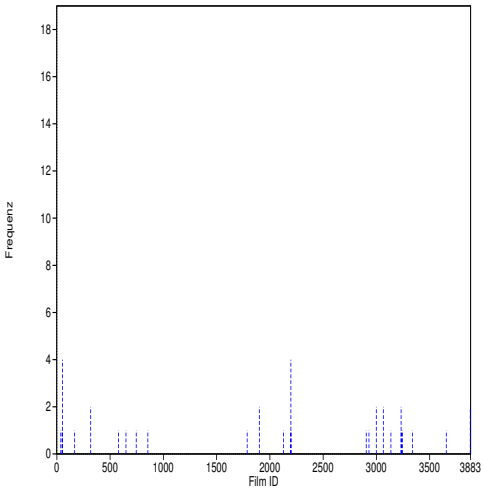


Abbildung 6.6: Verteilung Film IDs Rang 20, kollaborativer Algorithmus

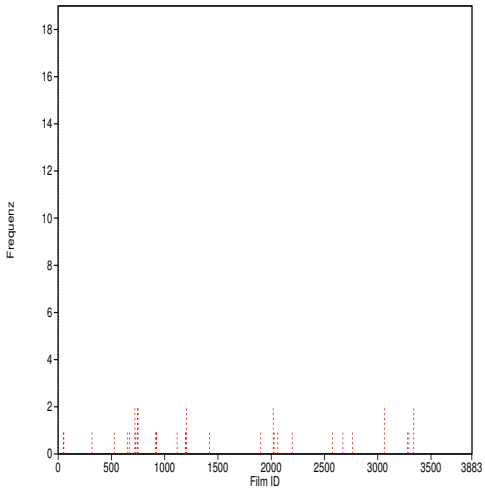


Abbildung 6.7: Verteilung Film IDs Rang 20, Hybridsystem

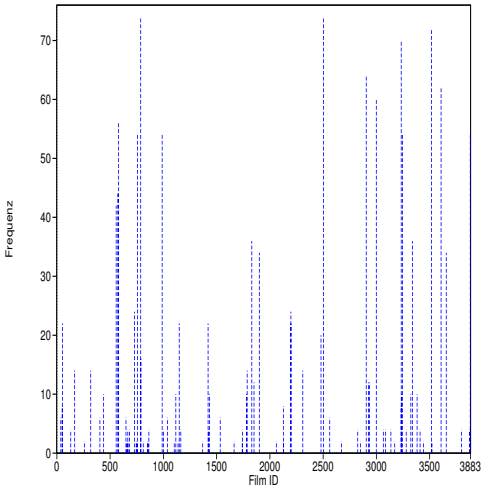


Abbildung 6.8: Verteilung Film IDs Rang 1-20, kollaborativer Algorithmus

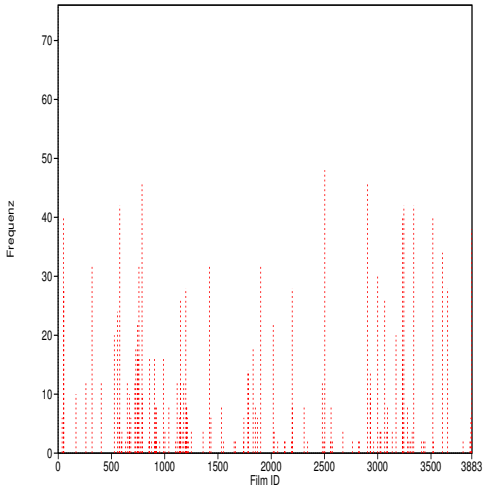


Abbildung 6.9: Verteilung Film IDs Rang 1-20, Hybridsystem

Nutzerbeteiligung und Zeitaufwand der praktischen Versuche

Wie in Abschnitt 5.2 erwähnt, war die Beteiligung der Versuchspersonen an den verschiedenen Bewertungsdurchgängen (initiale Filmbewertung zur Erstellung der Nutzerpräferenzen, Interessantheitsbewertung für kollaborativen/Hybridalgorithmus) nicht konstant. Besonders extrem war das Absinken der Beteiligung von der initialen Filmbewertung zur ersten Interessantheitsbewertung um 18%. Dies mag vielleicht damit zusammenhängen, dass die initiale Filmbewertung mit einem konkreten Nutzen verbunden war, da auf diese Filmbewertungen die Empfehlungen folgten. Die Interessantheitsbewertung hatte jedoch für die Anwender keinen direkten Nutzen mehr. Das Absinken der Beteiligung von der ersten zur zweiten Interessantheitsbewertung fiel dann auch wesentlich geringer aus. Der anfängliche Verlust von Versuchspersonen mag auch damit zusammenhängen, dass der *MovieVoter* (siehe Abschnitt 4.4), das Programm, mit dem die Nutzerinnen Bewertungen abgeben und Empfehlungen betrachten konnten, häufige Überarbeitungen erfuhr. Diese Überarbeitungen dienten dazu, Anregungen der Versuchspersonen, was gewünschte Verbesserungen des Programms anging, so schnell wie möglich einzubauen und entdeckte Fehler zu beseitigen. Auch die anfänglich recht komplizierte manuelle Installation wurde durch Verbesserungen voll automatisiert. Das für ein Update erforderliche Herunterladen der neuesten Version von einer speziell für den *MovieVoter* angelegten Webseite und das anschließende Installieren bedeutete für einige Nutzer jedoch sicherlich zu hohe Kosten in Form von zu investierender Zeit verglichen mit dem Nutzen von Filmempfehlungen. Im Nachhinein gesehen, wäre eine reine Web-Applikation besser gewesen, da der Aufwand für die Testpersonen bei solch einer Anwendung wesentlich gering gewesen wäre. Auch dies ist eine aus den praktischen Versuchen gewonnene Erfahrung, genau wie die Tatsache, dass ein Empfehlungssystem ständige Anreize bieten muss, um die Anwender (denen es schließlich seine Existenzberechtigung verdankt) zu motivieren. Eine weitere Erfahrung ist ein gewisses Verständnis des Autors dafür, dass viele Forscher auf praktische Tests verzichten und stattdessen Laboruntersuchungen vorziehen. Der Zeitaufwand für die praktischen Versuche dieser Arbeit war enorm und machte sicherlich den größten Anteil an der Arbeit aus. Wie erwähnt wurde das *MovieVoter*-Programm ständig erweitert, diese Änderungen mussten wiederum auf der erwähnten Webseite eingepflegt und die Beschreibungen auf dieser Webseite den neuen Funktionalitäten des Programms angepasst werden. Auch das manuelle Verschicken der Empfehlungen und Vorbereiten der Bewertungen für Auswertungen nahm viel Zeit in Anspruch. Viele Testpersonen wurden auch telefonisch bei Problemen unterstützt, d.h. es fand insgesamt eine rege Kommunikation zwischen dem Autor und den Versuchsteilnehmern statt. Somit kann der Autor nachvollziehen, dass viele Forscher diesen Aufwand scheuen und praktische Versuche vermeiden. Jedoch haben gerade die praktischen Versuche gezeigt, dass auf sie nicht

verzichtet werden kann, da viele Erfahrungen und neue Erkenntnisse ohne eine reale Anwendung der implementierten Empfehlungsverfahren nicht hätten gewonnen werden können.

Zusammenfassend gesehen, hat der zweite Durchgang der praktischen Versuche eindeutig gezeigt, dass ein Hybridsystem einem rein kollaborativen Ansatz überlegen ist. Dies äußert sich sowohl in dem Anteil der interessanten an den insgesamt generierten Empfehlungen, also dem praktischen Nutzen für die Anwender, als auch in der Fähigkeit, personalisierte Empfehlungslisten zu erstellen. Durch den Hybridansatz können die Vorteile beider Verfahren, des kollaborativen und des eigenschaftsorientierten miteinander verbunden und die spezifischen Schwächen beider Vorgehensweisen aufgefangen werden.[15]
Nach Abschluss der praktischen Versuche ist somit die Zeit gekommen, ein Gesamtfazit der gemachten Erfahrungen zu ziehen.

[15] Nicht umsonst ist die Hybridisierung kollaborativer mit eigenschaftsorientierten Verfahren diejenige, die in wissenschaftlichen Publikationen am häufigsten anzutreffen ist (siehe z.B. [Basu et al. 1998], [Claypool et al. 1999] oder [Good et al. 1999]).

7 Diskussion

„Look, we've all got something to contribute to this discussion.
And I think what you should contribute from now on is silence."

Rimmer, *Red Dwarf*, 1988

Die Ziele dieser Arbeit wurden erreicht. Kapitel 2 hat mit seiner Definition und Darstellung der geschichtlichen Entwicklung von Empfehlungssystemen, sowie dem zusammenfassenden Klassifikationssystem diesen unübersichtlichen Themenbereich kompakt zusammengefasst, so wie es in keiner der zahlreichen recherchierten, wissenschaftlichen Publikationen bisher der Fall war. Damit werden Leserinnen und Leser in die Lage versetzt, sich bei Interesse einen grundlegenden Überblick zu verschaffen und in anderen Abhandlungen angesprochene spezielle Themen in den Gesamtkontext einordnen zu können.
Die Hauptthese aus Abschnitt 1.1, dass die alleinige Qualitätsmessung von Empfehlungsverfahren mittels hypothetischer Bewertungsmaße unter Laborbedingungen problematisch ist, weil sich die damit gemessene Qualität und der konkrete Nutzen für Anwender in der Realität nicht konform verhalten, wurde in Kapitel 5 anhand der Literatur und eigenen Überlegungen diskutiert und durch die praktischen Versuche dieser Arbeit bewiesen. Die ergänzende Verwendung praxisbezogenerer Qualitätsmaße bei der Evaluierung von Empfehlungssystemen ist also definitiv angebracht. Zudem zeigten die Versuche, dass die kollaborativen Verfahren in der Praxis nicht so gut abschneiden wie ihr Ruf und die vielfach publizierten exzellenten Ergebnisse in Labortests vermuten lassen würden. Das in dieser Arbeit benutzte kollaborative System liefert zwar wie erwartet viele den Nutzern gänzlich unbekannte Empfehlungen in Form von Nischenfilmen, die inhaltliche Verbindung zu den Präferenzen der Nutzerinnen in Form der von ihnen gesehenen, hoch bewerteten Filme ist jedoch zu gering, um den Geschmack der Testpersonen zu treffen, was in ungültigen Empfehlungen resultiert. Eine außerdem aus den praktischen Tests gewonnene Erkenntnis ist, dass Nutzer eher einer „inneren Bewertungsskala" zu folgen scheinen, die „fuzzy" und deren Granularität bezogen auf unterschiedliche Bewertungsklassen entsprechend niedriger ist, so dass geringe Verbesserungen der hypothetischen Vorhersagegenauigkeit von diesen Nutzern gar nicht wahrgenommen werden können. Probleme von Anwendern mit Empfehlungssystemen und deren Wünsche, um die generierten Empfehlungen besser nutzen zu können, sind zusammen mit den zuvor genannten Erkenntnissen Erfahrungen, die nur in praktischen Tests gesammelt werden können. Auch die mangelnde Divergenz der unterschiedlichen Empfehlungslisten des kollaborativen Algorithmus für an Top-Positionen vorkommende Filme könnte zwar rein theoretisch auch unter Laborbedingungen gemessen werden, würde aber in solch einer Umgebung wahrscheinlich nicht dieselbe Aufmerksamkeit erregen wie in den praktischen Versuchen. All dies zeigt noch einmal, wie wichtig praktische Versuche für Empfehlungssysteme sind und zwar ungeachtet des extremen Zeit- und Arbeitsaufwands, der auch zu den Erfahrungen dieser Arbeit zählt.
Dieser Aufwand ermöglichte es jedoch gerade, auch die letzte These aus Abschnitt 1.1 zu beweisen. Die auf den praktischen Erfahrungen basierende Verbesserung des ursprünglichen Empfehlungsalgorithmus zu einem Hybridsystem in Kapitel 6, erbrachte eine erhebliche Verbesserung des praktischen Nutzens in Form der gemessenen Interessantheit der generierten Empfehlungen für die Nutzerinnen. Die Entscheidung, das vom kollaborativen Algorithmus erzeugte Rauschen auf die Gefahr hin zu reduzieren, mögliche interessante Empfehlungen zu verlieren, hat sich als richtig erwiesen. Ein kollaboratives oder eigenschaftsbasiertes Empfehlungsverfahren alleine hätte wegen der mehrfach erwähnten Gründe ein solch gutes Ergebnis bzgl. der Interessantheit nicht erreichen können. Dabei war allerdings die Wahl des richtigen Hybridtyps, der Reihenfolge der Einzelsysteme im Kaskaden-Hybrid und des trade-offs zwischen Nutzen und Kosten für den Erfolg entscheidend. Daher zeigt diese Arbeit auch, dass nur sorgfältig geplante Hybridsysteme die Qualität von Empfehlungen verbessern können.

Bedenkt man dabei, dass Hybridsysteme - von wenigen Ausnahmen abgesehen - erst in den letzten Jahren einer genaueren Erforschung und Weiterentwicklung unterzogen wurden, so lohnt sich also auch hier ein entsprechendes Umdenken. Das ungenutzte Potenzial im Bereich der Hybridsysteme ist dabei groß. [BURKE 2002] hat in seinen Untersuchung die möglichen und sinnvollen Hybriden aus existierenden Ansätzen zur automatischen Empfehlungsgenerierung zusammengetragen. Nur die wenigsten dieser möglichen Hybridisierungen wurden bisher erforscht. An möglichen, nicht redundanten Hybridsystemen ergeben sich insgesamt 39 noch nicht erforschte Varianten, während im Vergleich dazu erst 14 Varianten untersucht und implementiert wurden.
Somit trägt diese Arbeit dazu bei, zukünftige Forschungen im Bereich der Empfehlungssysteme in die Richtung praktischer Evaluierungen von solchen Systemen zu lenken, die als Hybride die Vorteile der verschiedenen Verfahren unter möglichst großem Ausschluss der spezifischen Nachteile in sich vereinigen. So wird auch der Autor dieser Arbeit die genannten Möglichkeiten privat weiter erforschen[1] und plant jetzt schon die Implementierung webbasierter Empfehlungssysteme, deren Verbesserungen auf den Erkenntnissen dieser Arbeit beruhen.

[1] Neben weiteren Hybridansätzen z.B. eine Erfassung des Interessantheitsbegriffs mittels Regeln wie in der in Abschnitt 2.3.1 beschriebenen Methode von [MORIK 2002].

Literaturverzeichnis

[AGGARWAL et al. 1999] AGGARWAL, CHARU C., J. L. WOLF, K.-L. WU und P. S. YU (1999). *Horting hatches an egg: a new graph-theoretic approach to collaborative filtering*. In: *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 201–212, New York, NY, USA. ACM Press.

[AHA et al. 1991] AHA, DAVID W., D. KIBLER und M. K. ALBERT (1991). *Instance-Based Learning Algorithms*. Mach. Learn., 6(1):37–66.

[AVERY und ZECKHAUSER 1997] AVERY, CHRISTOPHER und R. ZECKHAUSER (1997). *Recommender systems for evaluating computer messages*. Commun. ACM, 40(3):88–89.

[BAEZA-YATES und RIBIERO-NETO 1999] BAEZA-YATES, R. und B. RIBIERO-NETO (1999). *Modern Information Retrieval*. Addison-Wesley Longman.

[BALABANOVIC und SHOHAM 1997] BALABANOVIC, MARKO und Y. SHOHAM (1997). *Fab: content-based, collaborative recommendation*. Commun. ACM, 40(3):66–72.

[BASU et al. 1998] BASU, CHUMKI, H. HIRSH und W. W. COHEN (1998). *Recommendation as Classification: Using Social and Content-Based Information in Recommendation*. In: *AAAI/IAAI*, S. 714–720. citeseer.ist.psu.edu/basu98recommendation.html.

[BLUM und MITCHELL 1998] BLUM, AVRIM und T. MITCHELL (1998). *Combining Labeled and Unlabeled Data with Co-training*. In: *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*. citeseer.ist.psu.edu/blum98combining.html.

[BREESE et al. 1998] BREESE, JOHN S., D. HECKERMAN und C. KADIE (1998). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, S. 43–52. citeseer.ist.psu.edu/breese98empirical.html.

[BURKE 2002] BURKE, ROBIN (2002). *Hybrid Recommender Systems: Survey and Experiments*. User Modeling and User-Adapted Interaction, 12(4):331–370.

[CALLAN et al. 1992] CALLAN, JAMES P., W. B. CROFT und S. M. HARDING (1992). *The INQUERY Retrieval System*. In: *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, S. 78–83. citeseer.ist.psu.edu/26307.html.

[CLAYPOOL et al. 1999] CLAYPOOL, M., A. GOKHALE, T. MIRANDA, P. MURNIKOV, D. NETES und M. SARTIN (1999). *Combining Content-Based and Collaborative Filters in an Online Newspaper*. citeseer.ist.psu.edu/claypool99combining.html.

[CLAYPOOL et al. 2001] CLAYPOOL, MARK, D. BROWN, P. LE und M. WASEDA (2001). *Inferring User Interest*. IEEE Internet Computing, 5(6):32–39.

[CLEVERDON und KEAN 1968] CLEVERDON, C. und M. KEAN (1968). *Factors Determining the Performance of Indexing Systems.* Aslib Cranfield Research Project.

[COVER und HART 1967] COVER, T.M. und P. HART (1967). *Nearest Neighbor pattern classification*. Institute of Electrical and Electronics Engineers Transactions on Information Theory, 13:21–27.

[DEMPSTER et al. 1977] DEMPSTER, A.P., N. LAIRD und D. RUBIN (1977). *Maximum likelihood from incomplete data via the EM algorithm (with discussion)*. J. Royal Stat. Soc., Series B 39, S. 1–38.

[DESTATIS 2004] DESTATIS (2004). *Ausstattung privater Haushalte mit Unterhaltungselektronik*. Technischer Bericht, Statistisches Bundesamt Deutschland. http://www.destatis.de/basis/d/evs/budtab62.php.

[DESTATIS 2005] DESTATIS (2005). *Informationstechnologie in Unternehmen und Haushalten 2004*. Technischer Bericht, Statistisches Bundesamt Deutschland. http://www.destatis.de/download/d/veroe/pb_ikt_04.pdf.

[DOMINGOS 1996] DOMINGOS, P. (1996). *Unifying instance-based and rule-based induction*. Machine Learning, 24:141–168.

[EMDE und WETTSCHERECK 1996] EMDE, WERNER und D. WETTSCHERECK (1996). *Relational Instance Based Learning*. In: SAITTA, LORENZA, Hrsg.: *Machine Learning - Proceedings 13th International Conference on Machine Learning*, S. 122 – 130. Morgan Kaufmann Publishers. citeseer.ist.psu.edu/article/emde96relational.html.

[FISCHER et al. 2002] FISCHER, SIMON, R. KLINKENBERG, I. MIERSWA und O. RITTHOFF (2002). *Yale: Yet Another Learning Environment – Tutorial*. Technischer Bericht CI-136/02, Collaborative Research Center 531, University of Dortmund, Dortmund, Germany. ISSN 1433-3325.

[FORSTINGER 1999] FORSTINGER, HARALD (1999). *Analyse gegenwärtiger Suchdienste und Konzepte für künftige Wissensauffindung*. Diplomarbeit, Technische Universität Graz. http://www.iicm.edu/thesis/hforstinger.

[FREEDMAN 1998] FREEDMAN, SAMUEL G. (1998). *Asking Software to Recommend a Good Book*. New York Times.

[GATES 1972] GATES, G.W. (1972). *The reduced nearest neighbor rule*. IEEE Transactions on Information Theory, S. 431–433.

[GOLDBERG et al. 1992] GOLDBERG, DAVID, D. NICHOLS, B. M. OKI und D. TERRY (1992). *Using collaborative filtering to weave an information tapestry*. Commun. ACM, 35(12):61–70.

[GOLDBERG et al. 2001] GOLDBERG, KEN, T. ROEDER, D. GUPTA und C. PERKINS (2001). *Eigentaste: A Constant Time Collaborative Filtering Algorithm*. Inf. Retr., 4(2):133–151.

[GOOD et al. 1999] GOOD, NATHANIEL, J. B. SCHAFER, J. A. KONSTAN, A. BORCHERS, B. SARWAR, J. HERLOCKER und J. RIEDL (1999). *Combining collaborative filtering with personal agents for better recommendations*. In: *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, S. 439–446, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[HANLEY und MCNEIL 1982] HANLEY, J.A. und B. MCNEIL (1982). *The meaning and use of the area under a receiver operating characteristic (ROC) curve.*. Radiology, 143:29–36.

[HART 1968] HART, P.E. (1968). *The condensed nearest neighbor rule*. Institute of Electrical and Electronics Engineers Transactions on Information Theory, 14:515–516.

[HEISE 2003] HEISE (2003). *Amazon wegen Kaufempfehlungen verklagt*. Online News. http://www.heise.de/newsticker/meldung/38712.

[HERLOCKER et al. 1999] HERLOCKER, JONATHAN L., J. A. KONSTAN, A. BORCHERS und J. RIEDL (1999). *An algorithmic framework for performing collaborative filtering*. In: *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, S. 230–237, New York, NY, USA. ACM Press.

[HERLOCKER et al. 2000] HERLOCKER, JONATHAN L., J. A. KONSTAN und J. RIEDL (2000). *Explaining collaborative filtering recommendations*. In: *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, S. 241–250, New York, NY, USA. ACM Press.

[HERLOCKER et al. 2004] HERLOCKER, JONATHAN L., J. A. KONSTAN, L. G. TERVEEN und J. T. RIEDL (2004). *Evaluating collaborative filtering recommender systems*. ACM Trans. Inf. Syst., 22(1):5–53.

[HILL et al. 1995] HILL, WILL, L. STEAD, M. ROSENSTEIN und G. FURNAS (1995). *Recommending and evaluating choices in a virtual community of use*. In: *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, S. 194–201, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[HOFMANN 2001] HOFMANN, THOMAS (2001). *Learning What People (Don't) Want*. In: *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, S. 214–225, London, UK. Springer-Verlag.

[IFM 2002] IFM (2002). *Network-Marketing-Konzepte: Ein Vergleich Deutschland und USA*. Technischer Bericht, Institut für Mittelstandsforschung Bonn. http://www.ifm-bonn.org/ergebnis/145.htm.

[JAIN et al. 1999] JAIN, A. K., M. N. MURTY und P. J. FLYNN (1999). *Data clustering: a review*. ACM Computing Surveys, 31(3):264–323. citeseer.ist.psu.edu/jain99data.html.

[JOACHIMS 1999] JOACHIMS, THORSTEN (1999). *Transductive Inference for Text Classification using Support Vector Machines*. In: BRATKO, IVAN und S. DZEROSKI, Hrsg.: *Proceedings of ICML-99, 16th International Conference on Machine Learning*, S. 200–209, Bled, SL. Morgan Kaufmann Publishers, San Francisco, US. citeseer.ist.psu.edu/joachims99transductive.html.

[KEENEY und RAIFFA 1976] KEENEY, R.L. und H. RAIFFA (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York.

[KOLODNER 1993] KOLODNER, J.L. (1993). *Case-Based Reasoning*. Morgan Kaufmann.

[LANG 1995] LANG, K. (1995). *Newsweeder: Learning to filter news*. In: *Proceedings of the 12th International Conference on Machine Learning*, S. 331–339.

[LITTLESTONE 1988] LITTLESTONE, N. (1988). *Learning quickly when irrelevant attributes abound: A new linearthreshold algorithm.*. Machine Learning, 2:258–318.

[LITTLESTONE 1989] LITTLESTONE, N. (1989). *Mistake bounds and logarithmic linearthreshold learning algorithms.*. Doktorarbeit, U.C. Santa Cruz.

[LITTLESTONE 1991] LITTLESTONE, N. (1991). *Redundant noisy attributes, attribute errors, and linearthreshold learning using winnow.*. In: *Proceedings of the Fourth AnnualWorkshopon Computational Learning Theory*, S. 147–156. Morgan Kaufmann.

[MAES et al. 1999] MAES, PATTIE, R. H. GUTTMAN und A. G. MOUKAS (1999). *Agents that buy and sell*. Communications of the ACM, 42(3):81–91. citeseer.ist.psu.edu/article/maes99agents.html.

[MCNEE et al. 2002] MCNEE, SEAN M., I. ALBERT, D. COSLEY, P. GOPALKRISHNAN, S. K. LAM, A. M. RASHID, J. A. KONSTAN und J. RIEDL (2002). *On the recommending of citations for research papers*. In: *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, S. 116–125, New York, NY, USA. ACM Press.

[MITCHELL 1997] MITCHELL, TOM M. (1997). *Machine Learning*. MCGraw-Hill, New York.

[MOONEY und ROY 2000] MOONEY, RAYMOND J. und L. ROY (2000). *Content-based book recommending using learning for text categorization*. In: *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, S. 195–204, New York, NY, USA. ACM Press.

[MORIK et al. 1993] MORIK, K., S. WROBEL, J.-U. KIETZ und W. EMDE (1993). *Knowledge Acquisition and Machine Learning: Theory Methods and Applications*. Academic Press, London, New York.

[MORIK 2002] MORIK, KATHARINA (2002). *Detecting Interesting Instances*. In: *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery*, S. 13–23, London, UK. Springer-Verlag.

[MORITA und SHINODA 1994] MORITA, MASAHIRO und Y. SHINODA (1994). *Information filtering based on user behavior analysis and best match text retrieval*. In: *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, S. 272–281, New York, NY, USA. Springer-Verlag New York, Inc.

[MUI et al. 2001] MUI, L., C. ANG und M. MOHTASHEMI (2001). *A Probabilistic Model for Collaborative Sanctioning*. Technischer Bericht 617, MIT LCS.

[NETWORK-WIZARDS 2005] NETWORK-WIZARDS (2005). *Internet Domain Survey*. http://www.isc.org/ds.

[NEWMAN 1997] NEWMAN, WILLIAM M. (1997). *Better or just different? On the benefits of designing interactive systems in terms of critical parameters*. In: *DIS '97: Proceedings of the conference on Designing interactive systems*, S. 239–245, New York, NY, USA. ACM Press.

[NICHOLS 1998] NICHOLS, D. (1998). *Implicit rating and filtering*. In: *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*, S. 31–36. ERCIM. citeseer.ist.psu.edu/nichols98implicit.html.

[PAZZANI 1999] PAZZANI, MICHAEL J. (1999). *A Framework for Collaborative, Content-Based and Demographic Filtering*. Artificial Intelligence Review, 13(5-6):393–408. citeseer.ist.psu.edu/pazzani99framework.html.

[PERRY 2002] PERRY, PAUL (2002). *Ressources on Collaborative Filtering*. http://www.paulperry.net/notes/cf.asp.

[PINE II 1993] PINE II, B. JOSEPH (1993). *Mass Customization*. Harvard Business School Press, Boston, Massachusetts.

[RAGHAVAN und WONG 1986] RAGHAVAN, V.V. und S. WONG (1986). *A Critical Analysis of Vector Space Model for Information Retrieval*. Journal of the American Society for Information Science, 37(5):279–287.

[REDDY et al. 2002] REDDY, P. KRISHNA, M. KITSUREGAWA, P. SREEKANTH und S. S. RAO (2002). *A Graph Based Approach to Extract a Neighborhood Customer Community for Collaborative Filtering*. In: *DNIS '02: Proceedings of the Second International Workshop on Databases in Networked Information Systems*, S. 188–200. Springer-Verlag.

[RESNICK et al. 1994] RESNICK, P., N. IACOVOU, M. SUCHAK, P. BERGSTORM und J. RIEDL (1994). *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. In: *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, S. 175–186, Chapel Hill, North Carolina. ACM. citeseer.ist.psu.edu/resnick94grouplens.html.

[RESNICK und VARIAN 1997] RESNICK, PAUL und H. R. VARIAN (1997). *Recommender systems*. Commun. ACM, 40(3):56–58.

[RITTHOFF et al. 2001] RITTHOFF, OLIVER, R. KLINKENBERG, S. FISCHER, I. MIERSWA und S. FELSKE (2001). *Yale: Yet Another Machine Learning Environment*. In: KLINKENBERG, RALF, S. RÜPING, A. FICK, N. HENZE, C. HERZOG, R. MOLITOR und O. SCHRÖDER, Hrsg.: *LLWA 01 – Tagungsband der GI-Workshop-Woche Lernen – Lehren – Wissen – Adaptivität*, Nr. 763 in *Forschungsberichte des Fachbereichs Informatik, Universität Dortmund*, S. 84–92, Dortmund, Germany. ISSN 0933-6192.

[ROBERTSON et al. 1992] ROBERTSON, STEPHEN E., S. WALKER, M. HANCOCK-BEAULIEU, A. GULL und M. LAU (1992). *Okapi at TREC*. In: *Text REtrieval Conference*, S. 21–30. citeseer.ist.psu.edu/robertson96okapi.html.

[SALTON 1971] SALTON, G. (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood, Cliffs, New Jersey.

[SALTON und BUCKLEY 1987] SALTON, GERARD und C. BUCKLEY (1987). *Term Weighting Approaches in Automatic Text Retrieval*. Technischer Bericht, Ithaca, NY, USA.

[SARWAR et al. 2000a] SARWAR, B., G. KARYPIS, J. KONSTAN und J. RIEDL (2000a). *Application of dimensionality reduction in recommender systems–a case study*. citeseer.ist.psu.edu/sarwar00application.html.

[SARWAR et al. 2000b] SARWAR, BADRUL, G. KARYPIS, J. KONSTAN und J. RIEDL (2000b). *Analysis of recommendation algorithms for e-commerce*. In: *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, S. 158–167, New York, NY, USA. ACM Press.

[SARWAR et al. 2001] SARWAR, BADRUL M., G. KARYPIS, J. A. KONSTAN und J. REIDL (2001). *Item-based collaborative filtering recommendation algorithms*. In: *World Wide Web*, S. 285–295. citeseer.ist.psu.edu/article/sarwar01itembased.html.

[SARWAR et al. 1998] SARWAR, BADRUL M., J. A. KONSTAN, A. BORCHERS, J. L. HERLOCKER, B. N. MILLER und J. RIEDL (1998). *Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System*. In: *Computer Supported Cooperative Work*, S. 345–354. citeseer.ist.psu.edu/article/sarwar98using.html.

[SCHAFER et al. 1999] SCHAFER, J. BEN, J. A. KONSTAN und J. RIEDL (1999). *Recommender systems in e-commerce*. In: *ACM Conference on Electronic Commerce*, S. 158–166. citeseer.ist.psu.edu/benschafer99recommender.html.

[SCHEIN et al. 2001] SCHEIN, A., A. POPESCUL, L. UNGAR und D. PENNOCK (2001). *Generative models for cold-start recommendations*. citeseer.ist.psu.edu/schein01generative.html.

[SCHÖLKOPF und SMOLA 2002] SCHÖLKOPF, B. und A. SMOLA (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press.

[SCOTT 2000] SCOTT, J. (2000). *Social Network Analysis: A Handbook, 2nd Edition*. Sage Publications, London.

[SHARDANAND und MAES 1995] SHARDANAND, UPENDRA und P. MAES (1995). *Social information filtering: algorithms for automating 'word of mouth'*. In: *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, S. 210–217, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[SOMMER et al. 1994] SOMMER, E., W. EMDE, J.-U. KIETZ und S. WROBEL (1994). *Mobal 4.1 user guide*. Technischer Bericht 617, GMD, St. Augustin.

[SWEARINGEN und SINHA 2001] SWEARINGEN, K. und R. SINHA (2001). *Beyond algorithms: An HCI perspective on recommender systems*. citeseer.ist.psu.edu/swearingen01beyond.html.

[TERVEEN und HILL 2001] TERVEEN, L. und W. HILL (2001). *Beyond Recommender Systems: Helping People Help Each Other.*. In: CARROLL, J. M., Hrsg.: *Human-Computer Interaction in the New Millennium*, S. 487–509. ACM Press.

[TITTEL 1999] TITTEL, ED (1999). *PC Magazine - Spyware, Viruses, and Malware. All the weapons you need to battle malicious software - and win*. Wiley & Sons.

[TSOCHANTARIDIS und HOFMANN 2002] TSOCHANTARIDIS, IOANNIS und T. HOFMANN (2002). *Support Vector Machines for Polycategorial Classification*. In: *ECML, LNAI 2430*, S. 456–467. T. Elomaa et al. (Eds.).

[TURPIN und HERSH 2001] TURPIN, ANDREW H. und W. HERSH (2001). *Why batch and user evaluations do not give the same results*. In: *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, S. 225–231, New York, NY, USA. ACM Press.

[UHLENDORF 2004] UHLENDORF, MARTIN (2004). *Die Frankfurter Buchmesse in Zahlen - facts & figures*. http://www.buchmesse.de/imperia/md/content/pdf/unternehmen/factsfigures/facts_figures_04.pdf.

[UNGAR und FOSTER 1998] UNGAR, L. und D. FOSTER (1998). *Clustering Methods For Collaborative Filtering*. In: *Proceedings of the Workshop on Recommendation Systems*. AAAI Press, Menlo Park California. citeseer.ist.psu.edu/ungar98clustering.html.

[VAPNIK 1995] VAPNIK, V.N. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag, Berlin.

[WETTSCHERECK et al. 1997] WETTSCHERECK, DIETRICH, D. W. AHA und T. MOHRI (1997). *A Review and Empirical Evaluation of Feature Weighting Methods for aClass of Lazy Learning Algorithms*. Artif. Intell. Rev., 11(1-5):273–314.

[WITTEN und FRANK 1999] WITTEN, IAN H. und E. FRANK (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

[YAO 1995] YAO, Y. Y. (1995). *Measuring retrieval effectiveness based on user preference of documents*. J. Am. Soc. Inf. Sci., 46(2):133–145.

[YU et al. 2003] YU, KAI, X. XU, M. ESTER und H.-P. KRIEGEL (2003). *Feature Weighting and Instance Selection for Collaborative Filtering: An Information-Theoretic Approach*.*. Knowl. Inf. Syst., 5(2):201–224.

[ZACHARIAS 2001] ZACHARIAS, MICHAEL M. (2001). *Direktvertrieb und Network-Marketing in Österreich*. Technischer Bericht, Fachhochschule Worms, Lehrstuhl European Business Management/Handelsmanagement. http://www.fh-worms.de/ebm-hm/professoren/zach/Studie-Kurzfassung%20DVNetwork...19.11.01.pdf.

Index