Wie funktionieren Recommendersysteme?

Didaktische Aufarbeitung der Recommender-algorithmen User-based und Item-based Collaborative Filtering

Prof. Dr. Rainer Groß
Julian Knoll (M.A.)

Fakultät Informatik
Technische Hochschule Nürnberg

Kurzzusammenfassung:

Mit dem rapiden Wachstum der Informationstechnologie zum Anfang des 21. Jahrhunderts und der damit verbundenen starken Zunahme der Datenmenge, ist Software zum Filtern und Auffinden von relevanten Informationen unverzichtbar geworden. Gerade für Unternehmen, deren Geschäftsmodell auf digitalisierten Unternehmensprozessen beruht, ist es essenziell dem jeweiligen Kunden automatisiert passende Produkte anzubieten. Recommenderalgorithmen befassen sich genau mit dieser Problemstellung und versuchen Produktempfehlungen für Kunden möglichst gut zu personalisieren.

Die bislang vorliegenden Veröffentlichungen beschreiben Recommenderalgorithmen lediglich in groben Zügen. Der vorliegende Artikel befasst sich daher mit der didaktischen Aufarbeitung zweier grundlegender Algorithmen und bietet einen transparenten und allgemein verständlichen Zugang zu diesem Thema. Die beiden Algorithmen werden anhand eines durchgängigen Beispiels von der Berechnung der Ähnlichkeitsmaße bis zur Erstellung der Empfehlungsrangliste illustriert. Darüber hinaus wird eine Methodik vorgestellt, um die Empfehlungsqualität im Rahmen von Simulationsstudien mit ROC-Kurven zu evaluieren.

ISSN 1867 5433 Sonderdruck Schriftenreihe der Technischen Hochschule Nürnberg Georg Simon Ohm Nr. 64, Juli 2016

Inhalt

1 Einführung

Um Entscheidungen im täglichen Leben zu treffen, vertrauen Menschen oft auf Empfehlungen. Diese können unter anderem von anderen Personen, Nachrichtensendungen oder aus Testberichten stammen. Analog dazu liefern Recommender- oder auch Empfehlungssysteme Benutzern von IT Anwendungen personalisierte Empfehlungen, zum Beispiel zu Büchern, Filmen oder anderen Produkten. Ziel dabei ist es (Kauf-)Entscheidungen zu unterstützen bzw. Kunden für zusätzliche Käufe zu gewinnen. Verallgemeinert gesprochen ermittelt ein Recommendersystem automatisiert und aktiv aus einer Gesamtmenge von Objekten (Items), die für einen Benutzer (User) potentiell „nützlichen" Objekte.
Recommendersysteme haben ein breites Einsatzgebiet. Beispiele sind Text-, Video-, Audio-, Bild- und Personenempfehlungen (Klahold, 2009, S. 4). Am bekanntesten sind jedoch Produkt-Empfehlungen. Diese kennt man u.a. von dem Einkaufsportal Amazon. Interessiert man sich beispielsweise für ein spezielles Buch aus dem Bereich Wirtschaftsinformatik werden dem User weitere Bücher in Abhängigkeit zum aktuell betrachteten Buch angeboten. Abbildung 1 zeigt die kundenindividuellen Empfehlungen von Amazon zum Buch „Wirtschaftsinformatik. Eine Einführung".

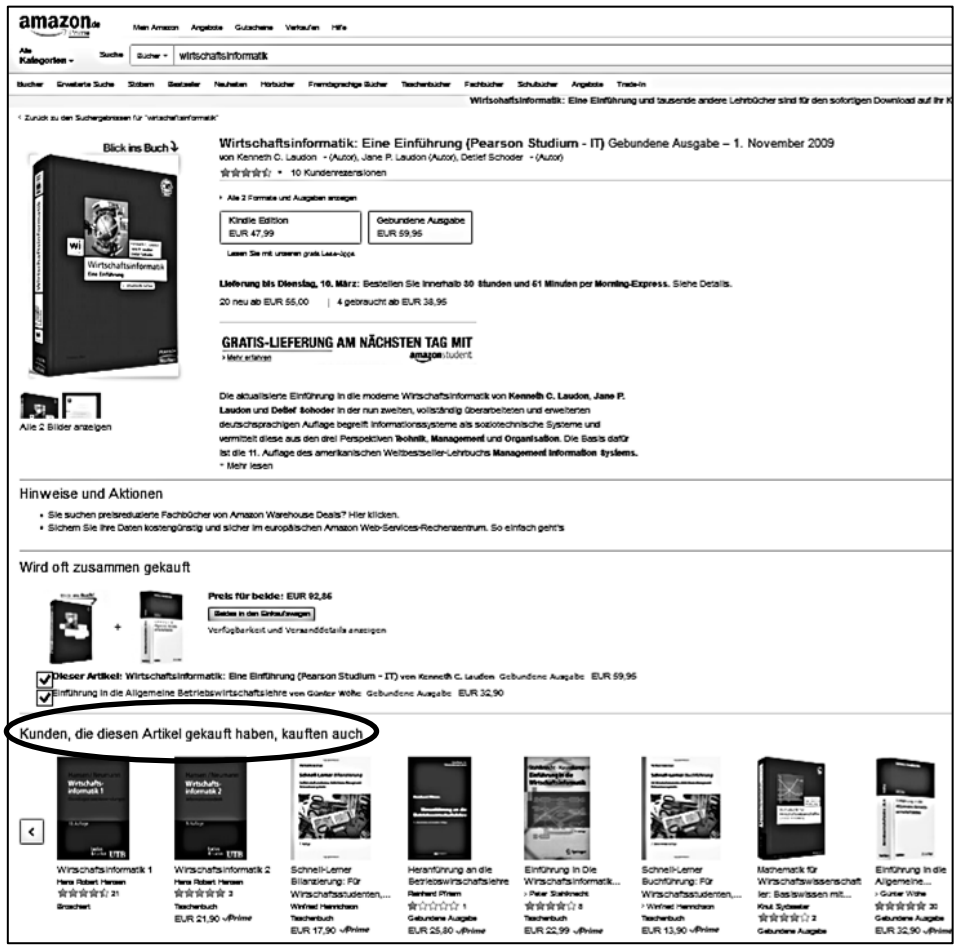

Abbildung 1: Beispiel für ein Recommender System (Amazon EU Société à responsabilité limité, 2015).

Mittlerweile hat sich die Forschung zu Recommenderalgorithmen als eigenes Forschungsgebiet etabliert. Publikationen verwenden häufig Neighborhood-based Collaborative Filtering (CF) Algorithmen, um zum Thema hinzuführen und das Forschungsgebiet zu illustrieren. Trotz alledem wurde bisher noch kein Ansatz veröffentlicht, der die Algorithmen in allgemein verständlicher Form beschreibt, so dass diese beispielsweise Studierenden in kurzer Zeit verständlich gemacht werden können.
Anhand eines durchgehenden Beispiels werden sowohl die verschiedenen Ähnlichkeitsmaße für Recommenderalgorithmen als auch die Funktionsweise der Algorithmen selbst eingehend beschrieben und durch zu-

sätzliche Grafiken verdeutlicht. Des Weiteren wird das Design einer Studie gezeigt, die universell für die Evaluierung der Qualität von Recommenderalgorithmen eingesetzt werden kann. Aufbau und Durchführung dieser Studie werden wiederum anhand eines konkreten Beispiels dargestellt. So wird ein Instrumentarium geschaffen, auf dessen Basis die Algorithmen auf einer wissenschaftlichen Ebene angewendet und bewerten werden können.
Kapitel 2 nimmt zunächst eine Einordnung von Neighborhood-based CF Algorithmen in den Kontext der Recommenderalgorithmen vor und erläutert, warum diese sich besonders gut im didaktischen Umfeld einsetzen lassen. Die Beschreibung der Ähnlichkeitsmaße und Erläuterung der Neighborhood-based CF Algorithmen erfolgt in Kapitel 3. Anschließend wird der Aufbau einer Evaluationsstudie in Kapitel 4 darstellt. Kapitel 5 gibt einen kurzen Ausblick auf die zukünftige Forschung und deren mögliche Integration in die Lehre.

2 Klassifikation von Recommenderalgorithmen

Su und Khoshgoftaar (Su & Khoshgoftaar, 2009) unterscheiden bei Recommendersystemen zwischen Content-based Filtering, Collaborative Filtering und hybriden Recommendern. Die Methoden des Collaborative Filtering lassen sich wiederum unterteilen in: Neighborhood-based (auch: Memory-based oder Heuristic-based) und Model-based CF (Desrosiers & Karypis, 2011) (Abbildung 2).

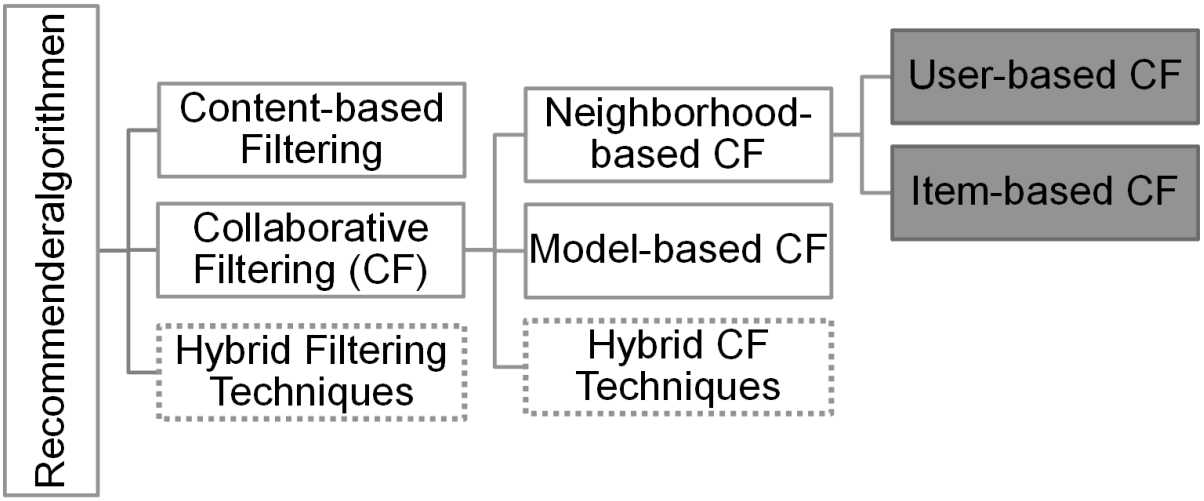

Abbildung 2: Einteilung der Recommenderalgorithmen

2.1 Einordnung von Collaborative Filtering

Im Gegensatz zu Collaborative Filtering Verfahren, die die Ähnlichkeit der Bewertungen von Usern oder Items für Prognosen und Empfehlungen verwenden, ermitteln auf „content“ also Inhalt basierende Recommendersysteme Empfehlungen, indem inhaltliche Übereinstimmungen zwischen den Eigenschaften der Items oder User für die Analyse und Prognose herangezogen werden.
Beide Ansätze haben Nachteile: CF Verfahren benötigen Informationen über die Bewertung der Items durch die User und Content-based Verfahren benötigen Informationen über die Eigenschaften der Items. Mit hybriden Techniken z. B. Content-boosted CF Algorithmen (Su & Khoshgoftaar, 2009) kombiniert man beide Ansätze und versucht dadurch die Einschränkungen zu überwinden. Weiterentwicklungen von hybriden Techniken greifen auf die Bildung komplexerer statistischer Modelle sowie auf die gesamte Palette der Verfahren des Machine Learning zurück (Shi, Larson, & Hanjalic, 2014).
Da also insbesondere diese hybriden Verfahren viel Potenzial bieten, sind diese meist Gegenstand von aktuellen Publikationen. Aufgrund der hohen Komplexität von hybriden Recommenderalgorithmen ist es jedoch didaktisch sinnvoller, die Funktionsweise von Empfehlungssystemen anhand von leichter nachvollziehbareren Algorithmen zu erklären. Hierbei bieten sich CF Verfahren an, da sie im Gegensatz zu Content-based Verfahren auch keinerlei inhaltliche Information über Produkte oder Eigenschaften von Benutzern benötigen.

2.2 Einordnung von Neighborhood-based CF

Die Verfahren im Bereich CF lassen sich grob in zwei Klassen unterteilen. Die Neighborhood-based CF-Algorithmen generieren Empfehlungen direkt auf Basis von Ähnlichkeiten zwischen Usern (User-based CF) oder Items (Item-based CF). Model-based CF-Algorithmen hingegen, schätzen auf Basis eines statistischen Modells zunächst Parameter, die sie dann in Kombination mit dem Modell für die Erzeugung von Empfehlungen verwenden (Shi, Larson, & Hanjalic, 2014).
Da der Zwischenschritt der Parameterschätzung und die nicht triviale Auswahl eines geeigneten Modells bei den Neighborhood-based Verfahren entfallen, erscheinen diese eher intuitiv und leichter verständlich als die übrigen Verfahren unter den Recommenderalgorithmen. Des Weiteren besitzen diese Verfahren auch in der Praxis eine hohe Bedeutung und dienen beispielsweise als Grundlage für das Empfehlungssystem von Amazon (Linden, Smith, & York, 2003). Unter Berücksichtigung der genannten Aspekte erscheint es also sinnvoll, eine Hinführung zum Thema Recommendersysteme anhand der Neighborhood-based CF-Algorithmen vorzunehmen.

3 Neighborhood-based CF

Die grundlegende Idee bzw. Annahme von CF-Algorithmen ist:

„Wenn die Benutzer X und Y Produkte ähnlich bewerten oder ähnliches Verhalten diesen gegenüber zeigen, dann werden sie sich gegenüber anderen Produkten auch ähnlich verhalten."

Neigborhood-based CF Ansätze verfolgen diese grundsätzliche CF-Idee und versuchen aus der Ähnlichkeit der Wertschätzungen von Benutzer X und Y gegenüber den Produkten eine Prognose für die Wertschätzung von Benutzer X oder Y für andere Produkte abzuleiten. (Desrosiers & Karypis, 2011) unterscheiden dabei zwischen drei Typen von Informationen die Benutzer über Produkte liefern: skalare, binäre und unäre Informationen.

- Skalare Informationen erfolgen auf Ratingskalen entweder numerisch codiert (z. B. 1 bis 5) oder verbal (z. B. sehr gut, gut, neutral, schlecht, sehr schlecht). Grundsätzlich liefern diese Skalen ordinales Niveau, das aber zum Teil als metrisch angesehen und verwendet wird.
- Binäre Informationen haben nur zwei mögliche diametrale Ausprägungen (z. B. Zustimmung/Ablehnung (like/dislike), interessiert/nicht interessiert).
- Unäre Informationen halten die Interaktionen zwischen Benutzern und Produkten fest ohne expliziten Hinweis über die tatsächliche Wertschätzung des Benutzers in Bezug auf das Produkt. Beispiele sind: Kauf, Online-Klickverhalten oder Mausbewegungen. Obwohl die tatsächliche Wertschätzung des Benutzers bei unären Informationen fehlt, liefern diesen dennoch interessante Hinweise über die Vorlieben der Benutzer.

Die Art des Feedbacks durch den Benutzer kann also variieren. Beispielsweise können die Benutzer in einem Einkaufsportal nachdem sie Bücher gekauft und gelesen haben, diese auf einer Ratingskala bewerten und so ihre Wertschätzung gegenüber dem Buch explizit ausdrücken. Ebenso kann Feedback aus der Kaufhistorie oder dem Zugriffsverhalten der Benutzer gewonnen werden. Zum Beispiel kann die Zeit gemessen werden, die ein Benutzer auf einer Webseite für ein spezielles Buch verbringt und so als Indikator für Präferenzen des Benutzers verwendet werden.

3.1 Ähnlichkeitsmaße

Es ist zunächst zu klären, wann zwei Benutzer Produkte ähnlich bewerten und wie man diesen Sachverhalt messen kann. Dies kann anhand einer sogenannten User-Item-Matrix erfolgen. Tabelle 1 zeigt eine User-Item-Matrix, die die Bewertungen von 5 Usern $(u_i, i = 1, \dots, 5)$ für 6 Bücher (=Items), $(v_j, j = 1, \dots, 6)$ enthält. Es liegen explizite Bewertungen $w_{i,j}$ der User für die Items auf einer 5-stufigen Ratingskala vor, wobei 1 sehr schlecht und 5 sehr gut bedeutet. Fehlende Bewertungen sind mit NA („not available", also nicht verfügbar) gekennzeichnet.
Aus Tabelle 1 kann man z. B. entnehmen, dass der User 1 das Buch 5 mit 1 $(w_{1,5} = 1)$ also sehr schlecht bewertet hat. Dagegen bewertet User 2 Buch 5 mit sehr gut $(w_{2,5} = 5)$.

	v_1	v_2	v_3	v_4	v_5	v_6
u_1	NA	3	2	NA	1	2
u_2	1	4	2	5	NA	4
u_3	2	NA	4	3	5	NA
u_4	1	1	NA	1	4	4
u_5	2	3	4	1	NA	2

Tabelle 1: Beispiel User-Item-Matrix

3.1.1 Korrelationskoeffizient nach Bravais Pearson

Wie kann man nun konkret Ähnlichkeit messen? Dazu gibt es viele Möglichkeiten. Eine einfache, aber sehr bekannte, Methode verwendet den Korrelationskoeffizienten nach Bravais-Pearson (Fahrmeier, Künstler, Pigeot, & Tutz, 1999, S. 136ff.). Streng genommen handelt es sich bei Bewertungen um ordinal skalierte Merkmale, für die die Berechnung des Korrelationskoeffizienten nach Bravais-Pearson nicht sinnvoll ist, da dieser ein metrisches Skalenniveau verlangt. Allerdings kann man bei einer 5-stufigen Ratingskala durchaus von einer guten Annäherung an ein metrisches Skalenniveau ausgehen, so dass vereinfachend mit dem Korrelationskoeffizienten gearbeitet werden kann.
Der Korrelationskoeffizienten nach Bravais-Pearson wird entweder für alle Tupel von Zeilenvektoren oder für alle Tupel von Spaltenvektoren der User-Item-Matrix berechnet. Ersteres kommt bei User-based CF zur Anwendung, da hier die Ähnlichkeiten der User zueinander bestimmt werden. Berechnet man die Korrelationskoeffizienten über die Tupel der Spaltenvektoren, dann betrachtet man die Ähnlichkeiten zwischen den Items. Diese Ähnlichkeitsbestimmung ist Basis für Item-based CF.
Die allgemeine Formel für die Berechnung des Zusammenhangs zwischen zwei Merkmalen X und Y mittels des Korrelationskoeffizienten nach Bravais-Pearson $r(bp)_{x,y}$ lautet (Fahrmeier, Künstler, Pigeot, & Tutz, 1999, S. 136ff.):

$$r(bp)_{x,y} = \frac{cov_{x,y}}{s_x \cdot s_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \\ = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sqrt{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}\sqrt{\sum_{i=1}^{n} y_i^2 - n\bar{y}^2}}$$

mit $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ und $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

Dabei gilt für $r(bp)_{x,y}$ der Wertebereich: $-1 \leq r(bp)_{x,y} \leq +1$. Für $r(bp)_{x,y} > 0$ liegt ein positiver, für $r(bp)_{x,y} < 0$ ein negativer und für $r(bp)_{x,y} = 0$ kein linearer Zusammenhang zwischen den Merkmalen X und Y vor.
Obige Formel kann man nun zur paarweisen Berechnung des Zusammenhangs zwischen den Bewertungen der User oder der Items heranziehen. Als Beispiel dient die Berechnung der Stärke des Zusammenhangs der Bewertungen zwischen Buch 2 und Buch 5. Dazu verwendet man die zweite und die fünfte Spalte der User-Item-Matrix (Tabelle 1).
Die Formel zur Berrechnung des Korrelationskoeffizienten nach Bravais-Pearson sieht auf die Beispieldaten angepasst wie folgt aus:

$$r(bp)_{v_2,v_5} = \frac{\sum_{i=1}^{n}\left(w_{i,2} - \bar{w}_{v_2}\right)\left(w_{i,5} - \bar{w}_{v_5}\right)}{\sqrt{\sum_{i=1}^{n}\left(w_{i,2} - \bar{w}_{v_2}\right)^2}\sqrt{\sum_{i=1}^{n}\left(w_{i,5} - \bar{w}_{v_5}\right)^2}}$$

mit $\bar{w}_{v_2} = 1/n\sum_{i=1}^{n} w_{i,2}$ und $\bar{w}_{v_5} = 1/n\sum_{i=1}^{n} w_{i,5}$

Berechnungsbeispiel für $r(bp)_{v_2,v_5}$:

i	$i=1$	2	3	4	5		
	u_1	u_2	u_3	u_4	u_5	$\sum$	$\overline{w}_{v_j}$
v_2	3	(4)	NA	1	(3)	$3+1=4$	$4/2=2$
v_5	1	NA	(5)	4	NA	$1+4=5$	$5/2=2{,}5$
$w_{i,2}-\overline{w}_{v_2}$	1	(2)	NA	-1	(1)		
$w_{i,5}-\overline{w}_{v_5}$	-1,5	NA	(2,5)	1,5	NA		
$(w_{i,2}-\overline{w}_{v_2})\cdot(w_{i,5}-\overline{w}_{v_5})$	-1,5	NA	NA	-1,5	NA	-3	
$(w_{i,2}-\overline{w}_{v_2})^2$	1	NA	NA	1	NA	2	
$(w_{i,5}-\overline{w}_{v_5})^2$	2,25	NA	NA	2,25	NA	4,5	

Tabelle 2: Hilfstabelle zur Ermittlung des Korrelationskoeffizienten nach Bravais-Pearson

Damit ergibt sich für $r(bp)_{v_2,v_5}=\frac{\sum_{i=1}^{n}(w_{i,2}-\overline{w}_{v_2})(w_{i,5}-\overline{w}_{v_5})}{\sqrt{\sum_{i=1}^{n}(w_{i,2}-\overline{w}_{v_2})^2}\sqrt{\sum_{i=1}^{n}(w_{i,5}-\overline{w}_{v_5})^2}}=\frac{-3}{\sqrt{2}\sqrt{4{,}5}}=-1.$

Somit liegt ein perfekter negativer linearer Zusammenhang zwischen v_2 und v_5 vor. Bei der Berechnung ist zu beachten, dass nur die Bewertungen in die Berechnung des Korrelationskoeffizienten einfließen, die auch paarweise vollständig vorliegen. Im obigen Beispiel sind dies nur die Paare $i=1$ und $i=4$. Berechnet man die Korrelationskoeffizienten nach obigen Formeln für alle $r(bp)_{v_k,v_l}$ erhält man mit den Daten aus Tabelle 1 folgende Korrelationsmatrix der Items:

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	1,00	0,19	1,00	-0,30	1,00	-1,00
v_2	0,19	1,00	-0,50	0,76	-1,00	-0,23
v_3	1,00	-0,50	1,00	-0,87	1,00	-0,50
v_4	-0,30	0,76	-0,87	1,00	1,00	0,50
v_5	1,00	-1,00	1,00	1,00	1,00	1,00
v_6	-1,00	-0,23	-0,50	0,50	1,00	1,00

Tabelle 3: Korrelationsmatrix der Items

Wie man leicht erkennt ist die Matrix symmetrisch und die Hauptdiagonale ist durchgehend mit dem Wert 1 besetzt. Dadurch ist es nur notwendig die Werte oberhalb der Hauptdiagonalen zu berechnen. Analog geht man bei der Berechnung der Stärke des Zusammenhangs zwischen den Bewertungen der User vor. Anstelle von jeweils zwei Spaltenvektoren werden nun jeweils zwei Zeilenvektoren der User-Item-Matrix verwendet. Führt man die Berechnung mit den Daten aus Tabelle 1 für alle Paare durch, erhält man folgende Korrelationsmatrix der User:

	u_1	u_2	u_3	u_4	u_5
u_1	1,00	0,50	-1,00	-0,87	0,00
u_2	0,50	1,00	0,24	0,19	-0,45
u_3	-1,00	0,24	1,00	0,94	0,65
u_4	-0,87	0,19	0,94	1,00	0,00
u_5	0,00	-0,45	0,65	0,00	1,00

Tabelle 4: Korrelationsmatrix der User

3.1.2 Rang-Korrelationskoeffizient nach Spearman

Bei der Verwendung des Korrelationskoeffizienten nach Bravais-Pearson wurde unterstellt, dass eine 5-stufige ordinale Ratingskala eine gute Annäherung an ein metrisches Skalenniveau und der Korrelationskoeffizienten nach Bravais-Pearson damit auch geeignet für die Berechnung des Zusammenhangs zwischen Usern und Items ist. Streng genommen liegt im verwendeten Beispiel jedoch ordinales Skalenniveau vor, so dass es sinnvoller ist ein Zusammenhangsmaß zu verwenden, welches dieses Skalenniveau berücksichtigt. Der Rang-Korrelationskoeffizient nach Spearman misst den monotonen Zusammenhang zwischen zwei ordinal skalierten Merkmalen auf Basis des paarweisen Vergleichs der Rangplätze der Bewertungen und ist daher für den gegebenen Sachverhalt besser geeignet als der Korrelationskoeffizient nach Bravais-Pearson.
Bei der Berechnung des Rang-Korrelationskoeffizienten wird folgendermaßen vorgegangen. Zunächst ermittelt man anhand der Merkmalsausprägungen Rangplätze. Dabei kann es durchaus vorkommen, dass die ursprünglichen Merkmalsausprägungen identische Werte aufweisen. Diese identischen Werte nennt man Bindungen oder Ties. Um die Bindungen aufzulösen behilft man sich mit Durchschnittsrängen, d.h. jedem der identischen Werte wird als Rang das arithmetische Mittel der entsprechenden Ränge zugewiesen (Fahrmeier, Künstler, Pigeot, & Tutz, 1999, S. 142). Betrachten wir als Beispiel wieder die beiden Spaltenvektoren $\overrightarrow{v_2} = \{3{,}4, NA, 1{,}3\}$ und $\overrightarrow{v_5} = \{1, NA, 5{,}4, NA\}$ der Tabelle 1. Die Bewertungen werden zunächst in Ränge überführt. Dabei erkennt man, dass bei $\overrightarrow{v_2}$ die Bewertung „3“ zweimal vorkommt, so dass jedem der Messwerte der Durchschnittsrang $rg = (2 + 3)/2 = 2{,}5$ zugewiesen wird.

i	1	2	3	4	5
	u_1	u_2	u_3	u_4	u_5
v_2	3	4	NA	1	3
v_5	1	NA	5	4	NA
$rg_{\mathrm{i},2}$	2,5	1	NA	4	2,5
$rg_{\mathrm{i},5}$	3	NA	1	2	NA

Tabelle 5: Hilfstabelle zur Bestimmung der Ränge

Da Rangplätze metrisch skaliert sind, kann der Korrelationskoeffizient nach Spearman $r(sp)$ analog zum Korrelationskoeffizienten nach Pearson berechnet werden. Der Unterschied besteht lediglich darin, dass die ursprünglichen Bewertungen durch Rangplätze ersetzt werden. Für den Zusammenhang zwischen zwei Merkmalen X und Y ergibt sich damit folgende Formel:

$$r(sp)_{x,y} = \frac{\sum_{i=1}^{n}(rg(x_i) - \overline{rg}_x)\left(rg(y_i) - \overline{rg}_y\right)}{\sqrt{\sum_{i=1}^{n}(rg(x_i) - \overline{rg}_x)^2}\sqrt{\sum_{i=1}^{n}\left(rg(y_i) - \overline{rg}_y\right)^2}}$$

mit $\overline{rg}_x = 1/n \sum_{i=1}^{n} rg(x_i)$ und $\overline{rg}_y = 1/n \sum_{i=1}^{n} rg(y_i)$.
Dabei gilt für $r(sp)_{x,y}$ der Wertebereich: $-1 \leq r(sp)_{x,y} \leq +1$. Für $r(sp)_{x,y} > 0$ liegt ein positiver, für $r(sp)_{x,y} < 0$ ein negativer und für $r(sp)_{x,y} = 0$ kein montoner Zusammenhang zwischen den Merkmalen X und Y *vor.*
Die Formel sieht auf die Tabelle 1 und das obige Beispiel angepasst wie folgt aus:

$$r(sp)_{v_2,v_5} = \frac{\sum_{i=1}^{n}\left(rg_{\mathrm{i},2} - \overline{rg}_{v_2}\right)\left(rg_{\mathrm{i},5} - \overline{rg}_{v_5}\right)}{\sqrt{\sum_{i=1}^{n}\left(rg_{\mathrm{i},2} - \overline{rg}_{v_2}\right)^2}\sqrt{\sum_{i=1}^{n}\left(rg_{\mathrm{i},5} - \overline{rg}_{v_5}\right)^2}}$$

Berechnungsbeispiel für $r(sp)_{v_2,v_5}$:

i	1	2	3	4	5		
	u_1	u_2	u_3	u_4	u_5	$\sum$	$\overline{rg}_{v_j}$
$rg_{\mathrm{i},2}$	2,5	(1)	NA	4	(2,5)	2,5+4 = 6,5	6,5/2 = 3,25
$rg_{\mathrm{i},5}$	3	NA	(1)	2	NA	3+2 = 5	5/2 = 2,5
$rg_{\mathrm{i},2} - \overline{rg}_{v_2}$	-0,75	(-2,25)	NA	0,75	(-0,75)		
$rg_{\mathrm{i},5} - \overline{rg}_{v_5}$	0,5	NA	(-1,5)	-0,5	NA		
$\left(rg_{\mathrm{i},2} - \overline{rg}_{v_2}\right) \cdot \left(rg_{\mathrm{i},5} - \overline{rg}_{v_5}\right)$	-0,375	NA	NA	-0,375	NA	-0,75	
$\left(rg_{\mathrm{i},2} - \overline{rg}_{v_2}\right)^2$	0,5625	NA	NA	0,5625	NA	1,125	
$\left(rg_{\mathrm{i},5} - \overline{rg}_{v_5}\right)^2$	0,25	NA	NA	0,25	NA	0,5	

Tabelle 6: Hilfstabelle zur Ermittlung des Rangkorrelationskoeffizienten nach Spearman

Damit ergibt sich in unserem Beispiel für den Zusammenhang der Bewertungen zwischen Buch 2 und Buch 5

$$r(sp)_{v_2,v_5} = \frac{\sum_{i=1}^{n}\left(rg_{\mathrm{i},2} - \overline{rg}_{v_2}\right)\left(rg_{\mathrm{i},5} - \overline{rg}_{v_5}\right)}{\sqrt{\sum_{i=1}^{n}\left(rg_{\mathrm{i},2} - \overline{rg}_{v_2}\right)^2}\sqrt{\sum_{i=1}^{n}\left(rg_{\mathrm{i},5} - \overline{rg}_{v_5}\right)^2}} = \frac{-0{,}75}{\sqrt{1{,}125}\sqrt{0{,}5}} = -1$$

Somit liegt auch ein perfekter negativer Zusammenhang zwischen v_2 und v_5 vor. Bei der Berechnung ist wieder zu beachten, dass nur die Bewertungen in die Berechnung des Korrelationskoeffizienten einfließen, die auch paarweise vollständig vorliegen. Im obigen Beispiel sind dies nur die Paare $i = 1$ und $i = 4$.
Berechnet man die Korrelationskoeffizienten nach obigen Formeln für alle $r(sp)_{v_k,v_l}$ erhält man mit den Daten aus Tabelle 1 folgende Korrelationsmatrix der Items:

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	1,00	0,00	1,00	-0,24	1,00	-1,00
v_2	0,00	1,00	-0,50	0,87	-1,00	0,00
v_3	1,00	-0,50	1,00	-0,87	1,00	-0,50
v_4	-0,24	0,87	-0,87	1,00	1,00	0,50
v_5	1,00	-1,00	1,00	1,00	1,00	1,00
v_6	-1,00	0,00	-0,50	0,50	1,00	1,00

Tabelle 7: Korrelationsmatrix der Items (auf Basis Spearman)

Zur Vollständigkeit ist nachfolgend die mit dem Korrelationskoeffizienten nach Spearman erzeugte Matrix der User auf Basis der Daten aus Tabelle 1 angegeben.

	u_1	u_2	u_3	u_4	u_5
u_1	1,00	0,50	-1,00	-0,87	0,00
u_2	0,50	1,00	0,50	0,00	-0,50
u_3	-1,00	0,50	1,00	0,87	0,50
u_4	-0,87	0,00	0,87	1,00	0,00
u_5	0,00	-0,50	0,50	0,00	1,00

Tabelle 8: Korrelationsmatrix der User (auf Basis Spearman)

Liegen keine Bindungen vor, dann kann man den Koeffizienten nach Spearman mit folgender Formel vereinfacht bestimmen (Fahrmeier, Künstler, Pigeot, & Tutz, 1999, S. 145):

$$r(sp)_{x,y} = 1 - \frac{6 \cdot \sum_{i=1}^{n} d_i^2}{n \cdot (n^2 - 1)}$$

Bei den d_i^2 handelt es sich um die quadrierten Rangplatzdifferenzen zwischen den Bewertungspaaren.

3.1.3 Kosinus-Ähnlichkeit

Dieses Maß interpretiert den Winkel, den jeweils zwei Zeilen- oder zwei Spaltenvektoren der User-Item-Matrix aufspannen als Ähnlichkeitsmaß nach folgender Formel,

$$\cos(\vec{A}, \vec{B}) = \frac{\vec{A} \circ \vec{B}}{\|\vec{A}\| \cdot \|\vec{B}\|} = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2}}$$

mit $\vec{A} = \{a_1, a_2, \dots, a_n\}$ und $\vec{B} = \{b_1, b_2, \dots, b_n\}$. Im folgenden Berechnungsbeispiel nehmen wir für $\vec{U}$ und $\vec{V}$ die wieder die beiden Spaltenvektoren $\overrightarrow{v_2} = \{3{,}4, NA, 1{,}3\}$ und $\overrightarrow{v_5} = \{1, NA, 5{,}4, NA\}$ der User-Item-Matrix (Tabelle 1). Damit ergibt sich

$$0 \leq cos(\overrightarrow{v_2}, \overrightarrow{v_5}) = \frac{3 \cdot 1 + 1 \cdot 4}{\sqrt{3^2 + 1^2} \cdot \sqrt{1^2 + 4^2}} = \frac{7}{\sqrt{10} \cdot \sqrt{17}} = 0{,}5369 \leq 1.$$

Das Prinzip der Kosinus-Ähnlichkeit kann man am obigen Beispiel auch graphisch sehr anschaulich erklären. Reduziert man die beiden Vektoren $\overrightarrow{v_2}$ und $\overrightarrow{v_5}$ um die nicht vollständigen Bewertungspaare, dann erhält man die Vektoren $\overrightarrow{v_2^*} = \{3{,}1\}$ und $\overrightarrow{v_5^*} = \{1{,}4\}$. Der Kosinus des Winkels $\alpha = 57.53°$ zwischen den beiden Vektoren entspricht dem berechneten Wert der Kosinus-Ähnlichkeit von 0.5369 im Bogenmaß (Abbildung 3).

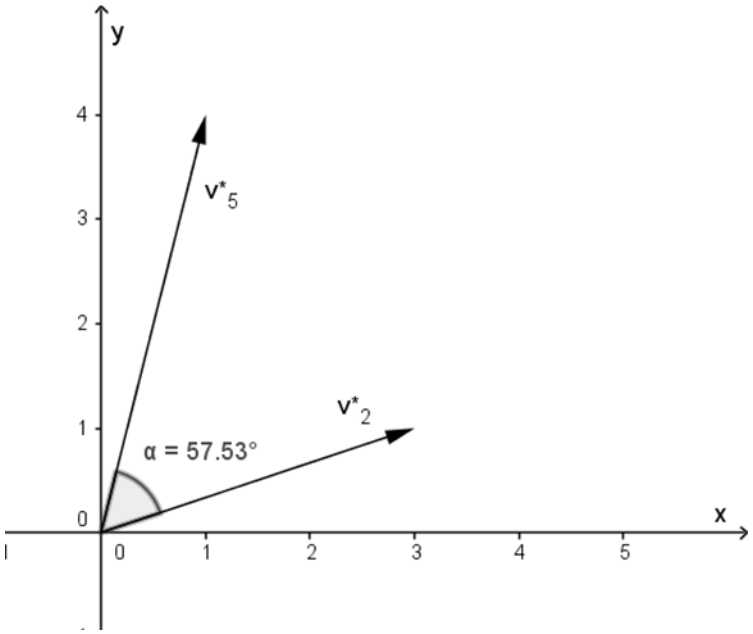

Abbildung 3: Graphische Veranschaulichung der Kosinus-Ähnlichkeit

Die Kosinus-Ähnlichkeit nimmt den Wert 1 an, wenn beide Vektoren genau richtungsgleich sind, wie z. B. die beiden Vektoren $\vec{A} = \{2{,}1\}$ und $\vec{B} = \{4{,}2\}$. Der Winkel zwischen $\vec{A}$ und $\vec{B}$ beträgt in diesem Fall $0°$, der wiederum dem Wert 1 im Bogenmaß entspricht. In diesem Fall liegt eine perfekte Ähnlichkeit vor.
Die Kosinus-Ähnlichkeit nimmt den Wert 0 an, wenn beide Vektoren aufeinander senkrecht stehen und einen $90°$-Winkel einschließen (Orthogonalität). Ein Beispiel sind die beiden Vektoren $\vec{C} = \{4{,}0\}$ und $\vec{D} = \{0{,}3\}$. In diesem Fall sind die Bewertungen perfekt unähnlich. Abbildung 4 veranschaulicht die beiden Beispiele.

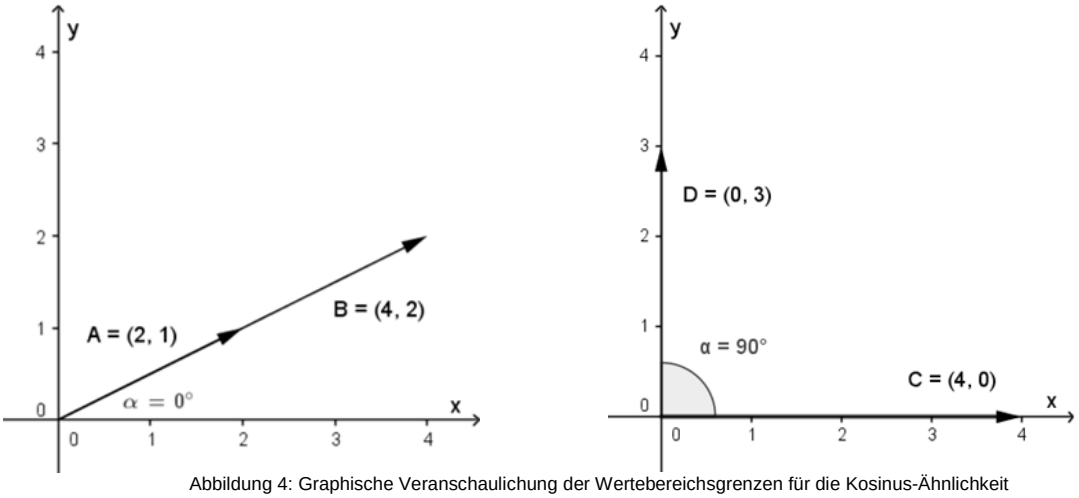

Abbildung 4: Graphische Veranschaulichung der Wertebereichsgrenzen für die Kosinus-Ähnlichkeit

In der Realität haben unterschiedliche User auch unterschiedliche Skalen vor Augen, wenn sie eine Bewertung auf einer Ratingskala durchführen. Diesen Effekt kann die Kosinus-Ähnlichkeit nicht berücksichtigen. Die modifizierte Kosinus-Ähnlichkeit $cos_{mod}(\vec{A}, \vec{B})$ berücksichtigt dies, indem für jede Bewertung der jeweilige Mittelwert

der Bewertungen abgezogen wird. Im Endeffekt erhält man dadurch wieder den Korrelationskoeffizienten nach Pearson.

$$cos_{mod}(\vec{A},\vec{B}) = \frac{\sum_{i=1}^{n}(a_i-\bar{a})\cdot(b_i-\bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i-\bar{a})^2}\cdot\sqrt{\sum_{i=1}^{n}(b_i-\bar{b})^2}}$$

Für das obige Beispiel lag der Korrelationskoeffizienten nach Pearson bei

$$cos_{mod}(\overrightarrow{v_2},\overrightarrow{v_5}) = r(bp)_{v_2,v_5} = \frac{\sum_{i=1}^{n}(w_{i,2}-\bar{w}_{v_2})(w_{i,5}-\bar{w}_{v_5})}{\sqrt{\sum_{i=1}^{n}(w_{i,2}-\bar{w}_{v_2})^2}\sqrt{\sum_{i=1}^{n}(w_{i,5}-\bar{w}_{v_5})^2}} = \frac{-3}{\sqrt{2}\sqrt{4{,}5}} = -1.$$

3.1.4 Jaccard-Koeffizient

Für binäre Daten kann man den Jaccard-Koeffizienten verwenden. Für die weiteren Überlegungen muss die User-Item-Matrix (Tabelle 1) folglich binär codiert werden. Dazu werden jetzt nur noch das Merkmal „Buch bewertet“ mit den Ausprägungen 0 für „Buch nicht bewertet“ und 1 für „Buch bewertet“ betrachtet. Demnach wird Tabelle 1 wie folgt umcodiert: Alle Tabellenelemente mit dem Wert NA werden auf 0 und alle anderen Elemente auf 1 gesetzt. Damit ergibt sich folgende binäre Matrix:

	v_1	v_2	v_3	v_4	v_5	v_6
u_1	0	1	1	0	1	1
u_2	1	1	1	1	0	1
u_3	1	0	1	1	1	0
u_4	1	1	0	1	1	1
u_5	1	1	1	1	0	1

Tabelle 9: Binäre User-Item-Matrix

Die Jaccard Metrik bildet den Quotienten aus der Schnitt- und der Vereinigungsmenge von zwei Zeilen- bzw. zwei Spaltenvektoren, wobei man unter Schnittmenge die Anzahl der Wertepaare versteht, die beide den Wert 1 haben. Unter Vereinigungsmenge versteht man die Anzahl der Wertepaare bei denen mindestens ein Element den Wert 1 hat:

$$Jaccard(\vec{A};\vec{B}) = \frac{|\vec{A}\cap\vec{B}|}{|\vec{A}\cup\vec{B}|}$$

Verwenden wir für $\vec{A}$ und $\vec{B}$ wieder die beiden $\overrightarrow{v_2} = \{1,1,0,1,1\}$ und $\overrightarrow{v_5} = \{1,0,1,1,0\}$ der binären User-Item-Matrix (Tabelle 9), so ergibt sich für den Jaccard-Koeffizienten

$$0 \le Jaccard(\overrightarrow{v_2},\overrightarrow{v_5}) = \frac{1+0+0+1+0}{1+1+1+1+1} = \frac{2}{5} = 0{,}4 \le 1$$

3.2 Prognosen

Auf Basis der ermittelten Ähnlichkeitsmatrizen lassen sich nun Prognosen und Empfehlungen für einen User, den wir im Folgenden als aktiven User (u_a) bezeichnen, erstellen. Prognosen haben das Ziel für ein nicht vorhandenes Element der User-Item-Matrix eine Schätzung zu ermitteln. Dafür verwendet man entweder die Ähnlichkeiten zwischen den Items (vgl. Kapitel 3.2.1) oder die Ähnlichkeiten zwischen den User (vgl. Kapitel 3.2.2).

Für die weiteren Berechnungen wird der Vektor des Users u_a mit folgenden Bewertungen eingeführt (Tabelle 10).

	v_1	v_2	v_3	v_4	v_5	v_6
u_a	NA	1	5	NA	4	NA

Tabelle 10: Bewertungen des aktiven Users für die Items

3.2.1 Prognosen mit Item-based CF

Item-based CF verwendet die Korrelationen der Bewertungen zwischen den Items. Eine Prognose für die Bewertung des User u_a für Item v_b nach Item-based CF kann man folgende Formel verwenden (Su & Khoshgoftaar, 2009):

$$P(bp)^{IB}_{u_a,v_b} = \frac{\sum_{j=1,b\neq j}^{m} w_{u_a,j} \cdot r(bp)_{v_b,v_j}}{\sum_{j=1,b\neq j}^{m} \left|r(bp)_{v_b,v_j}\right|}$$

Für die Berechnung benötigt man den Vektor der Bewertungen des Users u_a (Tabelle 10) und die Korrelationsmatrix der Items (Tabelle 3). Dabei sind $w_{u_a,j}$ alle Bewertungen des aktiven Users u_a mit Ausnahme des Items v_b dessen Wert prognostiziert werden soll. Die $w_{u_a,j}$ werden mit den Korrelationskoeffizienten nach Bravais-Pearson $r(bp)_{v_b,v_j}$ gewichtet und durch die Summe der Beträge von $r(bp)_{v_b,v_j}$ geteilt. $r(bp)_{v_b,v_j}$ steht dabei für die Korrelationen zwischen dem Item v_b das prognostiziert werden soll und allen anderen Items v_j. Als Beispiel dient die Berechnung der Prognose für das Item v_1 des aktiven Users u_a.

$$P(bp)^{IB}_{u_a,v_1} = \frac{w_{u_a,2} \cdot r(bp)_{1,2} + w_{u_a,3} \cdot r(bp)_{1,3} + w_{u_a,4} \cdot r(bp)_{1,4} + w_{u_a,5} \cdot r(bp)_{1,5} + w_{u_a,6} \cdot r(bp)_{1,6}}{\left|r(bp)_{1,2}\right| + \left|r(bp)_{1,3}\right| + \left|r(bp)_{1,4}\right| + \left|r(bp)_{1,5}\right| + \left|r(bp)_{1,6}\right|}$$
$$= \frac{1 \cdot 0{,}19 + 5 \cdot 1 + [NA \cdot (-0{,}3)] + 4 \cdot 1 + [NA \cdot (-1)]}{|0{,}19| + |1| + [|-0{,}30|] + |1| + [(-1)]} = \frac{0{,}19 + 5 + 4}{0{,}19 + 1 + 1} = \frac{9{,}19}{2{,}19} = 4{,}1963$$

Anhand des Beispiels kann man gut die Idee der Prognoseformel erkennen: Die jeweiligen Bewertungen des aktiven Users werden mit der Korrelation zwischen dem zu prognostizierendem Item und dem Item der jeweils betrachteten Bewertung gewichtet und anschließend aufsummiert. D. h., je ähnlicher sich die die Items sind, desto stärker wird die zugehörige Bewertung gewichtet, so dass bei perfekter positiver Korrelation die zugehörige Bewertung mit dem Faktor 1 in die Prognose einfließt.

Für Prognosen mit Item-based CF kann auch der Korrelationskoeffizient nach Spearman verwendet werden. Man erhält hier allerdings keine Prognose für die Bewertung eines Items durch den User sondern eine Prognose für den Rangplatz der Bewertung. Obige Formel ändert sich dann wie folgt

$$P(sp)^{IB}_{u_a,v_b} = \frac{\sum_{j=1,b\neq j}^{m} rg(w_{u_a,j}) \cdot r(sp)_{v_b,v_j}}{\sum_{j=1,b\neq j}^{m} \left|r(sp)_{v_b,v_j}\right|}$$

Mit den Beispieldaten ergibt sich folgende Prognose für den Rang von Items v_1 des aktiven Users u_a

$$P(sp)^{IB}_{u_a,v_1}$$
$$= \frac{rg(w_{u_a,2}) \cdot r(sp)_{1,2} + rg(w_{u_a,3}) \cdot r(sp)_{1,3} + rg(w_{u_a,4}) \cdot r(sp)_{1,4} + rg(w_{u_a,5}) \cdot r(sp)_{1,5} + rg(w_{u_a,6}) \cdot r(sp)_{1,6}}{|r(sp)_{1,2}| + |r(sp)_{1,3}| + |r(sp)_{1,4}| + |r(sp)_{1,5}| + |r(sp)_{1,6}|}$$
$$= \frac{3 \cdot 0 + 1 \cdot 1 + [NA \cdot (-0{,}24)] + 2 \cdot 1 + [NA \cdot (-1)]}{|0| + |1| + [|0{,}24|] + |1| + [(-1)]} = \frac{1+2}{1+1} = \frac{3}{2} = 1{,}5$$

Die Prognoseergebnisse mit den Korrelationskoeffizienten nach Bravais-Pearson und Spearman lassen sich nicht direkt vergleichen. Im ersten Fall erhalten wir eine Bewertung von $4{,}1963$ was einem gut bis sehr gut entspricht. Im zweiten Fall ist man mit einem Rangplatz von $1{,}5$ nahe am bestmöglichen Rangplatz 1. Beide Ergebnisse sind sich also sehr ähnlich.

3.2.2 Prognosen mit User-based CF

User-based CF verwendet die Korrelationen der Bewertungen zwischen den Usern. Eine einfache Prognose für die Bewertung des Users u_a für Item v_b lässt sich anhand folgender Formel erstellen (Su & Khoshgoftaar, 2009):

$$P(bp)^{UB}_{u_a,v_b} = \bar{w}_{u_a} + \frac{\sum_{i=1}^{n}(w_{i,b} - \bar{w}_{u_i}) \cdot r(bp)_{u_a,u_i}}{\sum_{i=1}^{n}|r(bp)_{u_a,u_i}|}$$

mit $\bar{w}_{u_a} = 1/m \sum_{j=1}^{m} w_{a,j} \quad und \quad \bar{w}_{u_i} = 1/m \sum_{j=1}^{m} w_{i,j}$

Dabei ist $\bar{w}_{u_a}$ der Mittelwert der Bewertungen des aktiven Users u_a und $\bar{w}_{u_i}$ sind die Mittelwerte der Bewertungen der jeweiligen anderen User über die jeweils bewerteten Items v_b. $r(bp)_{u_a,u_i}$ sind die Korrelationskoeffizienten nach Bravais-Pearson zwischen dem aktiven User und den jeweiligen anderen Usern (vgl. Kapitel 3.1.1). Tabelle 11 zeigt die Korrelationskoeffizienten nach Bravais-Pearson zwischen dem aktiven User und den jeweiligen anderen Usern.

	u_1	u_2	u_3	u_4	u_5
$r(bp)_{u_a,u_i}$	-0,72	-1,00	-1,00	1,00	1,00

Tabelle 11: Korrelationen nach Bravais-Pearson zwischen $\boldsymbol{u_a}$ und den anderen Usern

Auf Basis dieser Daten und der User-Item-Matrix (Tabelle 1) lässt sich mit obiger Formel beispielsweise eine Prognose für das Item v_1 des aktiven Users u_a erstellen:

$$P(bp)^{UB}_{u_a,v_1} = \bar{w}_{u_a} + \frac{\sum_{i=1}^{n}(w_{i,1} - \bar{w}_{u_i}) \cdot r(bp)_{u_a,u_i}}{\sum_{i=1}^{n}|r(bp)_{u_a,u_i}|}$$
$$= \bar{w}_{u_a} + \frac{(w_{1,1} - \bar{w}_{u_1}) \cdot r(bp)_{u_a,u_1} + (w_{2,1} - \bar{w}_{u_2}) \cdot r(bp)_{u_a,u_2} + \cdots + (w_{5,1} - \bar{w}_{u_5}) \cdot r(bp)_{u_a,u_5}}{|r(bp)_{u_a,u_1}| + |r(bp)_{u_a,u_2}| + \cdots + |r(bp)_{u_a,u_5}|} =$$
$$= \frac{10}{3} + \frac{\left[\left(NA - \frac{8}{4}\right) \cdot (-0{,}5)\right] + \left(1 - \frac{16}{5}\right) \cdot (-1) + \left(2 - \frac{14}{4}\right) \cdot (-1) + \left(1 - \frac{11}{5}\right) \cdot (1) + \left(2 - \frac{12}{5}\right) \cdot (1)}{[(-0{,}5)] + |-1| + |-1| + |1| + |1|}$$
$$= 3{,}33 + \frac{2{,}2 + 1{,}5 + (-1{,}2) + (-0{,}4)}{4} = 3{,}33 + \frac{2{,}1}{4} = 3{,}855$$

Anhand des Beispiels kann man gut die Idee, die hinter der Prognoseformel steht, erkennen: Ausgangspunkt ist der Mittelwert $\bar{w}_{u_a}$ der Bewertungen des aktiven Users. Addiert werden zu diesem Wert die Produkte der normierten Bewertungen $w_{i,1} - \bar{w}_{u_i}$ der anderen User für das Item v_1 multipliziert mit den jeweiligen Korrelationen $r(bp)_{u_a,u_i}$ der User u_i mit dem aktiven User u_a. Korrelieren die Bewertungen $w_{i,j}$ eines User u_i mit denen des aktiven Users u_a positiv, dann wird die normierte Bewertung $w_{i,1} - \bar{w}_{u_i}$ des User u_i zum Mittelwert $\bar{w}_{u_a}$ der Bewertungen des aktiven Users addiert. Bei negativer Korrelation wird die normierte Bewertung $w_{i,1} - \bar{w}_{u_i}$ des

User u_i vom Mittelwert $\overline{w}_{u_a}$ der Bewertungen des aktiven Users subtrahiert. Man beachte: Da $w_{1,1} = NA$ entfällt im Zähler der Ausdruck $\left(w_{1,1} - \overline{w}_{u_1}\right) \cdot r(bp)_{u_a,u_1}$ und im Nenner $\left|r(bp)_{u_a,u_1}\right|$.

Wie man sieht weicht der Prognosewert $P(bp)^{UB}_{u_a,v_1} = 3{,}855$ nach dem User-based Verfahren nur leicht von dem Wert $P(bp)^{IB}_{u_a,v_1} = 4{,}1963$ ab, der mit dem Item-based Verfahren ermittelt wurde.

Analog zu Prognosen mit Item-based CF können auch mit User-based CF Prognosen für den Rang mittels der folgenden Formel erstellt werden:

$$P(sp)^{UB}_{u_a,v_b} = \overline{rg}_{u_a} + \frac{\sum_{i=1}^{n}\left(rg(w_{i,b}) - \overline{rg}_{u_i}\right) \cdot r(sp)_{u_a,u_i}}{\sum_{i=1}^{n}\left|r(sp)_{u_a,u_i}\right|}$$

mit $\overline{rg}_{u_a} = \frac{1}{m}\sum_{j=1}^{m} rg(w_{a,j})$ und $\overline{rg}_{u_i} = \frac{1}{m}\sum_{j=1}^{m} rg(w_{i,j})$.

	u_1	u_2	u_3	u_4	u_5
$r(sp)_{u_a,u_i}$	-0,50	-1,00	-1,00	1,00	1,00

Tabelle 12: Korrelationen nach Spearman zwischen $\boldsymbol{u_a}$ und den anderen Usern

Mit den Beispieldaten und den Korrelationskoeffizienten nach Spearman zwischen dem aktiven User und den anderen Usern (Tabelle 12) ergibt sich folgende Prognose für den Rang von Items v_1 des aktiven Users u_a

$$\begin{aligned}
P(sp)^{UB}_{u_a,v_1} &= \overline{rg}_{u_a} + \frac{\sum_{i=1}^{n}\left(rg(w_{i,1}) - \overline{rg}_{u_i}\right) \cdot r(sp)_{u_a,u_i}}{\sum_{i=1}^{n}\left|r(sp)_{u_a,u_i}\right|} \\
&= \overline{rg}_{u_a} + \frac{\left(rg(w_{1,1}) - \overline{rg}_{u_1}\right) \cdot r(sp)_{u_a,u_1} + \left(rg(w_{2,1}) - \overline{rg}_{u_2}\right) \cdot r(sp)_{u_a,u_2} + \cdots + \left(rg(w_{5,1}) - \overline{rg}_{u_5}\right) \cdot r(sp)_{u_a,u_5}}{\left|r(sp)_{u_a,u_1}\right| + \left|r(sp)_{u_a,u_2}\right| + \cdots + \left|r(sp)_{u_a,u_5}\right|} \\
&= 2 + \frac{[(NA - 2{,}5) \cdot (-0{,}5)] + (5-3) \cdot (-1) + (4 - 2{,}5) \cdot (-1) + (4-3) \cdot (1) + (3{,}5 - 3) \cdot (1)}{[(-0{,}5)] + |-1| + |-1| + |1| + |1|} \\
&= 2 + \frac{-2 - 1{,}5 + 1 + 0{,}5}{4} = 2 + \frac{-2}{4} = 1{,}5
\end{aligned}$$

Wie man sieht gibt es in diesem Beispiel keine Abweichung zwischen der Item-based und User-based Rangprognose $P(sp)^{IB}_{u_a,v_1} = P(sp)^{UB}_{u_a,v_1} = 1{,}5$.

3.3 Empfehlungen

Empfehlungen entsprechen der originären Aufgabe eines Recommendersystems. Herlocker (Herlocker, Konstan, Terveen, & Riedl, 2004) hat diese Aufgabe „find good items“ genannt. Dabei werden grob gesprochen Items in eine Rangordnung gebracht, so dass das Item, das für den aktiven User am interessantesten ist, an erster Stelle und das Item, das für den aktiven User am wenigsten interessant ist, an letzter Stelle steht. Im Zusammenhang mit Recommendersystemen fällt auch häufig der Begriff Top N-Empfehlungen. Dabei steht N für die Anzahl der Items, die dem aktiven User empfohlen werden. Anders formuliert handelt es sich bei Top N-Empfehlungen um eine auf die N interessantesten Items verkürzte Rangordnung.

3.3.1 Empfehlungen mit Item-based CF

Für die Erstellung der Rangliste verwendet Item-based CF die Ähnlichkeit zwischen den Items (Karypis, November 2001). In Kapitel 3.3.1.1 wird zunächst der Algorithmus und im nachfolgenden Kapitel eine Variante des Algorithmus beschrieben.

3.3.1.1 Item-based Algorithmus

Der Algorithmus für die Ermittlung von Empfehlungen mittels Item-based CF kann in sechs Schritte unterteilt werden und wird anhand der Beispieldaten aus Tabelle 1 vorgestellt.

Schritt 1: Berechnung der Ähnlichkeitsmatrix der Items
Als Ähnlichkeitsmaß wird der Korrelationskoeffizient nach Spearman und die entsprechende Korrelationsmatrix der Items (Tabelle 7) verwendet. Der Algorithmus funktioniert analog mit den anderen vorgestellten Ähnlichkeitsmaßen (vgl. Kapitel 3.1).

Schritt 2: Identifikation der für den aktiven User u_a relevanten Items
Da die Empfehlungsliste für den aktiven User u_a erstellt werden soll, benötigt man zunächst die Bewertungen dieses Users (Tabelle 10). Da dem User u_a nur Items empfohlen werden sollen, die für ihn relevant sind, müssen die Items identifiziert werden, die der aktive User hoch bewertet hat. Es wird hier angenommen, dass Items für u_a relevant sind, wenn er diese mit 4 oder 5 bewertet hat. Im Beispiel trifft dies für die Items v_3 und v_5 zu.

Schritt 3: Reduktion der Korrelationsmatrix
Um die Items zu finden, die eine hohe Korrelation zu den Items haben, die für u_a relevant sind, wird die Korrelationsmatrix (Tabelle 7) auf die Zeilen reduziert, welche die für u_a relevanten Items enthalten. Im Beispiel sind dies die Zeilen 3 und 5. Es entsteht folgende reduzierte Korrelationsmatrix (Tabelle 13):

	v_1	v_2	v_3	v_4	v_5	v_6
v_3	1,00	-0,50		-0,87	1,00	-0,50
v_5	1,00	-1,00	1,00	1,00		1,00

Tabelle 13: Reduzierte Korrelationsmatrix der Items

Schritt 4: Aggregation der reduzierten Korrelationsmatrix
Für die Ermittlung der für u_a interessantesten Items werden jetzt die Spalten der reduzierten Korrelationsmatrix (Tabelle 13) aggregiert, in diesem Fall summiert. Es entsteht der Scorevektor s' für die Items (Tabelle 14). Dieser enthält für v_1 den Score 2, der für diese Konstellation maximal ist, da beide für u_a relevanten Items v_3 und v_5 eine Korrelation nach Spearman von 1 hatten. Dementsprechend passt v_1 perfekt zu v_3 und v_5.

	v_1	v_2	v_3	v_4	v_5	v_6
s'	2,00	-1,50	1,00	0,13	1,00	0,50

Tabelle 14: Scorevektor (Item-based CF)

Schritt 5: Reduktion des Scorevektors
Wenn man davon ausgeht, dass Items, die bewertet wurden von u_a auch „gekauft“ wurden, dann macht es keinen Sinn diese Items in eine Empfehlungsliste aufzunehmen. Daher wird der Scorevektor s' auf die Items reduziert, die u_a noch nicht bewertet hat. In unserem Beispiel sind die Items v_1, v_4 und v_6. Es entsteht der Scorevektor s'' (Tabelle 15).

	v_1	v_2	v_3	v_4	v_5	v_6
s''	2,00			0,13		0,50

Tabelle 15: Reduzierter Scorevektor (Item-based CF)

Schritt 6: Bildung der Rangliste
Abschließend werden die Items in die Rangordnung (Tabelle 16) gebracht, die die Empfehlungsreihenfolge widerspiegelt. Im Beispiel bekommt v_1 Rangplatz 1, da es das für u_a nach Item-based CF interessanteste Item

aufgrund der höchsten aggregierten Korrelation zu den für u_a relevanten Items v_3 und v_5 ist. Das Item v_6 bekommt Rangplatz 2, da es die zweithöchste aggregierte Korrelation zu den für u_a relevanten Items hat. Für das Item v_4 verbleibt Rangplatz 3.

	v_1	v_2	v_3	v_4	v_5	v_6
rk	1			3		2

Tabelle 16: Rangliste der Items (Item-based CF)

Abbildung 5 fasst die Vorgehensweise bei der Ermittlung von Empfehlungen mit Item-based CF zusammen. Die einzelnen Nummern in der Grafik entsprechen den Schritten in obiger Beschreibung.

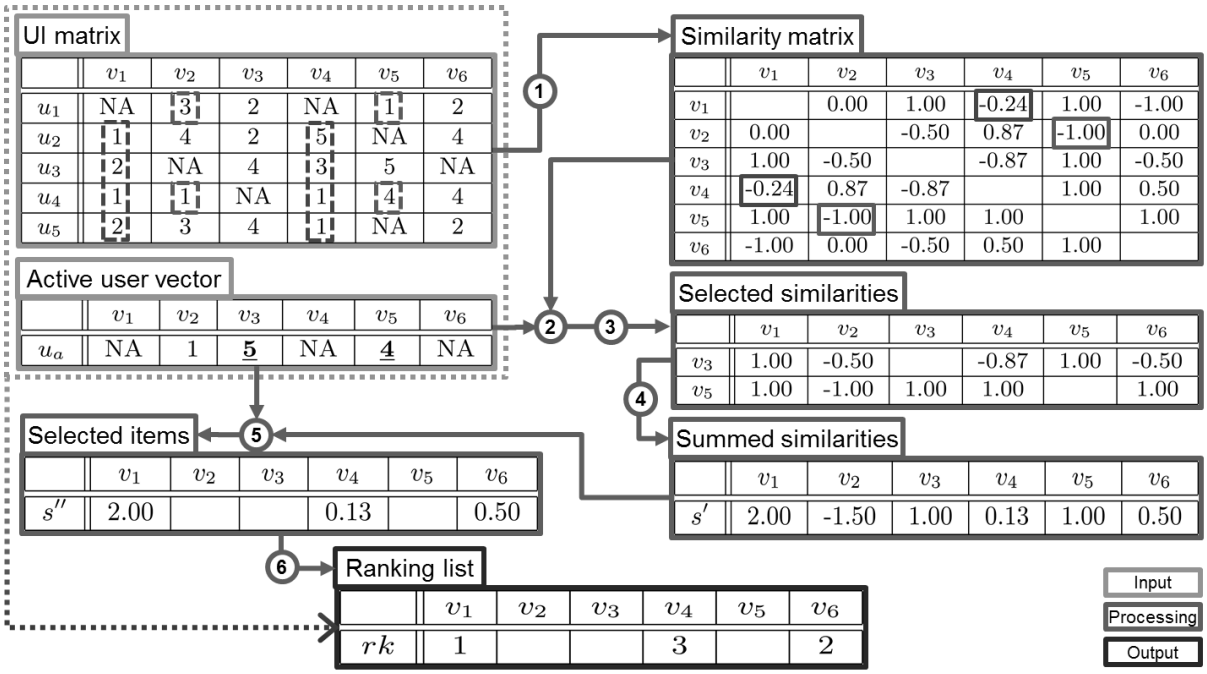

Abbildung 5: Ermittlung von Empfehlungen mit Item-based CF

3.3.1.2 Variante k nearest neighbours für Item-based CF

Die Variante „k nearest neighbours“ betrachtet nur die k-ähnlichsten Items der relevanten Items des aktiven Nutzers. D.h. der Algorithmus (vgl. Kapitel 3.3.1.1) ändert sich ab Schritt 3. Zunächst ordnet man die Werte der reduzierten Korrelationsmatrix (Tabelle 13) in absteigender Reihenfolge. Tabelle 17 zeigt das Ergebnis.

	Wert
$r(sp)_{v_3,v_1}$	1,00
$r(sp)_{v_5,v_1}$	1,00
$r(sp)_{v_5,v_3}$	1,00
$r(sp)_{v_5,v_4}$	1,00
$r(sp)_{v_3,v_5}$	1,00

	Wert
$r(sp)_{v_5,v_6}$	1,00
$r(sp)_{v_3,v_2}$	-0,50
$r(sp)_{v_3,v_6}$	-0,50
$r(sp)_{v_3,v_4}$	-0,87
$r(sp)_{v_5,v_2}$	-1,00

Tabelle 17: Korrelationen in absteigender Reihenfolge

Setzt man z. B. $k = 2$ so betrachtet man nur die zwei Items, die den relevanten Items des aktiven Users am ähnlichsten sind. Hier sind dies die Items bei denen der Korrelationskoeffizient den Wert 1,00 hat und trifft für sechs Itemkombinationen zu. Man kann nun per Zufall auf $k = 2$ Kombinationen reduzieren oder mit allen sechs Kombinationen weiterrechnen. Abbildung 6 zeigt die weiteren Schritte des Algorithmus, wenn man sich für die sechs Itemkombinationen entschieden hat.

Schritt: 3

	v_1	v_2	v_3	v_4	v_5	v_6
v_3	1.00	~~-0.50~~		~~-0.87~~	1.00	~~-0.50~~
v_5	1.00	~~-1.00~~	1.00	1.00		1.00

	v_1	v_2	v_3	v_4	v_5	v_6
~~s'~~	~~2.00~~	~~1.50~~	~~1.00~~	~~0.13~~	~~1.00~~	~~0.50~~
s'	2.00	0.00	1.00	1.00	1.00	1.00

Schritt: 4

	v_1	v_2	v_3	v_4	v_5	v_6
~~s''~~	~~2.00~~			~~0.13~~		~~0.50~~
s''	2.00			1.00		1.00

Schritt: 5

	v_1	v_2	v_3	v_4	v_5	v_6
~~rk~~	~~1~~			~~3~~		~~2~~
rk	1.0			2.5		2.5

Abbildung 6: Item-based Algorithmus für $k = 2$ nearest neighbours

3.3.2 Empfehlungen mit User-based CF

Dieser Algorithmus verwendet als Grundlage die Ähnlichkeiten zwischen den Usern. In Kapitel 3.3.2.1 wird zunächst wieder der Algorithmus und im nachfolgenden Kapitel eine Variante des Algorithmus beschrieben (Su & Khoshgoftaar, 2009).

3.3.2.1 User-based Algorithmus

Der Algorithmus für die Ermittlung von Empfehlungen mittels User-based CF kann in fünf Schritte unterteilt werden und wird wieder anhand der Beispieldaten aus Tabelle 1 vorgestellt.

Schritt 1: Berechnung des Ähnlichkeitsvektors zwischen dem aktiven User und den anderen Usern
Auf Basis der User-Item-Matrix wird ein Ähnlichkeitsvektor der User aufgebaut, der die Korrelationen des aktiven Users mit allen anderen Usern enthält. Als Ähnlichkeitsmaß wird wieder der Korrelationskoeffizient nach Spearman verwendet. Der Algorithmus funktioniert analog mit den anderen vorgestellten Ähnlichkeitsmaßen (vgl. Kapitel 3.1). Tabelle 12 zeigt diesen Ähnlichkeitsvektor, der bereits in Kapitel 3.2.2 für die Prognose nach User-based CF verwendet wurde.

Schritt 2: Berechnung der mit den Korrelationskoeffizienten aus Schritt 1 gewichteten User-Item-Matrix
Die Bewertungen $w_{i,j}$ der User-Item-Matrix (Tabelle 1) werden mit Korrelationskoeffizienten $r(sp)_{u_a,u_j}$ aus Schritt 1 des Algorithmus nach der Formel $w^*_{i,j} = r(sp)_{u_a,u_j} \cdot w_{i,j}$ gewichtet. Tabelle 18 zeigt die User-Item-Matrix mit den gewichteten Bewertungen $w^*_{i,j}$.

	v_1	v_2	v_3	v_4	v_5	v_6
u_1	NA	-1,50	-1,00	NA	-0,50	-1,00
u_2	-1,00	-4,00	-2,00	-5,00	NA	-4,00
u_3	-2,00	NA	-4,00	-3,00	-5,00	NA
u_4	1,00	1,00	NA	1,00	4,00	4,00
u_5	2,00	3,00	4,00	1,00	NA	2,00

Tabelle 18: Gewichtete User-Item-Matrix

Schritt 3: Aggregation der gewichteten User-Item-Matrix
Nun wird der Median der gewichteten Bewertungen pro Item gebildet. Bei gerader Anzahl der Bewertungen wurde hier der Mittelwert zwischen den „mittleren" Bewertungen als Median verwendet. Es entsteht der Scorevektor s', der die Mediane pro Item (Tabelle 19) enthält.

	v_1	v_2	v_3	v_4	v_5	v_6
s'	0,00	-0,25	-1,50	-1,00	-0,50	0,50

Tabelle 19: Scorevektor (User-based CF)

Schritt 4: Reduktion des Scorevektors
Es wird wiederum der Scorevektor s' auf die Items reduziert, die der aktive User noch nicht bewertet hat. In unserem Beispiel sind das die Items v_1, v_4 und v_6. Es entsteht der Scorevektor s'' (Tabelle 20).

	v_1	v_2	v_3	v_4	v_5	v_6
s''	0,00			-1,00		0,50

Tabelle 20: Reduzierter Scorevektor (User-based CF)

Schritt 5: Bildung der Rangliste
Abschließend werden die Items wieder in eine Rangordnung gebracht (Tabelle 21). Im Beispiel bekommt v_6 Rangplatz 1, da es das für u_a nach User-based CF interessanteste ist. Das Item v_1 bekommt Rangplatz 2 und für Item v_4 verbleibt wieder Rangplatz 3. Im Vergleich zu Item-based CF wurden also lediglich die Rangplätze von v_1 und v_6 getauscht.

	v_1	v_2	v_3	v_4	v_5	v_6
rk	2			3		1

Tabelle 21: Rangliste der Items (User-based CF)

Abbildung 7 fasst die Vorgehensweise bei der Ermittlung von Empfehlungen mit User-based CF zusammen. Die einzelnen Nummern in der Grafik entsprechen wieder den Schritten in obiger Beschreibung.

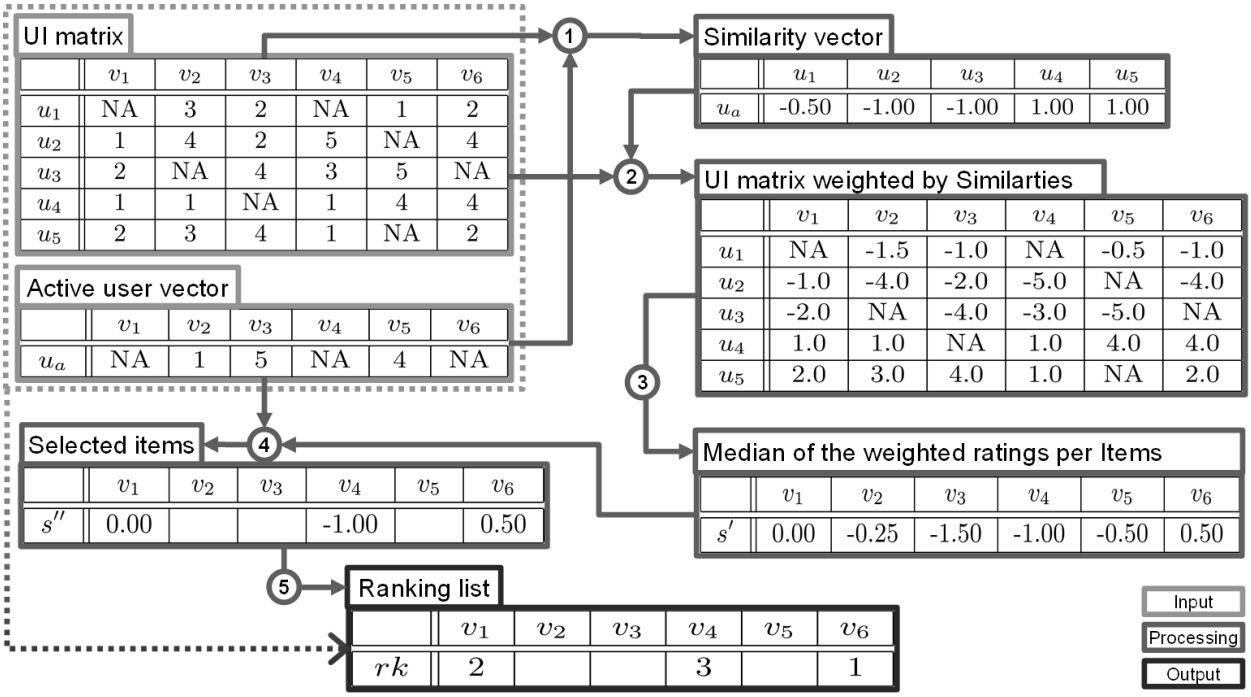

Abbildung 7: Ermittlung von Empfehlungen mit User-based CF

3.3.2.2 Variante k nearest neighbours für User-based CF

Die Variante „k nearest neighbours“ betrachtet nur die k-ähnlichsten User des aktiven Users. Der User-based Algorithmus (vgl. Kapitel 3.3.2.1) ändert sich hier ab Schritt 2. Abbildung 8 fasst die Vorgehensweise für $k = 2$ zusammen.

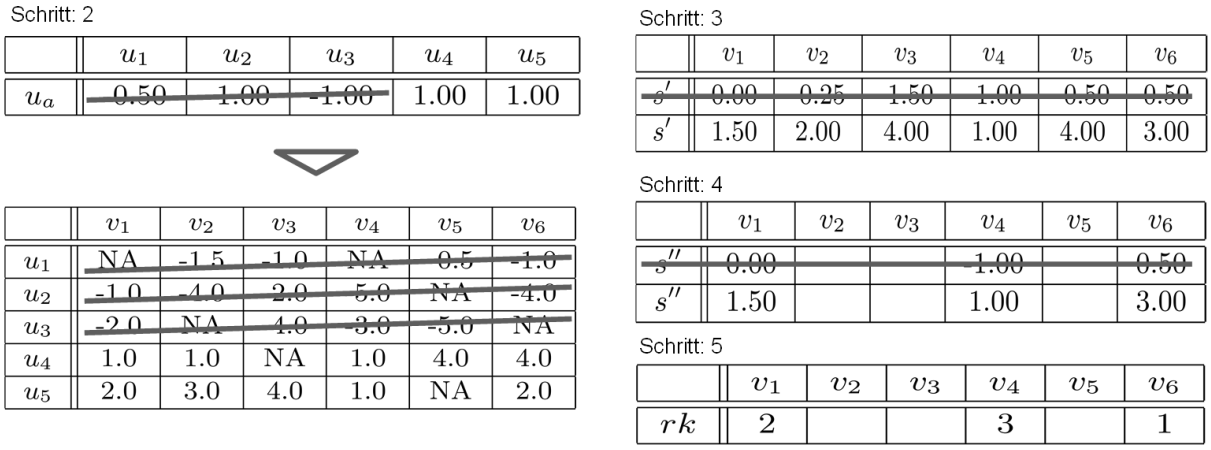

Abbildung 8: User-based Algorithmus für $k = 2$ nearest neighbours

Die beiden User u_4 und u_5 sind die dem aktiven User am ähnlichsten. Nur noch diese beiden User werden für die weiteren Schritte verwendet. Das Ergebnis ist im Beispiel identisch mit dem User-based Algorithmus (vgl. Kapitel 3.3.2.1).

3.4 Metriken für die Evaluierung

Herlocker (Herlocker, Konstan, Terveen, & Riedl, 2004) teilt die Metriken für die Genauigkeit von Recommendersystemen im Wesentlichen in zwei Klassen ein: Metriken für die Prognosegenauigkeit und Metriken für die Genauigkeit der Klassifikation.

3.4.1 Metriken für die Prognosegenauigkeit

Diese Metriken messen wie gut die Prognose des Recommender Systems im Vergleich zur tatsächlichen Bewertung des Users ist. Diese Metriken sind besonders interessant, wenn die prognostizierte Bewertung für ein Item dem User während der Systemnutzung als Hinweis angezeigt wird. Da man über die Prognosewerte auch ein Ranking der Items erstellen kann, kann die Prognosegenauigkeit auch als Maß für Fähigkeit des Recommender Systems, Items nach den Präferenzen des Users zu ordnen, verwendet werden. Ein häufig genannter Vertreter dieser Metriken ist Mean Absolute Error (MAE), der den Durchschnitt der absoluten Differenz zwischen den Prognose- und den tatsächlichen Bewertungen der User bestimmt.

$$MAE = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\left|P_{i,j} - r_{i,j}\right|}{n}$$

für alle $r_{i,j} \neq \{NA\}$
Neben MAE zählen auch Mean Square Error, Root Mean Squared Error und Normalized Mean Absolute Error zu den Vertretern dieser Klasse von Metriken (Herlocker, Konstan, Terveen, & Riedl, 2004), (Su & Khoshgoftaar, 2009).

3.4.2 Metriken für die Genauigkeit der Klassifikation

Diese Metriken messen die Häufigkeit mit der ein Recommender System richtige oder falsche Entscheidungen für die Empfehlung von Items trifft, in Abhängigkeit dessen, ob die Items tatsächlich relevant sind.

3.4.2.1 Precision und Recall

Als sehr bekannte Metriken in dieser Kategorie gelten Precision und Recall. Diese kommen ursprünglich aus dem Bereich Information Retrieval und werden seit den späten 1960er Jahre verwendet. Um Precision und Recall zu verstehen betrachten wir zunächst folgende Vier-Felder-Tafel mit den beiden Merkmalen Relevanz eines Items und Auswahl/Empfehlung eines Items:

	gefunden/ empfohlen (g)	nicht gefunden/ nicht empfohlen (n)	Summe
relevant (r)	$N_{r,g}$	$N_{r,n}$	N_r
irrelevant (i)	$N_{i,g}$	$N_{i,n}$	N_i
Summe	N_g	N_n	N

Tabelle 22: Kategorien von Items

Precision ist definiert als Quotient aus relevanten, gefundenen/empfohlenen Items und allen gefundenen/empfohlenen Items:

$$0 \leq P = \frac{N_{r,g}}{N_g} \leq 1$$

Recall ist der Quotient aus relevanten, gefundenen/empfohlenen Items und allen relevanten Items:

$$0 \leq R = \frac{N_{r,g}}{N_r} \leq 1$$

3.4.2.2 Receiver Operatering Characteristics curves

Receiver Operatering Characteristics (ROC) curves und die jeweils korrespondierende Kennzahl AUC (Area Under Curve) sind sehr anschauliche Metriken für die Evaluation der Empfehlungsgenauigkeit. Die ROC-Kurve ist eine grafische Darstellung des Zusammenhangs zwischen der True Positive Rate (TPR) auf der Y-Achse und der False Positive Rate (FPR) auf der X-Achse. Die TPR gibt den Anteil der empfohlenen Items, die auch tatsächlich für einen User relevant sind und allen relevanten Items. Die FPR ist der Anteil der irrelevanten Items, die dem User empfohlen wurden, unter allen irrelevanten Items. Zur Verdeutlichung betrachten wir die Vier-Felder-Tafel (Tabelle 22) mit den Ausprägungen „nicht empfohlen“, „ empfohlen“ und „irrelevant“ „relevant“ und die dazugehörigen absoluten Häufigkeiten.
Ein einfaches Beispiel soll obigen Begriffe und Metriken veranschaulichen: Gehen wir davon aus, ein Recommenderalgorithmus hat die 3 für einen User relevanten Items unter insgesamt 10 Items auf den Positionen 2, 4 und 7 empfohlen. Betrachtet man einen TOP-5-Recommender ergeben sich folgende Werte für die Elemente der Vier-Felder-Tafel (Tabelle 22): $N_{r,g} = 2, N_r = 3, N_{i,g} = 3, N_i = 7$. Für die TPR ergibt sich $\frac{N_{r,g}}{N_r} = {}^{2}/_{3} = 0{,}67$ und für FPR erhält man $\frac{N_{i,g}}{N_i} = {}^{3}/_{7} = 0{,}43$. Das heißt, dass 67% der relevanten Items und 43% der nicht relevanten Items vom TOP-5-Recommender empfohlen wurden.

Um die ROC-Kurve zu zeichnen, muss man schrittweise nach folgendem Algorithmus vorgehen:

- Es wird eine Rangliste der Items erstellt.
- Der Startpunkt der Grafik ist der Koordinatenursprung.
- Beginnend mit dem ersten Item der Rangliste wird pro Item die TPR und die FPR berechnet.
- Die ROC-Kurve ist eine Stufenfunktion der jeweils berechneten Punktepaare aus FPR und TPR

Tabelle 23 zeigt die jeweilige TPR und FPR für das obige Beispiel und Abbildung 9 die zugehörige ROC-Kurve.

Rangliste der empfohlenen Items	1	2	3	4	5	6	7	8	9	10
Item relevant $(0/1)$	0	1	0	1	0	0	1	0	0	0
True Positive Rate	$\frac{0}{3} = 0$	$\frac{1}{3} = 0{,}33$	0,33	0,67	0,67	1,00	1,00	1,00	1,00	1,00
False Positive Rate	$\frac{1}{7} = 0{,}14$	$\frac{1}{7} = 0{,}14$	0,29	0,29	0,43	0,57	0,57	0,71	0,86	1,00

Tabelle 23: True Positive Rate und False Positive Rate

Ein perfekter Recommender wird eine ROC-Kurve erzeugen, die aus einer vertikalen Line von 0 bis 1 und anschließend in einer horizontalen Line vom Punkt $(0|1)$ zum Punkt $(1|1)$ verläuft. Das heißt, er findet nur relevante Items bis alle relevanten Items gefunden sind. Ein zufälliger Recommender würde nicht zwischen irrelevanten und relevanten Items unterscheiden. Die ROC-Kurve verläuft dann entlang der Winkelhalbierenden. Die Fläche unterhalb der ROC-Kurve heißt AUC und ist eine Messgröße für die Qualität des Recommenderalgorithmus. Die Empfehlungsqualität ist umso besser, je größer der AUC-Wert ist. Der AUC-Wert kann auch als Wahrscheinlichkeit interpretiert werden, dass ein relevantes Item auch tatsächlich als solches klassifiziert wird.

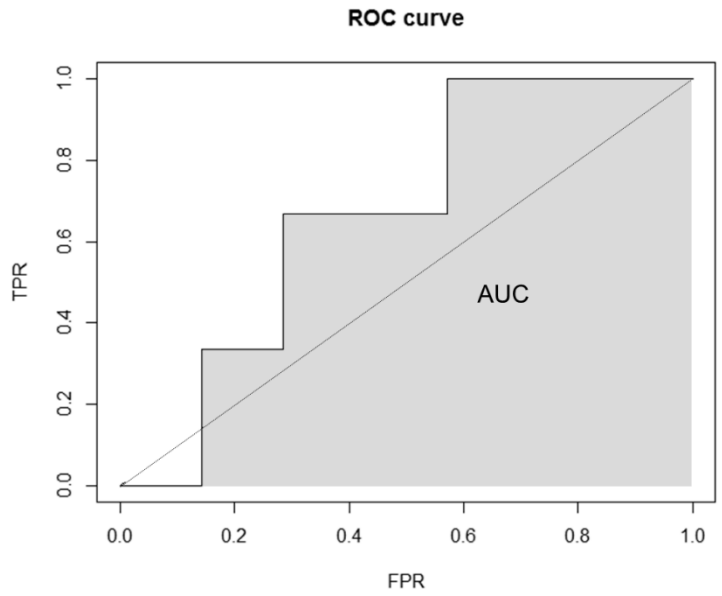

Abbildung 9: ROC-Kurve

4 Beispielhafte Simulationsstudie

Neben einem Grundverständnis zur Funktionsweise von Recommenderalgorithmen ist es zur Beteiligung an der Forschung in diesem Bereich von zentraler Bedeutung, den Aufbau von Simulationsstudien zur Feststellung der Empfehlungsqualität zu verstehen. Diese Empfehlungsqualität kann zwischen verschiedenen Algorithmen, Varianten oder Parametrierungen gemessen und anschließend verglichen werden.
Um dies didaktisch aufzuarbeiten wird in diesem Kapitel zunächst beispielhaft eine Simulationsstudie durchgeführt und die Ergebnisse der Studie kurz erläutert. In der vorliegenden Untersuchung steht jedoch die Darstellung des Konzepts einer Simulationsstudie im Vordergrund, die eigentlichen Ergebnisse werden also nur nachrangig behandelt.

4.1 Datensätze

Als Datenbasis für die Simulationen werden die Bewertungen des MovielLens-100k-Datensatz verwendet. Das MovieLens-Projekt wird von der University of Minnesota unterhalten und gibt anonymisierte Datensätze für die Forschung an Recommenderalgorithmen frei. Im 100k-Datensatz sind ca. 100.000 Bewertungen von 943 Benutzern bezüglich 1.682 Filmen auf einer Skala von 1 bis 5 Sternen enthalten.
Für die Simulationsstudie wurde der 100k-Datensatz hinsichtlich der Benutzer in insgesamt 50 jeweils disjunkte Trainingsdatensätze ($80\,\%$: 754 Benutzer) und Testdatensätze ($20\,\%$: 189 Benutzer) aufgeteilt. Dies ist eine gängige Vorgehensweise bei der Evaluation von Recommenderalgorithmen.

4.2 Metrik

Um die Empfehlungsqualität der Recommenderalgorithmen zu messen, werden die bereits in Kapitel 3.4.2.2 beschriebenen ROC-Kurven verwendet. Da ROC-Kurven in einem Diagramm die TPR gegen die FPR antragen, ist zunächst die Frage zu stellen, wie True Positive bzw. False Positive im Recommenderkontext definiert werden können. Ein Lösungsansatz besteht darin, dem Recommenderalgorithmus bestimmte Informationen vorzuenthalten und im Anschluss zu überprüfen, ob sich die durch den Algorithmus generierten Empfehlungen konsistent gegenüber den vorenthaltenen Informationen verhalten.
In der vorliegenden Simulationsstudie wird für jeden User der Testdatensätze ein sogenanntes „relevantes Item“ selektiert, das dieser mit 5 Sternen bewertet hat, und mit NA ausgetauscht. Anschließend werden 499 „irrelevante Items“ ausgesucht, für die zusammen mit dem relevanten Item eine Empfehlungsrangliste erstellt wird.

Der Rang des relevanten Items kann dann Aufschluss darüber geben, wie die Empfehlungen des Algorithmus wirklich sind – wird das relevante Item relativ weit vorne empfohlen spricht dies für eine hohe Empfehlungsqualität, liegt es eher auf den letzten Rängen geht man von einer niedrigen Empfehlungsqualität aus.
Um die ROC-Kurve nun zu zeichnen, wird für jeden der 500 Plätze auf der Empfehlungsrangliste pro User die TPR und FPR ermittelt. Hierbei gibt es pro User in den Testdatensätzen genau einen Sprung abhängig von dem relativen Rang des relevanten Items. Wird diese beispielsweise genau auf dem Rang 250 empfohlen, ergibt sich ein relativer Rang von $250/500 = 0{,}5$. Demzufolge erhält man für die Ränge 1 bis 249 eine TPR von 0 und ab einer FPR von $0{,}5$ einer TPR von 1 – die ROC-Kurve springt also bei einem Wert von $0{,}5$ auf der X-Achse auf der Y-Achse von dem Wert 0 auf den Wert 1.
Wird nun nicht lediglich ein User, sondern alle User der Testdatensätze betrachtet, erhält man durch die Aggregation der Ergebnisse eine Treppenfunktion, die einer Kurve gleicht. Zusätzlich abstrahieren die Ergebnisse dadurch vom einzelnen User und werden repräsentativ für die Grundgesamtheit.

4.3 Aufteilung in Trainings- und Testdaten

Eine Aufteilung in disjunkte Trainings- und Testdatensätze ist nötig, um die reale Problemlage von Recommendersystemen zu erfassen. In der Realität versucht ein Recommenderalgorithmus bestimmte Strukturen in ihm vorliegenden Daten zu erfassen und auf deren Basis möglichst gute Empfehlungen zu erzeugen.
In der Simulation werden dem Recommenderalgorithmus nur die Daten des Trainingsdatensatzes zur Verfügung gestellt. Deren Strukturen soll der Algorithmus dann so gut wie möglich erfassen. Anschließend werden durch den Algorithmus Empfehlungen für die User (bzw. die relevanten Items) des Testdatensatzes erzeugt und diese mit den tatsächlichen Bewertungen verglichen. Dieser Vergleich gibt Aufschluss über die Empfehlungsqualität des Recommenderalgorithmus.
Um die Validität der Ergebnisse zu steigern und belastbare Schlussfolgerungen abzuleiten, wird der Vorgang der Aufteilung in Trainings- und Testdaten nicht nur einmal vorgenommen. Die Aufteilung des Datensatzes könnte sich schließlich zufällig positiv oder negativ auf die Empfehlungsqualität auswirken. Stattdessen wird der Datensatz mehrfach in jeweils disjunkte Trainings- und Testdatensätze aufgeteilt und der Empfehlungsvorgang damit wiederholt. Im Zuge dessen erhält man verlässlichere Ergebnisse.

4.4 Simulationsaufbau

Der Aufbau der Simulationsstudie ist für ein Paar von Trainings- und Testdatensätzen in Abbildung 10 dargestellt. Zunächst erfolgt die Aufteilung des MovieLens-100k-Datensatzes. Anschließend werden für den Datensatz für vier Ähnlichkeitsmaße mit zwei Algorithmen – also insgesamt acht – Simulationen durchgeführt.

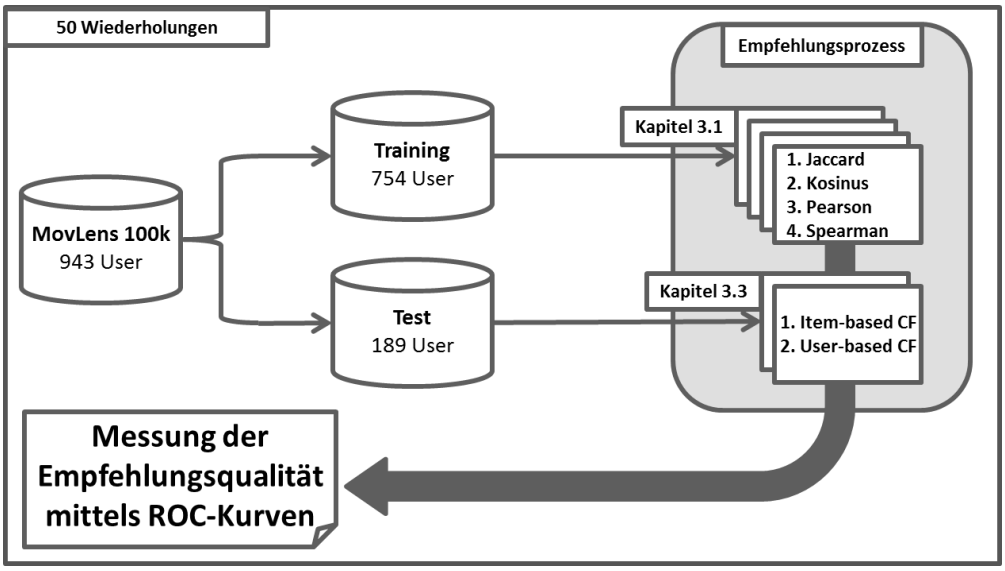

Abbildung 10: Aufbau der Simulationsstudie

Die Ränge der relevanten Items werden anschließend gespeichert. Pro Testdatensatz erhält man also acht Mal 189 Ränge für das jeweilige relevante Item. Im Anschluss an die Simulationen werden aus diesen Ergebnissen acht ROC-Kurven erzeugt und die zugehörigen AUC-Werte berechnet. Diese basieren dann auf jeweils 9.450 Rängen, die den relevanten Items zugewiesen wurden.
Als Umgebung zur Durchführung der Simulationsstudie bietet sich das Statistik-Software-Paket R an. R ist eine unter der GNU-Lizenz frei verfügbare Programmiersprache mit zugehöriger Laufzeitumgebung, die bei wissenschaftlichen Analysen immer mehr Anwendung findet und in einigen Disziplinen ein De-facto-Standard geworden ist (Tippmann, 2015).
Im Kontext der beiden behandelten Recommenderalgorithmen stellt R zwei wesentliche Funktionen bereit. Zum einen können durch die Funktion cor(x, [y], [use], [method]) sehr einfach Korrelationen zwischen Vektoren und Matrizen berechnet werden. Ein Beispiel hierfür wäre die Berechnung des Korrelationskoeffizienten nach Bravais-Pearson (Tabelle 3) aus der in Tabelle 1 abgebildeten User-Item-Matrix mit dem Befehl cor(User-Item-Matrix, use= "pairwise.complete.obs"). Zum anderen ist es mit der Funktion plot(x, [y], …) mit überschaubarem Aufwand möglich Diagramme und Graphen zu erstellen und so unter anderem die in Kapitel 3.4.2.2 vorgestellten ROC-Kurven zu zeichnen.

4.5 Interpretation der Ergebnisse

Wie schon erwähnt, ist Gegenstand der Untersuchung die Aufarbeitung der Recommenderalgorithmen im Rahmen eines didaktischen Ansatzes. In Folgenden soll ein Beispiel gegeben werden, wie Ergebnisse einer Simulationsstudie aussehen und wie diese sinnvollerweise interpretiert werden können. Obwohl im Rahmen der Simulationsstudie durchaus belastbare Ergebnisse erzeugt wurden, sind diese hinsichtlich der Intention dieser Untersuchung eher zweitrangig.

4.5.1 Ergebnisse Item-based CF

Die ROC-Kurven für den Item-based CF Algorithmus sind für die unterschiedlichen Ähnlichkeitsmaße in Abbildung 11 dargestellt. In der Legende hinter der jeweiligen Bezeichnung des Ähnlichkeitsmaßes sind die AUC-Werte angegeben.

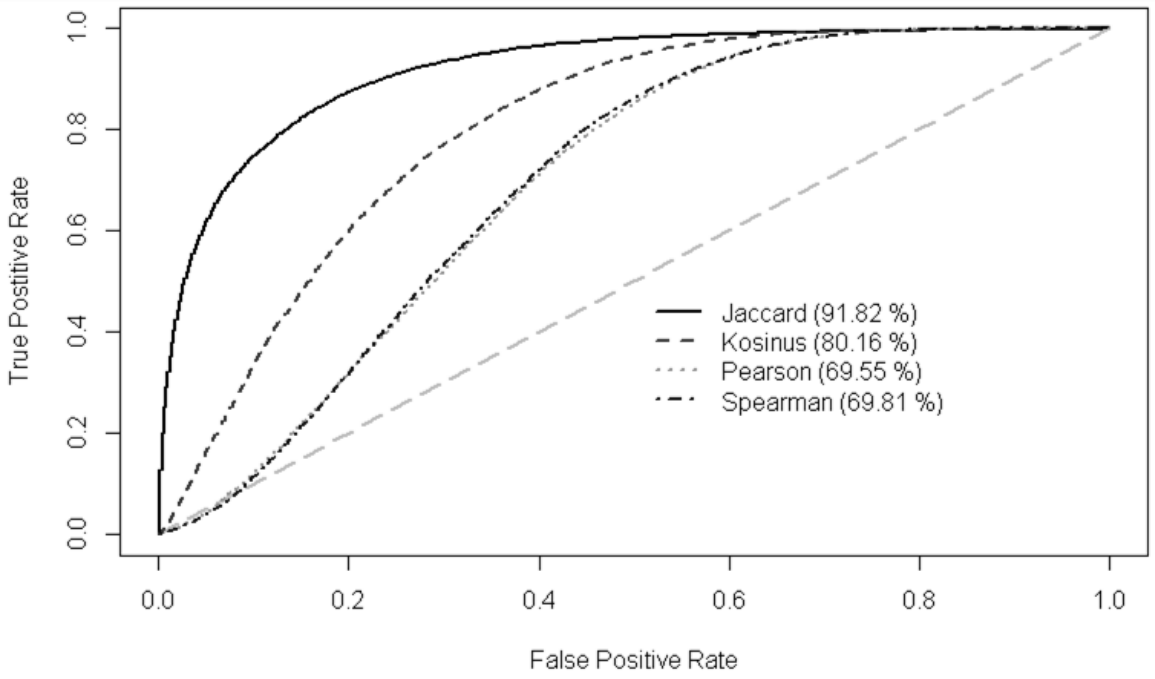

Abbildung 11: ROC-Kurven des Item-based CF Algorithmus

Zunächst lassen sich die vier Ähnlichkeitsmaße anhand der ROC-Kurven in drei Gruppen einteilen:

- Das beste Ergebnis (AUC-Wert 91,82 %) brachte der Einsatz des Jaccard-Koeffizienten. Auch die Form der ROC-Kurve zeigt, dass der Recommenderalgorithmus, dass zahlreiche relevante Items auf Top-10-Rängen empfohlen wurden.
- Im AUC-Wert um ca. 10 % abgeschlagen zeigten die Simulationen unter Verwendung der Kosinus-Ähnlichkeit das zweitbeste Ergebnis. Die Form der ROC-Kurve ist im Verhältnis zu der des Jaccard-Koeffizienten deutlich Flacher, schlägt sich jedoch besser als der grau eingezeichnete Zufallsrecommender.
- Wiederum um ca. 10 % niedriger lagen die Ergebnisse des Korrelationskoeffizienten nach Bravais-Pearson sowie des Rangkorrelationskoeffizienten nach Spearman. Die Form der ROC-Kurve zeigt, dass die beiden Ähnlichkeitsmaße ungefähr auf den ersten 100 Rängen keine nennenswerten Vorteile zu einem zufälligen Recommender bringen.

Unter den Bedingungen dieser Simulation kann bezüglich der Empfehlungsqualität also eine klare Entscheidung zu Gunsten des Jaccard-Koeffizienten getroffen werden.

4.5.2 Ergebnisse User-based CF

Abbildung 12 zeigt die Ergebnisse der Simulationen beim Einsatz des User-based-CF-Algorithmus. Die zugehörigen AUC-Werte sind wieder in Klammern angegeben.

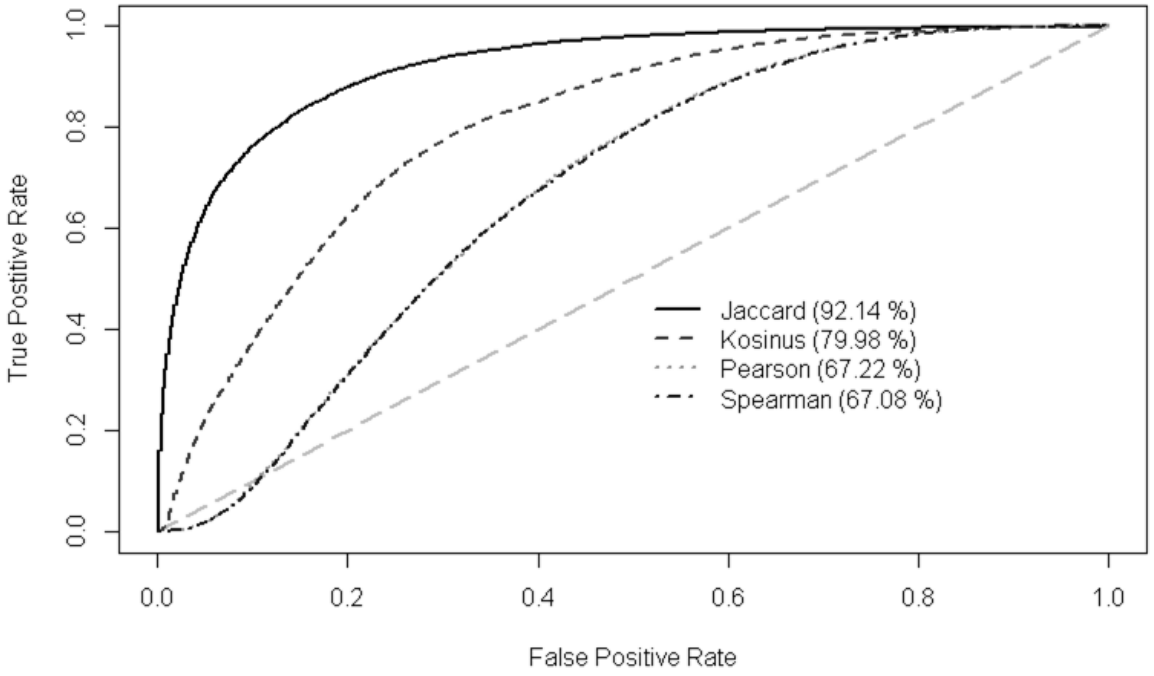

Abbildung 12: ROC-Kurven des User-based CF-Algorithmus

Auf den ersten Blick erscheinen die Ergebnisse sehr ähnlich und können hinsichtlich der Ähnlichkeitsmaße wieder in die gleichen drei Gruppen eingeteilt werden. Bei der Betrachtung der Form der ROC-Kurven fällt auf, dass die Ähnlichkeitsmaße nach Pearson und Spearman auf den ersten Rängen sogar schlechtere Leistungen zeigen, also ein Recommenderalgorithmus der Items auf Zufallsbasis empfiehlt. Die Abstände zwischen den drei Gruppen haben sich auf ca. 12 % vergrößert, das heißt die Streuung hat in den Ergebnissen zugenommen.

4.5.3 Vergleich Item-based und User-based CF

Beim Vergleich von Abbildung 11 und Abbildung 12 fällt zunächst auf, dass die AUC-Werte der beiden Algorithmen mit demselben Ähnlichkeitsmaß relativ nahe zu einander stehen – ganz im Gegensatz zu den AUC-Werten

bei Betrachtung verschiedener Ähnlichkeitsmaße. Dies bedeutet, dass das verwendete Ähnlichkeitsmaß die Empfehlungsqualität deutlich stärker beeinflusst als die Wahl des CF-Algorithmus Algorithmus.
Bei den Simulationen mit dem Jaccard-Koeffizient und der Kosinus-Ähnlichkeit bestehen nur sehr geringe Differenzen in den AUC-Werten. Auch die Form der ROC-Kurven ist sehr ähnlich. Insofern kann in diesen Fällen keine klare Entscheidung für den einen oder anderen Algorithmus erfolgen.
Im Gegensatz dazu liegt der AUC-Wert bei Verwendung des Korrelationskoeffizienten nach Bravais-Pearson und des Rangkorrelationskoeffizienten nach Spearman bei Verwendung des Item-based CF-Algorithmus jeweils um über 2 % höher. Die ROC-Kurven haben eine ähnliche Form, jedoch verlaufen sie beim Item-based CF-Algorithmus etwas steiler. Insofern. Kann bei Verwendung eines der beiden Korrelationskoeffizienten tendenziell eine Entscheidung für den Item-based CF-Algorithmus getroffen werden.

5 Fazit und Ausblick

In diesem Artikel wurde anhand eines durchgängigen Beispiels die Funktionsweise von User-based CF und Item-based CF von der Berechnung der Ähnlichkeitsmaße bis zur Erstellung der Empfehlungsrangliste gezeigt. Darüber hinaus erfolgte die Vorstellung einer Methodik, mit der die Empfehlungsqualität im Rahmen einer Simulationsstudie mit ROC-Kurven evaluiert werden kann. Dadurch wird einem breiteren Publikum der Zugang zum Themenbereich Recomenderalgorithmen vereinfacht.
Die in diesem Beitrag gewählte Form der didaktischen Aufarbeitung lässt sich auch auf hybride state-of-the-art Recommenderalgorithmen mit höherer Komplexität anwenden. In diesem Zusammenhang sind vor allem Verfahren der Tensorfaktorisierung, Faktorisierungsmaschinen, graph-basierte Ansätze und neuronale Netze zu nennen. Gerade hier bieten sich Möglichkeiten für zukünftige Forschungsansätze. Die transparente und verständliche Aufarbeitung dieser Algorithmen kann eine sinnvolle Grundlage für die wissenschaftliche Ausbildung in diesem Bereich liefern.
Des Weiteren hilft eine didaktische Aufarbeitung dieser komplexen Algorithmen die in diesem Kontext gewonnenen wissenschaftlichen Erkenntnisse in die praktische Unternehmenswirklichkeit zu übertragen. Im Zuge dessen kann der Einsatz dieser Algorithmen Unternehmen dabei unterstützen, automatisiert relevante Informationen zu finden und damit Geschäftsprozesse effektiver und effizienter zu gestalten. Ziel sollte dabei sein, heutige state-of-the-art Recommenderalgorithmen als Teil von betrieblicher Standardsoftware zu etablieren.

Literatur

Amazon EU Société à responsabilité limité. (8. März 2015). Abgerufen am 8. März 2015 von http://www.amazon.de/Wirtschaftsinformatik-Eine-Einf%C3%BChrung-Pearson-Studium/dp/3827373484/ref=sr_1_3?s=books&ie=UTF8&qid=1425817332&sr=1-3&keywords=wirtschaftsinformatik

Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendeation Methods. In F. Ricci, R. Lior, B. Shapira, & P. Kantor, *Recommender Systems Handbook* (S. 107-144). New York: Springer.

Fahrmeier, L., Künstler, R., Pigeot, I., & Tutz, G. (1999). *Statistik.* Berlin: Springer-Verlag.

Herlocker, J., Konstan, J., Terveen, L., & Riedl, J. (January 2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems, Vol. 22, No. 1*, S. 5–53.

Karypis, G. (November 2001). Evaluation of item-based top-N recommendation algorithms. *Proceedings of the International Conference on Information and KnowledgeManagement (CIKM '01)* (S. 247–254). Atlanta, Ga, USA: "Evaluation of item-based top-N recommendation.

Klahold, A. (2009). *Methoden von Empfehlungssystemen. Empfehlungssysteme: Recommender Systems— Grundlagen, Konzepte und Lösungen.* Wiesbaden: Vieweg und Teubner.

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE, 2003, 7. Jg., Nr. 1*, (S. 76-80).

Shi, Y., Larson, M., & Hanjalic, A. (1. July 2014). Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. *ACM Computing Surveys (CSUR), Volume 47*, S. Article No. 3.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence, vol. 2009, Article ID 421425*, S. 1-19.

Tippmann, S. (13. Februar 2015). *Programming tools: Adventures with R*. Abgerufen am 23. Juni 2016 von nature International weekly journal of science: http://www.nature.com/news/programming-tools-adventures-with-r-1.16609