Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw

Information Engineering 1: Information Retrieval

Kategorisierung/Recommender mittels Information Retrieval

Kapitel 5

Martin Braschler

Agenda

- Recommender basierend auf Kategorisierung/Klassifikation
- Content-based Verfahren
  - Rocchio
  - kNN
  - Bayes
- Collaborative Filtering
- Thesauri
- Klassifikationen
- Social Tagging

Material u.a. von Prof. H.-P. Frei, Ellery Smith

Dokumentkategorisierung

- Definition: Bei der Kategorisierung werden Dokumente anhand ihres Inhaltes einer speziellen Kategorie (im Allg. Teil einer Informationsstruktur) zugewiesen.
- Eine Kategorie sollte innerhalb einer bestimmten Domäne einen wohl definierten Bereich darstellen. Mit anderen Worten: Es geht um den Umgang mit Bedeutung, obwohl wir in den meisten Fällen nur "rohe Daten" zur Verfügung haben

- Abgrenzung: wir betrachten Methoden, die Kategorisierung als Suchproblem behandeln, und für die wir die besprochenen Ansätze (Vektorraummodell etc.) adaptieren können

Kategorisierung Aufgaben

- Es gibt eine Menge unterschiedlicher Aufgaben:
- Indexierung (Verschlagwortung)
  - **Manuelle Methoden** der Indexierung sind für Online-Kollektionen schwerfällig und kostenintensiv. Die Frage ist, wie man die menschliche Indexierung semi-automatisieren kann. Das kontrollierte Vokabular der Indexierungssprache bildet die Kategorien.
- Recommender (klassisch: Routing/Filtering)
  - Die einkommenden Dokumente werden regelmässig gegen ein **Userprofil resp. Themenprofil** getestet, um zu bestimmen, wo erstere einzuordnen sind (Dokumenten-Feed). Verwandt: Push-Dienste

Kategorisierung Aufgaben (cont.)

- Clustering
  - Gruppieren von Kollektionen (z.B. Memos, E-Mails) in eine Menge von sich gegenseitig ausschliessenden Kategorien – die aus den Daten gebildet werden. Schwierig wenn Dokumente kurz sind. → Es gibt Ausreisser!
- Annotation
  - Gruppieren von Dokumenten (z.B. wissenschaftliche Artikel) mit weiterführender, dazugehöriger Information.

Unterschied Kategorisierung/Klassifikation

- Kennen Sie den Unterschied Kategorisierung/Klassifikation?

Unterschied Kategorisierung / Klassifikation

- Eine „Mitgliedschaft“ in einer **Kategorie** ist nicht exklusiv, die Grenzen können verschwimmen/-überlappen → In diesem Sinne sind nicht notwendigerweise alle einer Kategorie zugeordneten Informationseinheiten gleich gute Repräsentanten für die Kategorie. Ein Objekt/Dokument kann zu mehreren Kategorien gehören
- eine **Klassifikation** besteht aus starren, exakt disjunkten Klassen, die hierarchisch angeordnet sind.
- Auch in der Literatur ständig falsch verwendet ☹

Content-based vs. Collaborative

Wir unterscheiden grundsätzlich:

1. «Content-based» Verfahren: die Kategorisierung erfolgt aufgrund des «Inhalts» (resp. der Beschreibung) des Objektes (welches deshalb vollständig digital vorliegen muss)
2. «Collaborative»-Verfahren: die Kategorisierung erfolgt aufgrund von externen Signalen, wie z.B. Bewertungen oder Clicks (es reicht daher ein digitales Surrogat)

(Content-Based)-Verfahren mit Wurzeln im Information Retrieval

- Rocchio Model
- kNN, Nearest Neighbor Algorithmus
- Bayes Klassifizierung

Skizze Rocchio

Rocchio Model

- Rocchio modelliert ein Kategorie C mittels einem Kategorie-Repräsentanten c. Der Repräsentant c = (c_1,….c_n) ist ein Vektor und wird konstant aktualisiert.
- Die Komponenten vom (neuen) c sind:

$$c_j = \alpha c'_j + \beta \frac{1}{n_c} \sum_{D \in C} d_j - \gamma \frac{1}{n - n_c} \sum_{D \notin C} d_j$$

- Wobei:
  - D: Dokumente der Kollektion (d_j = einzelnes Dokument)
  - n: Totale Anzahl von Dokumenten in der Kollektion
  - n_c: Anzahl von Dokumenten in Kategorie C
  - α, β, γ: kontrolliert den relativen Einfluss der drei Gewichtungskomponenten

Rocchio Algorithmus

- Der Kategorie-Repräsentant c ist wie ein normales Dokument mit dem Unterschied, dass er nicht real existiert („hypothetisch“). Er wird z.B. initial aus positiven Beispielen generiert.
- Die Idee besteht darin, den Repräsentanten in Richtung der positiven Beispiele und weg von den negativen Beispielen zu bewegen. → kommt Ihnen das bekannt vor?
- Rocchio‘s Kategorisierung
  - Neue Dokumente werden als Anfrage zum Vergleich mit den Repräsentanten verwendet (z.B. Winkel berechen, Vektormodell)
  - Falls s(D,C) > delta wird das Dokument D der Kategorie C zugeteilt. delta ist ein Grenzwert, der geeignet bestimmt werden muss.

Bewertung Rocchio Methode

- Dieser Algorithmus
  - ist einfach zu implementieren und sehr effizient (→ wieso?). Er wird vielfach als Grundlage in Kategorisierungs-Experimenten gebraucht.
  - ein gravierender Nachteil ist, dass er nicht robust ist (vor allem wenn die Anzahl von negativen Instanzen gross wird).
  - die Festlegung der Parameter ist **knifflig** und hängt stark von der Art und Grösse der Kollektion ab.
  - hat Probleme mit Kategorien, die mehrere Facetten haben (→ wieso?).
  - Verbesserte Versionen von Rocchio können deutlich effektiver sein (ähnlich komplizierteren Verfahren).
  - Um den Vorgang zu starten, ist eine Trainingskollektion notwendig (→ wozu?)
  - verwandt mit dem Relevance Feedback-Verfahren des gleichen Urhebers

Skizze kNN

Nearest Neighbor Algorithmus (kNN)

- Die kNN (k Nearest Neighbor) Methode verwendet ein Ähnlichkeitsmass (Euklidische Distanz, Kosinus) und eine Regel, wie Dokumente D Kategorien zuzuordnen sind.
- Einfache Regeln:
  - bestimme die k Dokumente, die am ähnlichsten zu Dokument D sind, d.h. die k nächsten "Nachbarn".
  - ordne Dokument D einer oder mehreren Kategorien zu, die bereits den Nachbarn zugeordnet sind
- Beachte:
  - Um den Vorgang zu starten, ist eine (ziemlich grosse) Trainingskollektion notwendig.

Erweiterte kNN Klassifikation

- Idee:
  - Je weiter ein Dokument D vom Nachbar D_j entfernt ist (Ähnlichkeitsmass!), desto weniger trägt es zum Entscheid bei, Dokument D in die Kategorie C_j zuzuordnen.
  - Mit anderen Worten: Berechne Wert s_c für jede potentielle Klasse C_j (eine simple Variante ist wie folgt):

$$sc(C_j, D) = \sum_{D_i \in kNN(D)} sim(D, D_i) \bullet a_{i,j}$$

  - Wobei kNN(D) die Menge von k nächsten Nachbarn von D ist.
  - $a_{i,j}$=1 falls Dokument D_i zu Klasse C_j gehört und $a_{i,j}$=0 andernfalls.
  - Probleme:
    - Richtige Wahl von k
    - Richtige Wahl der Funktion s_c, und der maximalen Anzahl zugeordneter Kategorien
    - Schwellwert bei Dokumenten, die mehreren Kategorien angehören sollten

Bewertung kNN

- Effektiv
- Relativ einfach, stabile Schwellwerte zu finden

- Langsam
- Die Wahl eines einzelnen Wertes "k" ist zu simpel

Bayes Klassifizierung

- *Gegeben:* Kategorie C_i mit einer angemessenen Anzahl von bereits zugeordneten Objekten (Trainingsdaten).
- *Methode:* Bilde statistische Modelle aus diesen Kategorien. Benutze diese, um vorherzusagen, zu welcher Klasse ein neues Objekt D gehört.
- Wir kennen $P(t|C_i)\ \forall\ t, C_i$, sind aber eigentlich interessiert an: $P(C_i|t)$ oder noch spezifischer in $P(C_i|D)$
  - wobei D für die *Menge von Merkmalen* in Objekt/Dokument D steht
- Die Wahrscheinlichkeit, dass D zu C_i gehört ist (Bayes'Rule):

$$P(C_i \mid D) = \frac{P(D \mid C_i)P(C_i)}{P(D)}$$

Bayes Klassifizierung

- Mit anderen Worten: „alte“ Objekte der Klasse C_i bestimmen für uns:
  - Die Merkmale nach den zu suchen ist
  - Erwartete Merkmalshäufigkeit in „neuen“ Objekten
- Mit $D=(t_1,\ldots,t_n)$ und in Beziehung zu Klasse C_i, wir können sagen:

$$P(D \mid C_i) = \prod_{j=1}^{n} P(t_j \mid C_i)$$

- Was bedeutet diese Annahme? Ist dies eine brauchbare Annahme?
- Die vorherigen Wahrscheinlichkeiten $P(C_i)$ und $P(D)$ müssen berechnet werden. Es gibt verschiedenen Wege um $P(t|C_i)$ zu berechnen: zähle die Anzahl Merkmale, binär (Vorkommen/Nicht-Vorkommen), gewichtet...

Bewertung Bayes

- Sauberes Modell
- Einfach zu implementieren

- Performt schlecht, typischerweise schlechter als andere einfache Verfahren
- Die Unanbhängigkeitsannahme ist wohl zu simpel

Exkurs: Regelbasierte Methode

- Expertensysteme versuchen, gewünschte Kategorien mittels geeigneter Regeln zu beschreiben.
- Beispielhaftes Vorgehen
  - Selektion von geeigneten Beispieldokumenten
  - Manuelle Selektion von Stichwörtern, Verknüpfung mittels logischem Ausdruck: Was für eine Suchanfrage ergibt diese Beispielsdokumente als Resultat?
  - Auch: automatische Herleitung von Regeln (hier nicht besprochen)
- Beachte Sie, dass es extrem schwierig ist, beständige (lange gültige) Anfragen zu formulieren (sogenannte Benutzerprofile).

Regelbasierte Methode (cont.)

- Regelbasierte Kategorisierung funktioniert relativ gut für sehr "scharfe" Konzepte. Sie können auch ergänzend als Filter eingesetzt werden.
- Beispiel: US Dollar vs. Australian Dollar
  - Sehr ähnliche Terminologie
  - Die (Trainings- und Test-)dokumente lassen sich im Vektorraum nicht gut voneinander abgrenzen
- Beispiel: Dokumente von der Credit Suisse vs. über die Credit Suisse
  - Der Unterschied «von/über» schlägt sich nicht wirklich im Vokabular nieder
  - Metadaten sind entscheidend
- Verbesserungen sind insbesondere dann möglich, wenn die zu suchenden Konzepte in speziellen Feldern vom Text auftreten (z.B. Metdaten, Titel etc.).

Collaborative Filtering anhand «The Netflix Prize»

- Collaborative Filtering ist eine Alternative zu Content-based Categorization, die grundlegend anders funktioniert: die Empfehlungen entstehen aufgrund von externen Signalen, wie Bewertungen
- Wir illustrieren die Idee anhand des «Netflix Prize». Dies ist aber nur der Aufhänger; die Überlegungen sind allgemeingültig.

Hintergrund «The Netflix Prize»

- Netflix existiert bereits länger als der gleichnamige Streamingdienst. In dieser «Frühzeit» war Netflix ein Versandanbieter für Miet-DVDs.

- Grundlegendes Problem: die Kunden/innen konnten den Film nicht anspielen, bevor sie diesen bestellen – und ein Fehlgriff war ärgerlich (Kosten, Zeitverlust).

- Bestmögliche Empfehlungen waren also essentiell, und Netflix ein Vorreiter in dieser Hinsicht.

Hintergrund «The Netflix Prize»

- Ausgelobt wurde ein Preisgeld von $1 Million. Ziel war es, den Hausalgorithmus «CineMatch» um mindestens 10% zu schlagen

- Bessere Resultate als CineMatch wurden bereits nach 6 Tagen veröffentlicht, aber es waren 2 Jahre nötig, um die gewünschten 10% zu erreichen

- In Hollywood-Manier kam es zum Schluss zu einem Fotofinish: ein Team gewann mit 20 Minuten Vorsprung.

Collaborative Filtering: Setup

- Gegeben sei ein unvollständiges Datenset (als Matrix User x Film interpretierbar)

- Der Algorithmus muss die fehlenden Bewertungen (Skala 1-5 Sterne) ergänzen. Aus diesen folgen dann die Empfehlungen.

	Inception	Avatar	Titanic	The Godfather	...
User 1	★★★★	★★★★★	★★★★★	★★	...
User 2	★	★★	—	★★★★★	...
User 3	★★★★★	—	★★★	—	...
User 4	???	★★★★	???	★★	...

Warum Collaborative Filtering?

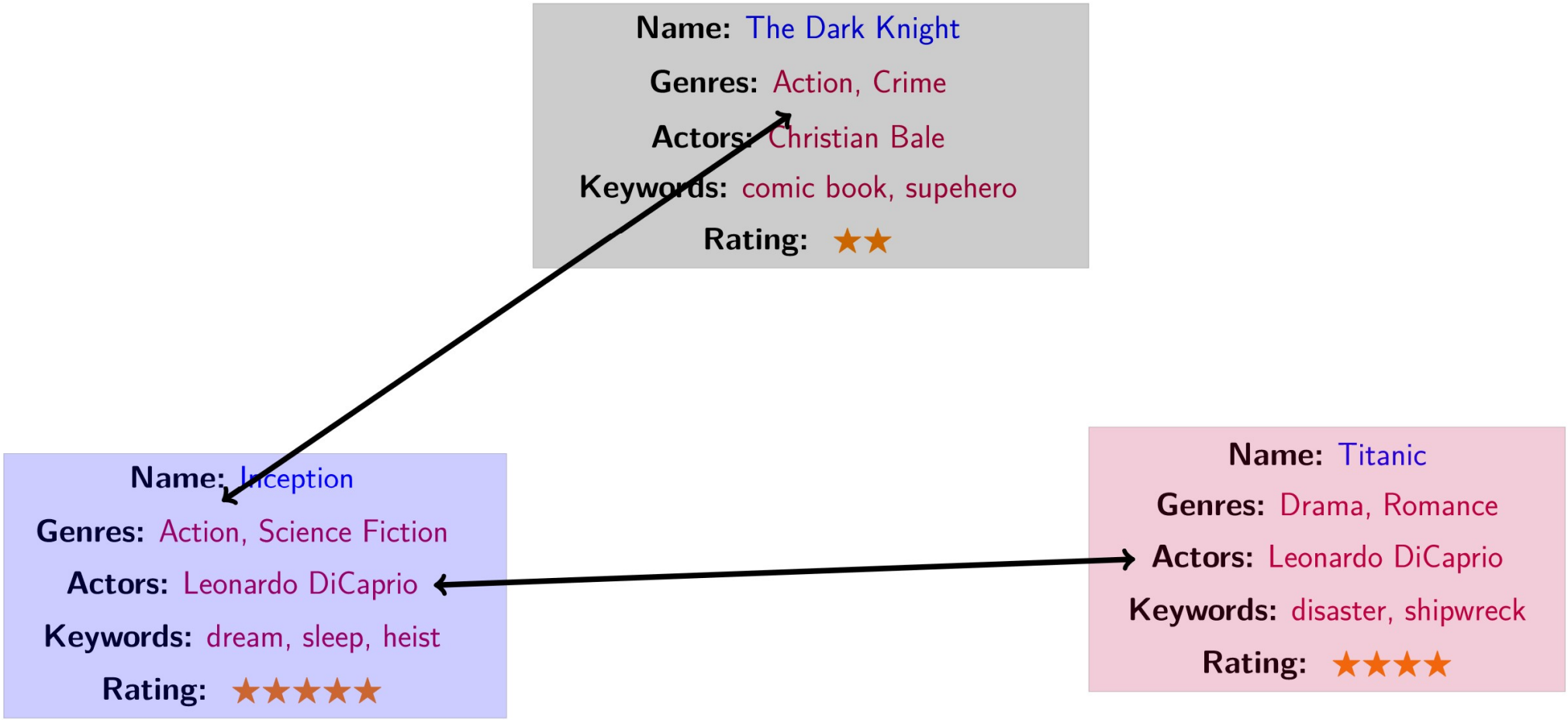
- Die Empfehlungen folgen aus den Bewertungen, nicht aus einer inhaltlichen Ähnlichkeit der Filme

- Beispiel: **User 1** scheint ähnliche Vorlieben wie **User 4** zu haben
- Wir folgern daraus, dass **User 4** *Titanic* und *Inception* mögen wird.

	Inception	Avatar	Titanic	The Godfather	...
User 1	★★★★	★★★★★	★★★★★	★★	...
User 2	★	★★	—	★★★★★	...
User 3	★★★★★	—	★★★	—	...
User 4	???	★★★★	???	★★	...

Content-Based Filtering

So, wie wir das Problem bis jetzt behandelt haben, müssten wir eine inhaltliche Ähnlichkeit z.B. auf Metadaten feststellen



Collaborative Filtering

Collaborative Filtering hat das Potential, abstrakte Beziehungen zwischen Daten und Vorlieben der Nutzer/innen aufzudecken

Beispiele

- Ist Jason Statham ein ähnlicher Schauspieler wie Vin Diesel?
- Sollen Tracks einer Tribute Band (z.B. “Kings of Floyd” o.ä.) vorgeschlagen werden?
- Soll ein Remake eines alten Filmes (z.B. “Ghostbusters” Reboot) empfohlen werden?

Collaborative Filtering

Content-basiert:

- beide spielen in vielen Actionstreifen. Aber auch Schauspielerinnen wie Scarlett Johansson haben ein ähnliches Portfolio – wir müssen also solche Tatsachen sauber gewichten
- Tribute Bands spielen die gleichen Songs, daher sehr ähnlich – aber auch in derselben Qualität?
- Reboots nutzen die selben Charaktere und Plotlines, daher sehr ähnlich – aber will der/die Nutzer/in nochmals “denselben” Film sehen?

Collaborative:

- Wenn die meisten Nutzer/innen, die Jason Statham mögen, auch gerne Filme mit Vin Diesel schauen, dann sind sie “ähnlich”
- Wenn die meisten Nutzer/innen die Tribute Band ignorieren, dann nicht “ähnlich”
- Wenn die meisten Nutzer/innen den Reboot ablehnen, dann nicht “ähnlich”

Collaborative Filtering

- Solche impliziten Beziehungen schlummern häufig in Daten, und die Nutzer/innen sind sich ihrer oft nicht bewusst

- Nutzer/innen folgen in ihren Präferenzen auch Mustern, z.B.
  - Nutzer/innen, die Action mögen, wollen keine Romantic Comedies
  - Kinder schauen einfache, kurze Filme, etc.

→ Ähnlichkeiten zwischen Nutzern können oft ein besserer Indikator für gute Empfehlungen sein als der eigentliche “Inhalt”

Collaborative Filtering

Gegeben: zwei ähnliche Nutzer/innen

User 1 mag wahrscheinlich Star Wars
User 2 mag wahrscheinlich keine spanischen Horrorfilme

Collaborative Filtering: kNN

Anpassung: k-Nearest Neighbours (kNN) Filtering

Ziel: Gesucht ist die Bewertung für Film *M* durch Nutzer/in *U*

1. Identifiziere die *k* ähnlichsten Nutzer/innen für *U* (U als Vektor darstellen, z.B. Cosinus-Ähnlichkeit)
2. Bilde Untermenge der Nutzer/innen, die Film *M* bewertet haben
3. Berechne den Durchschnitt der Scores

Collaborative Filtering

Beispiel: Amazons Item-to-Item Filtering

Ziel: Weitere Produkte zum Kauf vorschlagen

("Nutzer/innen, die **X** kaufen, kaufen auch **Y**")

1. Grundlage ist eine (binäre) Nutzer x Produkt-Matrix

	Item 1	*Item 2*	*Item 3*	*Item 4*	*Item 5*
User 1	1	1	0	0	1
User 2	0	1	0	0	1
User 3	1	0	1	1	0
User 4	0	0	1	1	0

Collaborative Filtering

Beispiel: Amazons Item-to-Item Filtering

2. Wir berechnen die Cosinus-Ähnlichkeit zwischen den Produktevektoren
3. Ähnlichstes Produkt (oder Produkte mit sim>delta) wird empfohlen

	Item 1	*Item 2*	*Item 3*	*Item 4*	*Item 5*
User 1	1	1	0	0	1
User 2	0	1	0	0	1
User 3	1	0	1	1	0
User 4	0	0	1	1	0

Produkt 2 → Produkt 5, Produkt 4 → Produkt 3

Collaborative Filtering

- Wir haben hiermit einerseits eine andere Interpretation von “Ähnlichkeit” – Vorlieben vs. inhaltliche Ähnlichkeit

- Dies ist insbesondere auch interessant, um eine grössere Diversität in die Resultate zu bekommen: nicht nur viele “Quasi-Doubletten”, sondern auch spannende “Ausreisser”

- Dies ist in vielen Bereichen entscheidend: wer einen Kühlschrank gekauft hat, braucht keine weiteren Kühlschränke mehr

Das Netflix Datenset

- Das Datenset bestand aus ca. **10,000,000 Bewertungen** von ca. **500,000 Nutzern/innen**

- Jedes System resp. Experiment musste weitere **3,000,000** Bewertungen liefern, in der Form von Fliesskommawerten zwischen 1.0 to 5.0

- Die Baseline hatte einen Fehler von ca. **0.95 stars**

Eine Bewertung dekonstruiert

Overall Average Rating:	+3.1 x ★
User-Critic Effect:	–0.3 x ★
Movie-Specific Deviation:	+0.7 x ★
Unkown Factor:	+0.6 x ★
Final Score:	4.1 x ★

Eine Bewertung kann als eine Summe von Komponenten aufgefasst werden. Einige dieser Komponenten können einfach bestimmt werden.

“Overall Average” ist der Durchschnitt über alle Filme

Die Frage ist also: wie weicht der konkrete Film ab?

Eine Bewertung dekonstruiert

Overall Average Rating:	+3.1 x ★
User-Critic Effect:	–0.3 x ★
Movie-Specific Deviation:	+0.7 x ★
Unkown Factor:	+0.6 x ★
Final Score:	4.1 x ★

Der “User-Critic effect” misst den Nutzer-Faktor: ist dies im Vergleich zur ganzen Population eine kritische oder eine wohlwollende Person? Sinngemäss wirkt die “Movie-Specific Deviation”: ist dies im Prinzip ein populärer oder ein ungeliebter Film?

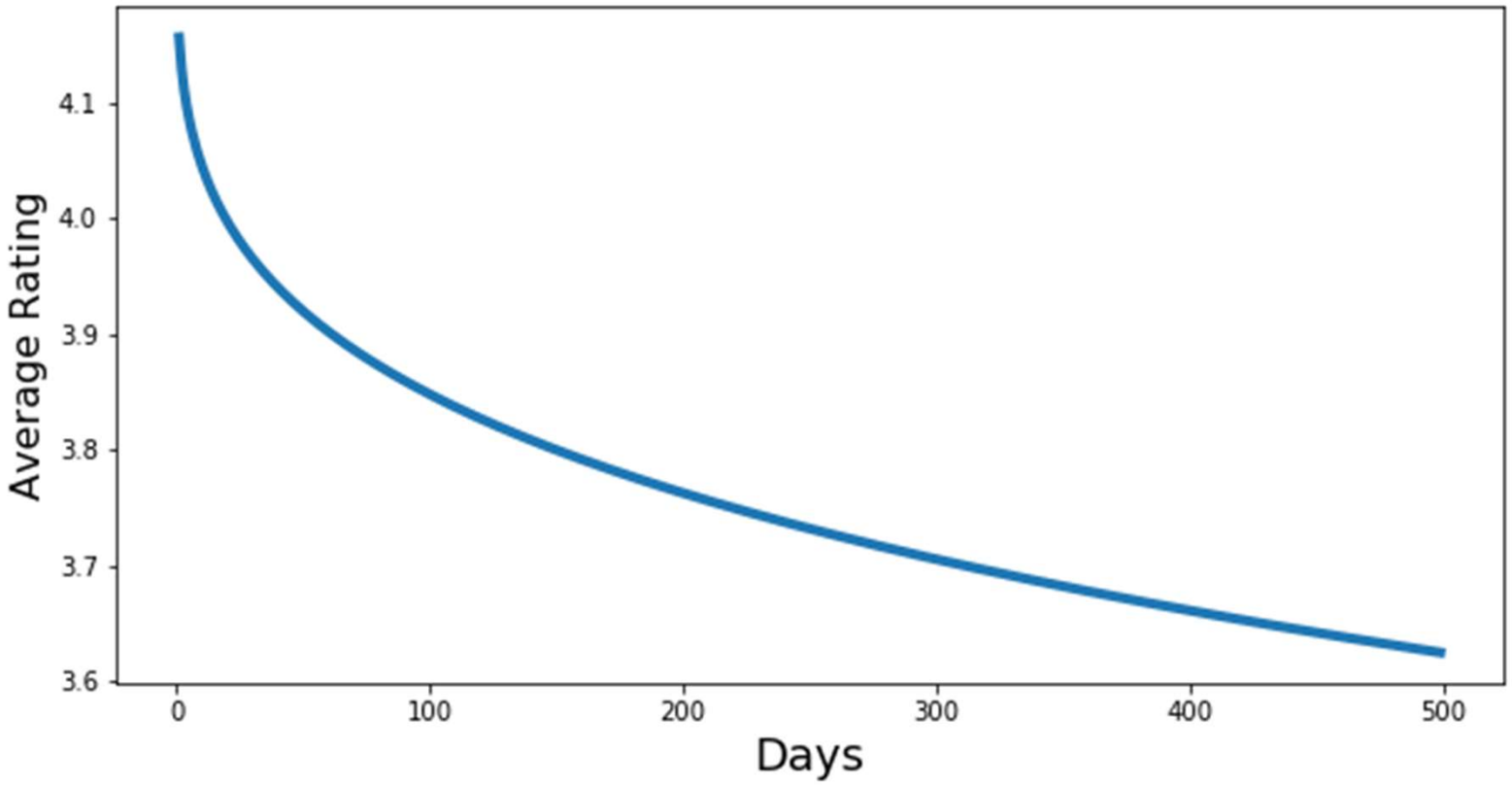
Eine Bewertung dekonstruiert

Overall Average Rating:	+3.1 x ★
User-Critic Effect:	–0.3 x ★
Movie-Specific Deviation:	+0.7 x ★
Unkown Factor:	+0.6 x ★
Final Score:	4.1 x ★

Wenn wir diese drei einfach bestimmbaren Faktoren entfernen, können wir den spannenden Teil isolieren: die konkrete Präferenz eines bestimmten Nutzers in Hinsicht auf diesen spezifischen Film – hier 0.6 Sterne höher als erwartet

Nutzer Biases

- Weitere Effekte müssen berücksichtigt werden: je mehr Filme ein Nutzer/in schaut, desto kritischer werden die Bewertungen



Nutzer Biases

Wenn wir einen Tag isoliert betrachten, dann gilt für einen spezifischen Nutzer/in, der/die mehrere Filme (nicht unbedingt an diesem Tag geschaut) bewertet:

- Die Tagesstimmung beeinflusst die Berwertungen. Alle Filme werden entweder besser oder schlechter bewertet
- Nur Filme, welche “Ausreisser” sind, werden bewertet, d.h, die besten und schlechtesten
- Wir erhalten tendenziell keine Bewertungen für mittelmässige Filme

Nutzer Biases

- Die Berücksichtigung solcher spezifischen Biases ist essentiell (Datenaufbereitung, Data Engineering)

- Der Einfluss ist grosser als der Algorithmus an sich. Bereits die Baseline funktioniert auf bereinigten Daten bedeutend (signifikant) besser)

Netflix-Prize: Sieger

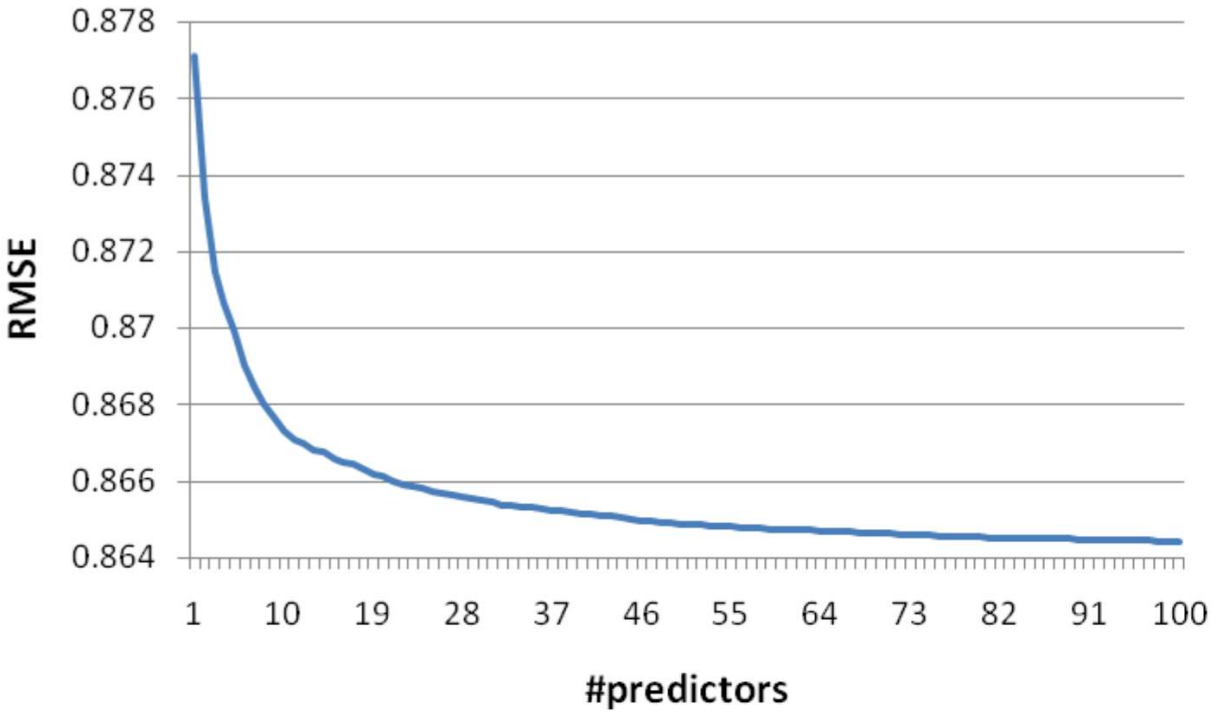
- Das Siegersystem war eine Linearkombination von **mehr als 200 verschiedenen Algorithmen** (“Kitchen-sink approach”)

- Unter anderem verwendet wurden:
  - Nearest Neighbours (kNN)
  - Matrix Factorisation (SVD)
  - Neural Networks
  - Decision Trees

- Dies wurde kombiniert mit einer Datenbereinigung (siehe oben)

Analyse Lösung Sieger

Aber: der “Return on investment” bei einer solchen Kombination sinkt rapide. Schon 2 Methoden bringen 75% der Verbesserung (und sind viel besser skalierbar!)



Reflexion Collaborative Filtering

- Unsere bestehenden Ansätze (z.B. kNN) können auch für Collaboratives Filering angepasst werden
- Möglichkeit, implizite “Signale” in den Daten zu nutzen
- Interessant in Sachen Diversität: nicht nur “Quasi-Doubletten”, sondern inhaltlich andere Objekte vorschlagen
- Probleme mit “Kaltstart”
- Sehr interessant: Hybrid aus content-based und collaborative