Informatik
Hauptcampus

HOCHSCHULE TRIER

# Recommender-Systeme

Recommendation Systems

Fabian Koßmann

Seminar

Betreuer: Prof. Dr. Christoph Schmitz

Trier, 28.02.2026

# Kurzfassung

Recommender-Systeme dienen dazu, aus großen Mengen digitaler Inhalte relevante Empfehlungen für Nutzerinnen und Nutzer abzuleiten. Diese Arbeit stellt die Grundlagen von Recommender-Systemen vor, mit Fokus auf content-basierte Verfahren und kollaboratives Filtern. Darüber hinaus werden ausgewählte Erweiterungen wie Clustering, graphbasierte Ansätze, klassifikationsbasierte Modelle sowie grundlegende Evaluationsmetriken kurz erläutert.

# Inhaltsverzeichnis

# Tabellenverzeichnis

# 1

# Einleitung und Motivation

Recommender-Systeme sind heute allgegenwärtig und prägen die Nutzung zahlreicher Online-Plattformen, etwa durch Produktempfehlungen auf Amazon, Videoempfehlungen auf YouTube oder Inhalte auf sozialen Netzwerken. Empfehlungen helfen Nutzerinnen und Nutzern, sich in der stetig wachsenden Menge verfügbarer Informationen und Produkte zurechtzufinden.
Darüber hinaus beeinflussen Recommender-Systeme zunehmend, wie sich Nutzerinnen und Nutzer informieren und welche Inhalte sie in der digitalen Welt wahrnehmen, wodurch sie auch zur Formung von Meinungen, Weltansichten und mitunter politischen Einstellungen beitragen können.
Angesichts der großen Anzahl an Items, Nutzern und Interaktionen sind manuelle Auswahlmechanismen praktisch nicht mehr umsetzbar, wodurch Recommender-Systeme zunehmend an Bedeutung gewinnen.
Darüber hinaus sind Recommender-Systeme nicht nur wirtschaftlich relevant, sondern auch Gegenstand intensiver Forschung in den Bereichen Information Retrieval, Machine Learning und Data Mining. Während im stationären Handel traditionell eine begrenzte Auswahl besonders stark nachgefragter Produkte präsentiert wurde, ermöglichen digitale Plattformen den Zugriff auf eine deutlich größere Vielfalt an Items. Dieses Phänomen wird häufig als *Long-Tail* bezeichnet [1, S. 321].
Empfehlungssysteme bieten hier den Vorteil, Nutzerinnen und Nutzern gezielt solche Items vorzuschlagen, die ihren individuellen Präferenzen entsprechen [1, S. 321].
In der Praxis funktionieren moderne Empfehlungssysteme teilweise so präzise, dass die Vorschläge als überraschend treffend wahrgenommen werden. Gleichzeitig treten jedoch auch Fehlempfehlungen auf, etwa wenn bereits bekannte, unerwünschte oder thematisch unpassende Items vorgeschlagen werden. Vor diesem Hintergrund ist es besonders interessant, die Funktionsweise und Grenzen von Recommender-Systemen näher zu untersuchen.
Ziel dieser Seminararbeit ist es, die grundlegenden Konzepte von Recommender-Systemen zu vermitteln. Dabei werden klassische content-basierte Empfehlungssysteme sowie das kollaborative Filtern ausführlicher behandelt. Darüber hinaus werden moderne Ansätze, etwa auf Basis von Machine-Learning-Methoden, sowie grundlegende Verfahren zur Bewertung von Empfehlungssystemen kurz vorgestellt.

# 2

# Was ist ein Recommender-System

Nach [1, S. 320] bezeichnen Recommender-Systeme eine Klasse von Systemen, deren Ziel es ist, die Reaktion von Nutzern auf bestimmte Vorschläge vorherzusagen, um möglichst relevante oder nützliche Items zu empfehlen. Grundlage hierfür bilden typischerweise frühere Interaktionen, explizite oder implizite Präferenzen sowie kontextbezogene Informationen der Nutzer.
Darüber hinaus lassen sich Recommender-Systeme gemäß [1, S. 320] grob in zwei Hauptkategorien einteilen, abhängig davon, wie Empfehlungen generiert werden:
**Content-basierte Recommender-Systeme** erzeugen Empfehlungen auf Grundlage der Merkmale (Features) der Items, indem sie diese mit dem individuellen Nutzerprofil abgleichen.
**Collaborative-Filtering-Ansätze** hingegen leiten Empfehlungen aus Ähnlichkeiten zwischen Nutzern oder zwischen Items ab, die sich aus vergangenen Bewertungs- oder Interaktionsdaten ergeben.

# 3

# User-Item-Matrix als Grundmodell

Gemäß [1, S. 320–321] basiert ein zentrales Modell vieler Recommender-Systeme auf der sogenannten *User–Item-Matrix* $UI$. In dieser Matrix repräsentieren die Zeilen einzelne Nutzerinnen und Nutzer, während die Spalten die verfügbaren Items abbilden. Die Matrixeinträge entsprechen den jeweiligen Präferenz- oder Bewertungswerten, die ein Nutzer einem Item zugewiesen hat.
In der Praxis ist die User–Item-Matrix typischerweise sehr dünn besetzt (*sparse*), da nur ein kleiner Teil aller möglichen Nutzer–Item-Kombinationen tatsächlich beobachtet wird. Der überwiegende Anteil der Einträge ist zunächst unbekannt. Ziel eines Recommender-Systems ist es daher, auf Basis der wenigen vorhandenen Präferenzinformationen fehlende Bewertungen vorherzusagen und somit unbekannte Einträge in der Matrix zu schätzen [1, S. 321].
Ein Beispiel für eine User–Item-Matrix, in der verschiedene Leserinnen und Leser Bücher bewertet haben, ist in Tabelle 3.1 dargestellt. Nicht vergebene oder unbekannte Bewertungen sind dabei durch ein ? gekennzeichnet.

| | Der Herr der Ringe | 1984 | Harry Potter 1 | Brave New World |
|---|---|---|---|---|
| Anna | 5 | 4 | 4 | ? |
| Ben | 3 | 5 | ? | 4 |
| Clara | ? | 4 | 5 | 3 |
| David | 4 | ? | 4 | 5 |
| Emma | 2 | 5 | ? | 4 |

Tabelle 3.1: Beispielhafte User-Item-Matrix für Bücher

Das oben genannte Beispiel mit den Nutzern Anna, Ben, Clara, David und Emma sowie den Büchern Der Herr der Ringe, 1984, Harry Potter 1 und Brave New World wird im weiteren Verlauf verwendet, um die unterschiedlichen Konzepte im Rahmen dieser Arbeit über Recommender-Systeme zu erläutern.

# 4

# Nutzerdaten in Recommender-Systemen

Die Verfügbarkeit und Qualität dieser Nutzerdaten ist entscheidend für die Leistungsfähigkeit eines Recommender-Systems, da sie die Grundlage zur Schätzung und Befüllung der User–Item-Matrix (vgl. Kapitel 3) bilden. Insbesondere zu Beginn der Nutzung eines Systems oder beim Hinzukommen neuer Nutzerinnen und Nutzer liegt häufig nur eine geringe Datenbasis vor, was zum sogenannten Kaltstartproblem führt.

## 4.1 Arten von Nutzerdaten

Gemäß [1, S. 323] lassen sich die für Recommender-Systeme verwendeten Nutzerdaten in zwei grundlegende Kategorien einteilen: explizite und implizite Nutzerdaten. Explizite Nutzerdaten entstehen durch bewusste Rückmeldungen der Nutzer, beispielsweise in Form von Bewertungen oder Likes. Implizite Nutzerdaten hingegen werden aus dem tatsächlichen Nutzerverhalten abgeleitet, etwa durch Klicks, Betrachtungsdauer oder Kaufhistorien.

## 4.2 Kaltstartproblem

Das *Kaltstartproblem* bezeichnet die Herausforderung, zu Beginn eines Recommender-Systems sowie beim Hinzukommen neuer Nutzerinnen und Nutzer oder neuer Items sinnvolle Empfehlungen zu generieren, obwohl nur wenige oder gar keine Interaktionsdaten vorliegen. In diesem Fall ist die User–Item-Matrix weitgehend unbesetzt, sodass die Präferenzmodelle der Nutzer nicht zuverlässig geschätzt werden können [2, S. 695–696].
Zur Abschwächung des Kaltstartproblems existieren verschiedene Lösungsansätze. Dazu zählen unter anderem die initiale Erhebung expliziter Nutzerangaben, etwa durch das Bewerten ausgewählter Items, sowie die Nutzung inhaltsbasierter Merkmale oder demographischer Informationen zur Schätzung eines anfänglichen Nutzerprofils. Diese Ansätze sind jedoch häufig mit zusätzlichem Nutzeraufwand verbunden oder können zu ungenauen und potenziell problematischen Annahmen führen [2, S. 696].

# 5

# Ähnlichkeits- und Vergleichsmaße

Für klassische Recommender-Systeme ist der Vergleich von Vektorrepräsentationen zentraler Bedeutung, da Empfehlungen typischerweise auf Ähnlichkeiten zwischen Nutzern, zwischen Items oder zwischen Nutzern und Items beruhen. Im Folgenden werden exemplarisch zwei gebräuchliche Ähnlichkeits- bzw. Distanzmaße vorgestellt und näher erläutert.

## 5.1 Kosinus-Distanz

Die Kosinusähnlichkeit misst die Ähnlichkeit zweier Vektoren anhand des Winkels zwischen ihnen und ist unabhängig von deren euklidischer Länge. Dadurch eignet sie sich sowohl für reellwertige als auch für ganzzahlige Feature-Vektoren und wird häufig in hochdimensionalen Merkmalsräumen eingesetzt [1, S. 327–328].
Exemplarisch wird im Folgenden die Kosinusähnlichkeit zwischen dem Nutzerprofil der Nutzerin Clara

$$UP_{\text{Clara}} = [0, 1, 1, 1, 1, 2, 0.51]$$

aus der User-Feature-Matrix (vgl. Tabelle 6.2) und dem Item-Profil des Buches *Der Herr der Ringe*

$$IP_{\text{Der Herr der Ringe}} = [1, 0, 0, 0, 1, 0, 1.02]$$

aus der Item-Feature-Matrix (vgl. Tabelle 6.1) berechnet.
Die Kosinusähnlichkeit ist definiert als

$$d_{\cos}(UP, IP) = \frac{UP \cdot IP}{\|UP\| \, \|IP\|}.$$

Für die gegebenen Vektoren ergibt sich somit

$$d_{\cos}(UP, IP) = \frac{1.5202}{2.874 \cdot 1.744} \approx 0.303.$$

Ein höherer Wert der Kosinusähnlichkeit deutet auf eine stärkere Übereinstimmung zwischen Nutzerpräferenzen und Item-Merkmalen hin und signalisiert damit eine höhere Relevanz des Items für den jeweiligen Nutzer.

## 5.2 Jaccard-Distanz

Die Jaccard-Distanz ist ein Maß zur Quantifizierung der Unähnlichkeit zweier Mengen und basiert auf dem Verhältnis der Größe ihrer Schnittmenge zur Größe ihrer Vereinigungsmenge. Sie eignet sich insbesondere für binäre oder mengenbasierte Datenrepräsentationen, wie sie beispielsweise bei Wortmengen, Tags oder kategorialen Item-Features auftreten [1, S. 325–326].
Die zugrunde liegende **Jaccard-Ähnlichkeit** ist definiert als

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

wobei $|A \cap B|$ die Anzahl der gemeinsamen Elemente und $|A \cup B|$ die Anzahl aller Elemente bezeichnet, die in mindestens einer der beiden Mengen enthalten sind [1, S. 82].
Aus der Jaccard-Ähnlichkeit lässt sich unmittelbar die **Jaccard-Distanz** ableiten:

$$d_J(A, B) = 1 - J(A, B),$$

wobei größere Distanzwerte auf eine geringere Übereinstimmung der betrachteten Mengen hinweisen [1, S. 106].
Als Beispiel seien zwei Mengen von Schlüsselwörtern betrachtet, die die Inhalte der Buchreihen *Der Herr der Ringe* ($A$) und *Harry Potter* ($B$) beschreiben:

$$A = \{\text{Zauberer}, \text{Schule}, \text{Magie}, \text{Zauberstab}\},$$

$$B = \{\text{Magie}, \text{Schule}, \text{Freundschaft}, \text{Abenteuer}\}.$$

Die zugehörige Schnitt- und Vereinigungsmenge ergeben sich zu

$$A \cap B = \{\text{Magie}, \text{Schule}\},$$

$$A \cup B = \{\text{Zauberer}, \text{Schule}, \text{Magie}, \text{Zauberstab}, \text{Freundschaft}, \text{Abenteuer}\}.$$

Damit berechnet sich die Jaccard-Ähnlichkeit zu

$$J(A, B) = \frac{2}{6} = \frac{1}{3},$$

und entsprechend die Jaccard-Distanz zu

$$d_J(A, B) = 1 - \frac{1}{3} = \frac{2}{3}.$$

Ein höherer Wert der Jaccard-Distanz weist auf eine geringere inhaltliche Übereinstimmung der betrachteten Mengen hin und signalisiert somit eine geringere Ähnlichkeit zwischen den entsprechenden Items [1, S. 326].

# 6

# Content-basierte Empfehlungen

Content-basierte Empfehlungen basieren darauf, Items anhand ihrer inhaltlichen Merkmale zu empfehlen.

## 6.1 Item Profile

Gemäß [1, S. 324–325] werden in content-basierten Recommender-Systemen sogenannte *Item-Profile* verwendet, um relevante Eigenschaften eines Items strukturiert zu erfassen. Ein Item-Profil besteht aus einer Menge von Merkmalen (Features), die das jeweilige Item beschreiben und für die Generierung von Empfehlungen herangezogen werden.
Ein Beispiel für das Item-Profil eines Buches könnte durch einen Feature-Vektor der Form

$$IP_{\text{features}} = [\text{Autor(en)}, \ldots, \text{Genre(s)}, \ldots, \text{Seitenanzahl}]$$

repräsentiert werden. Die konkreten Features hängen dabei von der Domäne sowie von der verfügbaren Metadatenbasis ab.
Die Feature-Vektoren mehrerer Items lassen sich zu einer *Item-Feature-Matrix* $IP$ zusammenfassen, in der jede Zeile einem Item und jede Spalte einem bestimmten Merkmal entspricht [1, S. 325]. Für das Item *Buch* könnte eine solche Matrix beispielhaft wie folgt aussehen:

| | Tolkien | Orwell | Rowling | Huxley | Fantasy | Dystopie | Seitenzahl (norm.) |
|---|---|---|---|---|---|---|---|
| Der Herr der Ringe | 1 | 0 | 0 | 0 | 1 | 0 | 1.02 |
| 1984 | 0 | 1 | 0 | 0 | 0 | 1 | 0.61 |
| Harry Potter 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0.46 |
| Brave New World | 0 | 0 | 0 | 1 | 0 | 1 | 0.45 |

Tabelle 6.1: Beispielhafte Item-Feature-Matrix für Bücher

### Feature-Extraktion

Eine zentrale Herausforderung content-basierter Recommender-Systeme besteht in der Extraktion geeigneter Item-Features, insbesondere bei nicht-atomaren Items

wie Textdokumenten, Bildern oder Videos, deren Inhalte nicht unmittelbar in strukturierter Form vorliegen [1, S. 325]. In solchen Fällen müssen die relevanten Merkmale zunächst aus den Rohdaten abgeleitet werden.
Für Textdokumente stellt der sogenannte *TF-IDF-Score* (*Term Frequency–Inverse Document Frequency*) ein gängiges Verfahren zur Feature-Extraktion dar. Dieser misst die Relevanz eines Begriffs für ein bestimmtes Dokument im Verhältnis zum gesamten Dokumentenkorpus. Begriffe mit hohen TF-IDF-Werten sind charakteristisch für ein Dokument und können daher als aussagekräftige Features zur Repräsentation des Dokuments herangezogen werden [1, S. 325–326].
Eine allgemeinere Methode zur Feature-Extraktion, die sich auch auf andere Medientypen wie Bilder oder Videos übertragen lässt, ist das sogenannte *Tagging*. Dabei werden Items mit beschreibenden Schlagwörtern (Tags) annotiert, die als semantische Merkmale zur Inhaltsbeschreibung dienen. Diese Tags können entweder manuell durch Nutzer oder automatisch durch entsprechende Algorithmen erzeugt werden und ermöglichen eine domänenübergreifende Repräsentation von Inhalten [1, S. 326–327].

### Repräsentation von Item-Features

In content-basierten Recommender-Systemen werden Items durch Feature-Vektoren repräsentiert, die relevante Eigenschaften des jeweiligen Items beschreiben [1, S. 324]. Diskrete Merkmale wie Genres, Schauspieler oder Regisseure lassen sich dabei häufig als binäre Features modellieren, wobei jede Vektorkomponente das Vorhandensein oder Fehlen eines Merkmals kennzeichnet.
Numerische Merkmale, etwa durchschnittliche Bewertungen oder Seitenzahlen, werden hingegen als reellwertige Komponenten im Feature-Vektor abgebildet, da eine binäre Darstellung die in den Zahlen enthaltene Struktur verlieren würde [1, S. 325]. In der Praxis können Item-Profile sowohl binäre als auch numerische Features enthalten. Bei der Berechnung von Ähnlichkeiten ist daher eine geeignete Skalierung numerischer Merkmale erforderlich, um deren Einfluss auf das Ergebnis angemessen zu steuern [1, S. 326].

## 6.2 User Profile

Nutzerprofile beschreiben, welche Item-Features ein Nutzer bevorzugt, und stellen eine zentrale Repräsentation in content-basierten Recommender-Systemen dar. Sie werden aus den vorhandenen Einträgen der User–Item-Matrix konstruiert, indem die vom jeweiligen Nutzer bewerteten Items berücksichtigt und deren Merkmalsvektoren aggregiert werden [1, S. 328–329].
Die Aggregation kann dabei optional durch die zugehörigen Nutzerbewertungen gewichtet werden, sodass stärker bewertete Items einen größeren Einfluss auf das resultierende Profil haben. Das resultierende Nutzerprofil enthält für jedes Feature eine aggregierte Präferenzbewertung.
Ausgehend von der in Tabelle 6.1 dargestellten Item-Feature-Matrix kann die Ag-

gregation der Nutzerpräferenzen beispielhaft wie folgt erfolgen: Kategoriale Merkmale wie Autor oder Genre werden typischerweise über die Häufigkeit der vom Nutzer konsumierten Items aggregiert, während numerische Merkmale, etwa die Seitenanzahl eines Buches, über statistische Kennzahlen wie den Mittelwert zusammengefasst werden.
Für die Nutzerin Clara lässt sich auf Basis der User–Item-Matrix in Tabelle 3.1 ein beispielhaftes Nutzerprofil konstruieren. Clara hat die Bücher *1984*, *Harry Potter 1* und *Brave New World* bewertet, was sich aus der entsprechenden Zeile der User–Item-Matrix

$$UI_{\text{Clara}} = [\,?, 4, 5, 3\,]$$

ablesen lässt.
Gemäß der Item-Feature-Matrix besitzen diese Bücher die folgenden Feature-Vektoren:

$$IP_{1984} = [0, 1, 0, 0, 0, 1, 0.61],$$

$$IP_{\text{Harry Potter 1}} = [0, 0, 1, 0, 1, 0, 0.46],$$

$$IP_{\text{Brave New World}} = [0, 0, 0, 1, 0, 1, 0.45].$$

Durch Aggregation dieser Item-Profile ergibt sich das Nutzerprofil von Clara:

$$UP_{\text{Clara}} = [0, 1, 1, 1, 1, 2, 0.51].$$

Die Nutzerprofile aller Nutzer lassen sich analog zu einer *User-Feature-Matrix UP* zusammenfassen, in der jede Zeile einem Nutzer und jede Spalte einem Feature entspricht [1, S. 325]. Eine beispielhafte Darstellung dieser Matrix ist nachfolgend angegeben.

| | Tolkien | Orwell | Rowling | Huxley | Fantasy | Dystopie | Seitenzahl (norm.) |
|---|---|---|---|---|---|---|---|
| Anna | 1.0 | 1.0 | 1.0 | 0.0 | 2.0 | 1.0 | 0.70 |
| Ben | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.69 |
| Clara | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 | 0.51 |
| David | 1.0 | 0.0 | 1.0 | 1.0 | 2.0 | 1.0 | 0.64 |
| Emma | 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 2.0 | 0.69 |

Tabelle 6.2: Beispielhafte User-Feature-Matrix für Bücher

## 6.3 Klassische Content-basierte Empfehlungen

Bei content-basierten Empfehlungssystemen wird die Relevanz eines Items für einen Nutzer auf Grundlage der Ähnlichkeit zwischen dem Nutzerprofil und dem Itemprofil bestimmt. Das Nutzerprofil beschreibt die Präferenzen eines Nutzers in Bezug auf einzelne Features, während das Itemprofil die Merkmalsausprägungen des jeweiligen Items repräsentiert.
Das Nutzerprofil der Nutzerin Anna sei beispielsweise gegeben durch

$$UP_{\text{Anna}} = [1, 1, 1, 0, 2, 1, 0.70].$$

Zur Abschätzung der Relevanz des Buches *Brave New World* wird dessen Itemprofil

$$IP_{\text{Brave New World}} = [0, 0, 0, 1, 0, 1, 0.45]$$

herangezogen und die Ähnlichkeit zwischen Nutzer- und Itemprofil, exemplarisch mithilfe der Kosinus-Ähnlichkeit, berechnet:

$$d_{\cos}\big(UP_{\text{Anna}}, IP_{\text{Brave New World}}\big) = \frac{1.315}{2.914 \cdot 1.484} \approx 0.30.$$

Der berechnete Ähnlichkeitswert stellt dabei keinen direkten Rating-Eintrag in der User–Item-Matrix dar, sondern dient als Relevanzmaß zur Rangordnung potenzieller Empfehlungen für den Nutzer.

# 7

# Kollaboratives Filtern

Beim kollaborativen Filtern werden Nutzer oder Items anhand ihrer Interaktionen als ähnlich identifiziert, um auf dieser Basis bislang unbekannte Items zu empfehlen, die von ähnlichen Nutzern positiv bewertet wurden bzw. ähnliche Items zu bereits bekannten Präferenzen darstellen.

## 7.1 Klassisches kollaboratives Filtern

Beim klassischen kollaborativen Filtern werden Empfehlungen auf Grundlage von Ähnlichkeiten zwischen Nutzern oder zwischen Items erzeugt. Dazu werden die Zeilen bzw. Spalten der User–Item-Matrix als Vektoren interpretiert und mithilfe geeigneter Ähnlichkeits- oder Distanzmaße miteinander verglichen.
Exemplarisch seien die Nutzerprofile von Anna und Ben als Vektoren der User–Item-Matrix gegeben. Ein höherer Wert der Kosinus-Ähnlichkeit weist dabei auf eine stärkere Übereinstimmung der Präferenzen der beiden Nutzer hin.
Zur Vorhersage unbekannter Bewertungen werden typischerweise die $k$ ähnlichsten Nutzer herangezogen. Die Bewertung eines Nutzers für ein bislang nicht bewertetes Item wird dabei als Mittelwert der entsprechenden Bewertungen dieser ähnlichsten Nutzer geschätzt [1, S. 333].
Dieses Vorgehen lässt sich analog auf Items übertragen. In diesem Fall werden die $k$ ähnlichsten Items bestimmt und deren Bewertungen aggregiert, um die Bewertung für ein noch nicht bewertetes Item zu schätzen [1, S. 333].
Es soll geschätzt werden, wie die Nutzerin Anna das Buch *Brave New World* bewertet, das sie bislang nicht bewertet hat. Auf Grundlage der User-Item-Matrix in Tabelle 3.1 ergeben sich die folgenden Kosinus-Distanzen zu Anna:

$$d_{\cos}(\text{Anna}, \text{Ben}) \approx 0.937,\ d_{\cos}(\text{Anna}, \text{Clara}) \approx 0.994,$$

$$d_{\cos}(\text{Anna}, \text{David}) \approx 0.994,\ d_{\cos}(\text{Anna}, \text{Emma}) \approx 0.87.$$

Damit sind Clara und David die beiden Nutzer mit der höchsten Ähnlichkeit zu Anna. Da deren Bewertungen für *Brave New World* gegeben sind durch

$$r_{\text{Clara,Brave New World}} = 3 \quad \text{und} \quad r_{\text{David,Brave New World}} = 5,$$

kann der unbekannte Eintrag als Mittelwert geschätzt werden:

$$r_{\text{Anna,Brave New World}} = \frac{3+5}{2} = 4.$$

Beim item-basierten kollaborativen Filtern werden analog nicht Nutzerprofile, sondern die Spalten der User–Item-Matrix miteinander verglichen, um Ähnlichkeiten zwischen Items zu bestimmen. Einem Nutzer werden anschließend solche Items empfohlen, die den von ihm bereits positiv bewerteten Items besonders ähnlich sind.

## 7.2 User-Item-Matrix und Binärisierung

Um eine robustere Berechnung der Ähnlichkeit zwischen Nutzern zu ermöglichen, können numerische Bewertungen binarisiert werden [1, S. 335]. Dabei werden Bewertungen oberhalb eines festgelegten Schwellwerts als positive Interaktionen interpretiert, während niedrigere Bewertungen sowie fehlende Einträge als nicht positive bzw. nicht informative Interaktionen behandelt werden.
In diesem Beispiel gelten Bewertungen von 3 bis 5 als positiv.
Formal ergibt sich daraus eine binäre Präferenzmatrix $UI_B$, die wie folgt definiert ist:

$$(UI_B)_{u,i} = \begin{cases} 1, & \text{falls } r_{u,i} \geq 3, \\ 0, & \text{sonst.} \end{cases}$$

Auf Grundlage der in Tabelle 3.1 dargestellten User-Item-Matrix ergibt sich die in Tabelle 7.1 dargestellte binärisierte Matrix. Diese enthält ausschließlich Informationen darüber, ob ein Nutzer ein bestimmtes Item positiv bewertet hat, unabhängig von der genauen Bewertungshöhe.

| | Der Herr der Ringe | 1984 | Harry Potter 1 | Brave New World |
|---|---|---|---|---|
| Anna | 1 | 1 | 1 | 0 |
| Ben | 1 | 1 | 0 | 1 |
| Clara | 0 | 1 | 1 | 1 |
| David | 1 | 0 | 1 | 1 |
| Emma | 0 | 1 | 0 | 1 |

Tabelle 7.1: Beispielhafte binärisierte User-Item-Matrix für Bücher

### 7.2.1 Rating Normalisierung

Zur Reduktion individueller Bewertungsstile können Ratings normalisiert werden, indem von jeder Bewertung der durchschnittliche Bewertungswert des jeweiligen Nutzers subtrahiert wird.
Dadurch werden Bewertungen relativ zum persönlichen Mittelwert interpretiert,

sodass positive und negative Abweichungen entstehen. Die anschließende Anwendung der Cosinus-Ähnlichkeit ermöglicht es, Nutzer mit ähnlichen Präferenzmustern anhand kleiner Winkel zwischen ihren Bewertungsvektoren zu identifizieren, während gegensätzliche Meinungen durch nahezu entgegengesetzte Vektoren repräsentiert werden [1, S. 335].
Die in Tabelle 7.2 dargestellte normalisierte Matrix wird durch Mittelwert-Zentrierung der User-Item-Matrix aus Tabelle 3.1 erzeugt.

| | Der Herr der Ringe | 1984 | Harry Potter 1 | Brave New World |
|---|---|---|---|---|
| Anna | 0.67 | -0.33 | -0.33 | ? |
| Ben | -1.00 | 1.00 | ? | 0.00 |
| Clara | ? | 0.00 | 1.00 | -1.00 |
| David | -0.33 | ? | -0.33 | 0.67 |
| Emma | -1.67 | 1.33 | ? | 0.33 |

Tabelle 7.2: Normalisierte User-Item-Matrix (Mittelwert-Zentrierung pro Nutzer)

## 7.3 Clustern von Nutzern und Items

Da die User–Item-Matrix, wie in Kapitel 3 beschrieben, typischerweise stark dünn besetzt ist, lassen sich Ähnlichkeiten zwischen einzelnen Nutzern oder Items häufig nur eingeschränkt direkt bestimmen.
Ein gängiger Ansatz zur Minderung dieses Problems ist das Clustering von Nutzern und/oder Items [1, S. 337–338].
Dabei können zunächst Items zu Item-Clustern zusammengefasst werden. Die Bewertung eines Nutzers für ein solches Item-Cluster wird als der Mittelwert der Bewertungen definiert, die der Nutzer den zugehörigen Items vergeben hat [1, S. 338].
Analog dazu lassen sich Nutzer zu Nutzer-Clustern gruppieren, wobei die Bewertung eines Nutzer-Clusters für ein Item oder Item-Cluster als durchschnittliche Bewertung aller Nutzer innerhalb des jeweiligen Clusters bestimmt wird [1, S. 338].
Auf diese Weise entsteht eine verdichtete, geclusterte User–Item-Matrix, welche die ursprüngliche Matrix approximiert.
Dieses Vorgehen reduziert die Sparsity der Daten und ermöglicht stabilere sowie robustere Schätzungen von Nutzerpräferenzen und Empfehlungen.
Auf Basis der User–Item-Matrix in Tabelle 3.1 werden die Items *Der Herr der Ringe* und *Harry Potter 1* aufgrund ähnlicher Bewertungsmuster zu einem Item-Cluster $C_1$ zusammengefasst, während die Items *1984* und *Brave New World* unverändert als einzelne Items bestehen bleiben. Die Bewertung eines Nutzers für das Item-Cluster $C_1$ ergibt sich als Mittelwert der zugehörigen Einzelbewertungen. Für die Nutzerin Anna gilt somit

$$r_{\text{Anna},C_1} = \frac{5+4}{2} = 4.5,$$

während sich für die übrigen Nutzer die Cluster-Bewertungen

$$r_{\text{Ben},C_1} = 3,\ r_{\text{Clara},C_1} = 5,\ r_{\text{David},C_1} = 4,\ r_{\text{Emma},C_1} = 2$$

ergeben.
Werden die Nutzer Ben und Emma zu einem Nutzer-Cluster $U_1$ zusammengefasst, so ergibt sich die Bewertung dieses Clusters für ein Item oder Item-Cluster als Mittelwert der entsprechenden Bewertungen der zugehörigen Nutzer. Für das Item-Cluster $C_1$ folgt

$$r_{U_1,C_1} = \frac{3+2}{2} = 2.5,$$

während sich für die einzelnen Items

$$r_{U_1,1984} = 5 \quad \text{und} \quad r_{U_1,\text{Brave New World}} = 4$$

ergeben.

| | $C_1$ (Der Herr der Ringe , Harry Potter 1) | 1984 | Brave New World |
|---|---|---|---|
| Anna | 4.5 | 4.0 | ? |
| Clara | 5.0 | 4.0 | 3.0 |
| David | 4.0 | ? | 5.0 |
| $U_1$ (Ben, Emma) | 2.5 | 5.0 | 4.0 |

Tabelle 7.3: Teilweise geclusterte User–Item-Matrix

## 7.4 Graphenbasierte Erweiterung

Ein alternativer Ansatz zur Erweiterung der User–Item-Matrix sind graphbasierte Verfahren. Dabei werden Nutzer und Items als Knoten eines Graphen modelliert, wobei Kanten Interaktionen oder Ähnlichkeiten zwischen ihnen repräsentieren. Im Gegensatz zu klassischen nachbarschaftsbasierten Methoden berücksichtigen graphbasierte Ansätze auch indirekte Beziehungen, indem Präferenzinformationen entlang der Kanten des Graphen propagiert werden. Der Einfluss eines Knotens auf einen anderen nimmt dabei mit zunehmender graphischer Distanz ab, was als Propagation und Abschwächung bezeichnet wird [2, S. 136].

# 8

# Klassifikationsalgorithmen

In den letzten Jahren kommen im Bereich der Recommender-Systeme zunehmend klassifikationsbasierte Ansätze zum Einsatz [1, S. 330].
Dabei wird die Vorhersage, wie gut ein Nutzer ein Item bewertet oder mag, als Klassifikationsproblem modelliert.
Mithilfe eines Machine-Learning-Modells wird vorhergesagt, welcher Klasse eine Nutzer–Item-Interaktion zugeordnet werden kann, beispielsweise *positiv* oder *negativ*.
Die möglichen Bewertungsausprägungen bzw. Scores werden dabei als Klassen interpretiert [1, S. 330].
Für diese Aufgabe können eine Vielzahl unterschiedlicher Machine-Learning-Algorithmen eingesetzt werden, darunter Entscheidungsbäume, neuronale Netze oder andere überwachte Lernverfahren. Die Leistungsfähigkeit solcher Systeme hängt maßgeblich von der Qualität und Repräsentation der verwendeten Features ab, die sowohl aus Nutzerdaten als auch aus Item-Eigenschaften abgeleitet werden können, ähnlich den Nutzerprofilen in content-basierten Empfehlungssystemen.
Auf Basis dieser Feature-Repräsentationen lernt das Modell implizit, welche Eigenschaften von Items ein Nutzer bevorzugt oder ablehnt, und nutzt dieses Wissen, um Vorhersagen für bislang unbekannte Nutzer–Item-Paare zu treffen.

# 9

# Metriken

Zur Bewertung und zum Vergleich der Leistungsfähigkeit unterschiedlicher Recommender-Systeme werden Evaluationsmetriken eingesetzt. Diese ermöglichen eine quantitative Beurteilung der Qualität erzeugter Empfehlungen anhand definierter Referenzgrößen. Im Folgenden werden exemplarisch zwei häufig verwendete Top-$k$-basierte Evaluationsmetriken vorgestellt, nämlich **Precision@k** und **Recall@k** [3, S. 62].
Sei $U$ ein Nutzer, $R_U$ die Menge der vom Recommender-System für diesen Nutzer generierten Empfehlungen in Form einer Top-$k$-Liste und $G_U$ die Menge der für den Nutzer tatsächlich relevanten Items (Ground Truth). Die Ground Truth $G_U$ wird üblicherweise aus historischen Nutzerinteraktionen wie Bewertungen, Klicks oder Käufen abgeleitet und dient als Referenz zur Bewertung der Empfehlungsqualität [3, S. 61].

## 9.1 Precision@k

Die Metrik *Precision@k* misst den Anteil relevanter Items innerhalb der Top-$k$ empfohlenen Items eines Nutzers. Sie ist definiert als

$$\text{Precision@k}(U) = \frac{|R_U \cap G_U|}{k}.$$

Ein hoher Wert der Precision@k weist darauf hin, dass ein großer Anteil der empfohlenen Items für den Nutzer relevant ist und die Empfehlungen somit eine hohe Präzision aufweisen [3][S. 62].
Zur Veranschaulichung der Metrik Precision@k wird die Nutzerin Anna betrachtet. Angenommen, ein Recommender-System empfiehlt Anna die folgenden Top-$k$ Items mit $k = 2$:

$$R_{\text{Anna}} = \{\textit{Brave New World}, \textit{1984}\}.$$

Die Ground-Truth-Menge der für Anna relevanten Items sei gegeben durch:

$$G_{\text{Anna}} = \{\textit{Brave New World}\}.$$

Die Schnittmenge der empfohlenen und relevanten Items ergibt sich somit zu:

$$R_{\text{Anna}} \cap G_{\text{Anna}} = \{\textit{Brave New World}\}.$$

Damit berechnet sich die Precision@2 für Anna als:

$$\text{Precision@2(Anna)} = \frac{|R_{\text{Anna}} \cap G_{\text{Anna}}|}{2} = \frac{1}{2} = 0.5.$$

## 9.2 Recall@k

Der *Recall@k* misst den Anteil der tatsächlich relevanten Items, die durch die Top-$k$ Empfehlungen eines Nutzers abgedeckt werden. Er ist definiert als

$$\text{Recall@k}(U) = \frac{|R_U \cap G_U|}{|G_U|}.$$

Diese Metrik beschreibt, wie vollständig die Menge relevanter Items durch die Empfehlungen erfasst wird, und bewertet somit die Abdeckung der relevanten Inhalte [3][S. 62].
Zur Illustration der Metrik Recall@k wird erneut die Nutzerin Anna betrachtet. Angenommen, ein Recommender-System empfiehlt Anna die folgenden Top-$k$ Items mit $k = 2$:

$$R_{\text{Anna}} = \{\textit{Brave New World}, \textit{1984}\}.$$

Die Ground-Truth-Menge der für Anna tatsächlich relevanten Items sei gegeben durch:

$$G_{\text{Anna}} = \{\textit{Brave New World}\}.$$

Die Schnittmenge der empfohlenen und relevanten Items ergibt sich zu:

$$R_{\text{Anna}} \cap G_{\text{Anna}} = \{\textit{Brave New World}\}.$$

Damit berechnet sich der Recall@2 für Anna als:

$$\text{Recall@2(Anna)} = \frac{|R_{\text{Anna}} \cap G_{\text{Anna}}|}{|G_{\text{Anna}}|} = \frac{1}{1} = 1.0.$$

# 10

# Zusammenfassung und Ausblick

In dieser Arbeit wurden zentrale Grundlagen von Recommender-Systemen dargestellt. Zunächst erhielten klassische Verfahren wie content-basierte Empfehlungen und kollaboratives Filtern eine ausführliche Betrachtung. Darauf aufbauend wurden moderne Erweiterungen skizziert, etwa graphbasierte Ansätze, klassifikationsbasierte Modelle sowie grundlegende Metriken zur Bewertung von Empfehlungssystemen.
Die Betrachtung zeigt, dass das Thema weiterhin an Relevanz gewinnt, da die Menge an verfügbaren Daten und Items stetig steigt und brauchbare Empfehlungen immer stärker gefiltert werden müssen. Recommender-Systeme sind inzwischen ein wesentlicher Bestandteil digitaler Plattformen und E-Commerce-Anwendungen, was auch in einflussreicher Fachliteratur betont wird [1, S. 321–323].
Über die technische Funktionsweise hinaus besitzen Recommender-Systeme auch eine gesellschaftliche Bedeutung: Sie strukturieren die Auswahl und Sichtbarkeit digitaler Inhalte und wirken sich damit auf Informationszugang und Meinungsbildungsprozesse aus.
Diese gesamtheitlichen Implikationen finden in interdisziplinären Forschungsfeldern Aufmerksamkeit.
Für weiterführende Arbeiten bieten sich mehrere Richtungen an:

- Tiefere Analyse aktueller Machine-Learning- und Deep-Learning-Methoden sowie deren Vergleich mit klassischeren Ansätzen hinsichtlich Robustheit, Transparenz und Fairness.
- Vertiefte Untersuchung gesellschaftlicher Wirkungen, etwa hinsichtlich Informationsvielfalt, Filtereffekten oder politischer Meinungsbildung.
- Entwicklung und Anwendung umfassender Bewertungs- und Qualitätsmetriken, die technische Leistung, Nutzungsakzeptanz und gesellschaftliche Effekte zugleich berücksichtigen.

Damit schließt die Arbeit an aktuelle Diskussionen an und zeigt zugleich Potenzial für weiterführende Forschung und Reflexion in Technik, Gesellschaft und Bildung.

## Literatur

[1] J. Leskovec, A. Rajaraman und J. D. Ullman, *Mining of Massive Datasets*, 3. Aufl. Cambridge University Press, 2020.

[2] F. Ricci, L. Rokach und B. Shapira, Hrsg., *Recommender Systems Handbook*. Springer, 2011, ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3.

[3] A. Felfernig, L. Boratto, M. Stettinger und M. Tkalčič, "Evaluating Group Recommender Systems," in *Group Recommender Systems*, Springer, 2018. DOI: 10.1007/978-3-319-75067-5_3.

# A

# Eigenständigkeitserklärung

☐ Die vorliegende Arbeit wurde als Einzelarbeit angefertigt.

☐ Die vorliegende Arbeit wurde als Gruppenarbeit angefertigt. Mein Anteil an der Gruppenarbeit ist im untenstehenden Abschnitt *Verantwortliche* dokumentiert:

☐ Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne unzulässige Hilfe Dritter angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche kenntlich gemacht. Darüber hinaus erkläre ich, dass ich die vorliegende Arbeit in dieser oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht habe.

☐ Es ist keine Nutzung von KI-basierten text- oder inhaltgenerierenden Hilfsmitteln erfolgt.

☐ Die Nutzung von KI-basierten text- oder inhaltgenerierenden Hilfsmitteln wurde von der/dem Prüfenden ausdrücklich gestattet. Die von der/dem Prüfenden mit Ausgabe der Arbeit vorgegebenen Anforderungen zur Dokumentation und Kennzeichnung habe ich erhalten und eingehalten. Sofern gefordert, habe ich in der untenstehenden Tabelle *Nutzung von KI-Tools* die verwendeten KI-basierten text- oder inhaltgenerierenden Hilfsmittel aufgeführt und die Stellen in der Arbeit genannt. Die Richtigkeit übernommener KI-Aussagen und Inhalte habe ich nach bestem Wissen und Gewissen überprüft.

Datum

Unterschrift der Kandidatin/des Kandidaten

## Nutzung von KI-Tools

| **KI-Tool** | **Genutzt für** | **Warum?** | **Wann?** | **Mit welcher Eingabefrage bzw. -aufforderung?** | **An welcher Stelle der Arbeit übernommen?** |
|---|---|---|---|---|---|
| ChatGPT | Formulierung zu verbessern | besseres Verständniss der Arbeit | Der gesammten Arbeit. | Formuliere folgendes besser: ... | - |
| ChatGPT | Rechtschreibung zu verbessern | Einhalten der Formalen deutschen Schreibweise | Korrigiere folgendes auf Rechtschreibung: ... | Der gesammten Arbeit. | - |
| ChatGPT | Erstellen von Tabellen (danach geprüft) | Das die Tabelle möglichst korrekt ist | Erstelle folgende Tabelle: ... | Der gesammten Arbeit. | - |
| ChatGPT | Erstellen von Formel (danach geprüft) | Das die Formel korrekt ist in Latex | Erstelle folgende Formel: ... | Der gesammten Arbeit. | - |
| ChatGPT | Zusammenfassen von Quellen (danach geprüft) | Das Quellen schneller verständlich sind (anschließend geprüft) | Fasse folgendes kurz zusammen: ... | Der gesammten Arbeit. | - |