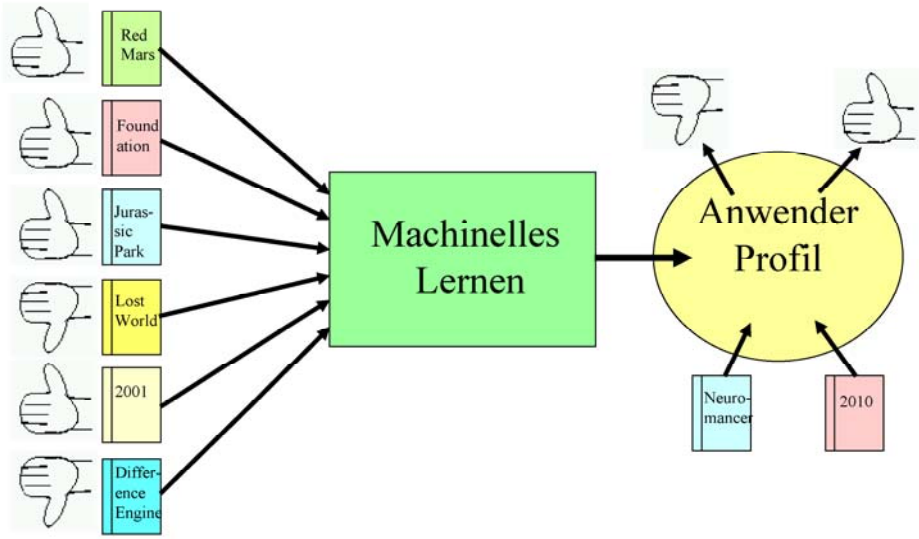
Recommender-Systeme

Kollaboratives Filtern &
inhaltsbasierte Empfehlungen

Empfehlungs-Systeme

- Systeme, um Nutzern Dinge zu empfehlen (z.B. Bücher, Filme, CDs, Webseiten, Newsgroup Nachrichten), die auf ihren vorigen Präferenzen basieren.
- Viele On-line-Läden liefern Empfehlungen (z.B. Amazon, CDNow).
- Recommender haben den Umsatz von On-Line-Läden erheblich gesteigert.
- Es gibt zwei grundlegende Ansätze für Empfehlungen:
  - kollaboratives Filtern (a.k.a. soziales Filtern)
  - inhaltsbasiertes Filtern

Buch-Recommender



Personalisierung

- Recommender ist spezielle Personalisierungs-software.
- Die Personalisierung betrifft die Anpassung an individuelle Bedürfnisse, Interessen und Präferenzen jedes Anwenders.
- Recommender umfassen:
  - Empfehlen
  - Filtern
  - Vorhersagen (z.B. Formular-Vervollständigungen)
- Aus geschäftlicher Perspektive werden Recommender als Teil des Customer Relationship Management (CRM) angesehen.

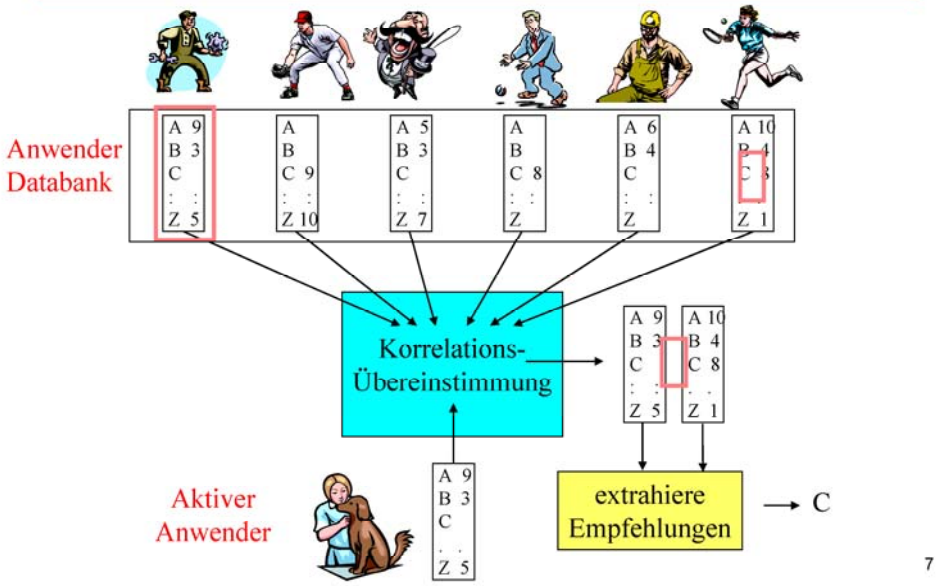
Machinelles Lernen und Personalisierung

- Machinelles Lernen kann das Lernen eines *Anwendermodells* oder *Profils* eines bestimmten Benutzers unterstützen, basierend auf:
  - Interaktionsmustern
  - bewerteten Beispielen
- Dieses Modell oder Profil kann dann verwendet werden um:
  - Objekte zu empfehlen
  - Informationen zu filtern
  - Verhalten vorherzusagen

Kollaboratives Filtern

- Pflegen einer Datenbank mit Anwender-bewertungen einer Vielzahl von Objekten.
- Finde für einen gegebenen Anwender andere, ähnliche Anwender, deren Bewertungen stark mit dem aktuellen Anwender korrelieren.
- Empfehle Objekte, die von diesen ähnlichen Anwendern hoch eingestuft werden, aber vom aktuellen Anwender noch nicht bewertet wurden.
- Nahezu alle vorhandenen kommerziellen Recommender verwenden diesen Ansatz (z.B. Amazon).

Kollaboratives Filtern



Kollaborative Filtermethode

- Gewichte alle Anwender in Bezug auf ihre Ähnlichkeit mit dem aktiven Anwender.
- Wähle eine Teilmenge der Anwender aus (*Nachbarn*), um sie zur Vorhersage zu verwenden.
- Normalisiere Bewertungen und berechne eine Vorhersage aus einem gewichteten Mittel der ausgewählten Nachbar-Bewertungen.
- Präsentiere Objekte mit den höchsten vorhergesagten Bewertungen als Empfehlungen.

Ähnlichkeitsgewichtung

- Verwende typischerweise den Pearson-Korrelations-koeffizienten zwischen Bewertungen für den aktiven Anwender *a* und einem weiteren Anwender *u*.

$$c_{a,u} = \frac{\text{covar}(r_a, r_u)}{\sigma_{r_a}\sigma_{r_u}}$$

r_a und r_u sind die Bewertungsvektoren für die m Objekte, die **sowohl** von a als auch von u bewertet sind.

$r_{u,j}$ ist die Bewertung von Anwender u für das Objekt j.

Kovarianz und Standard-Abweichung

- Kovarianz:

$$\text{covar}(r_a, r_u) = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{m}$$

$$\bar{r}_x = \frac{\sum_{i=1}^{m} r_{x,i}}{m}$$

- Standard-Abweichung:

$$\sigma_{r_x} = \sqrt{\frac{\sum_{i=1}^{m}(r_{x,i} - \bar{r}_x)^2}{m}}$$

Signifikanz-Gewichtung

- Es ist wichtig, keinen Korrelationen zu vertrauen, die nur auf sehr wenigen gemeinsam bewerteten Objekten basieren.
- Verwende *Signifikanzgewichte* $s_{a,u}$, die auf der Anzahl vom gemeinsam bewerteten Objekten, m basieren.

$$w_{a,u} = s_{a,u}c_{a,u} \qquad \text{mit} \qquad s_{a,u} = \begin{cases} 1 \text{ if } m > 50 \\ \frac{m}{50} \text{ if } m \le 50 \end{cases}$$

Nachbar-Selektion

- Auswahl der zu dem aktiven Anwender *a* am stärksten korrelierenden Anwender, die dann als Quelle der Vorhersagen dienen.
- Der Standardansatz ist, die n ähnlichsten Anwender u zu verwenden, basierend auf den Ähnlichkeitsgewichten $w_{a,u}$.
- Ein alternativer Ansatz ist es, alle Anwender einzuschließen, deren Ähnlichkeit zu a über einer gegebenen Schwelle liegt.

Bewertungs-Vorhersage

- Sage unter Verwendung der n ausgewählten Nachbar-Anwender $u \in \{1,2,\ldots,n\}$ für den aktiven Anwender a eine Bewertung $p_{a,i}$ für jedes Objekt i voraus .
- Um die verschiedenen Bewertungsniveaus unterschiedlicher Anwender zu berücksichtigen, basieren wir die Vorhersagen auf der Differenz zum Durchschnitt aller Bewertungen des jeweiligen Nutzers.
- Gewichte die Bewertungsbeiträge des Anwenders nach ihrer Ähnlichkeit mit dem aktiven Anwender.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{n} w_{a,u}}$$

Probleme mit kollaborativem Filtern

- **Kaltstart**: Es müssen bereits genug andere Anwender im System sein, um eine Übereinstimmung zu finden.
- **Seltenheit**: Wenn viele Objekte empfohlen werden sollen, ist die Anwender/Bewertungsmatrix dünn besetzt – selbst wenn es viele Anwender gibt – und es ist schwierig, Anwender zu finden, die die gleichen Objekte bewertet haben.
- **Erster Beurteiler**: koll. Filtern kann kein Objekt empfehlen, das nicht zuvor bewertet worden ist.
  - Neue Objekte
  - Exotische Objekte
- **Popularitäts-Ausrichtung**: koll. Filtern kann jemandem mit sehr speziellen Vorlieben keine Objekte empfehlen.
  - Die Methode neigt dazu, populäre Objekte zu empfehlen.

Inhaltsbasiertes Empfehlen

- Empfehlungen basieren hier eher auf Informationen über den Inhalt (die Eigenschaften) von Objekten als auf den Meinungen anderer Anwender.
- Verwendet einen Algorithmus für maschinelles Lernen, um ein Profil der Anwenderpräferenzen aus Beispielen zu erzeugen, die auf Merkmalsbeschreibungen des Inhalts basieren.
- Einige existierende Anwendungen:
  - Newsweeder (Lang, 1995)
  - Syskill und Webert (Pazzani et al., 1996)

Vorteile eines inhaltsbasierten Ansatzes

- Kein Bedarf an Daten über andere Anwender.
  - Kein Kaltstart-Problem und keine Seltenheitsprobleme.
- Ist fähig, Anwendern mit eindeutigen Vorlieben Empfehlungen auszusprechen
- Ist fähig, neue und unpopuläre Objekte zu empfehlen
  - Kein Erst-Beurteiler-Problem.
- Kann Erläuterungen zu den empfohlenen Objekten durch die Auflistung der Inhaltsmerkmale liefern, die die Empfehlung bewirkten.

Nachteile der inhaltsbasierten Methode

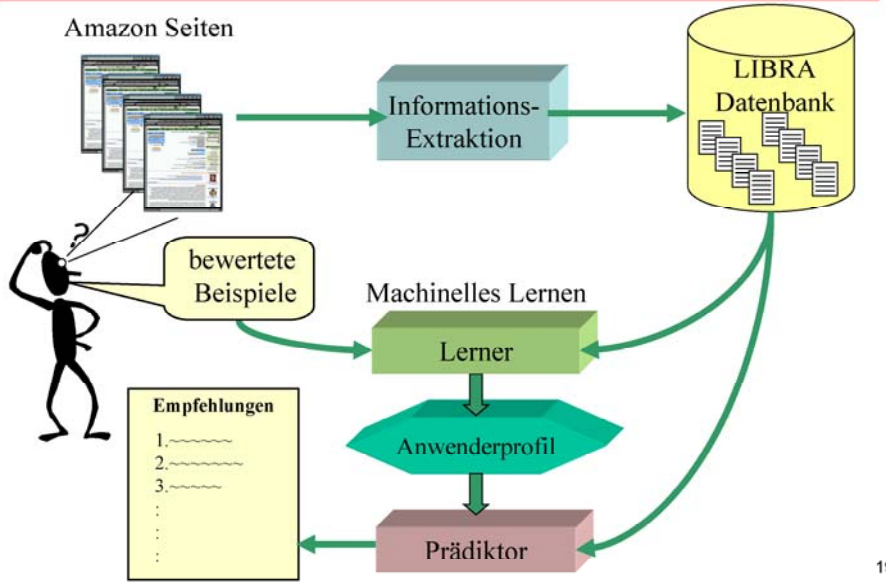
- Erfordert Inhalt, der sinnvoll durch Merkmale kodiert werden kann.
- Anwender-Vorlieben müssen als lernbare Funktion dieser Inhaltsmerkmale dargestellt werden können.
- Nicht fähig, Qualitätsbeurteilungen anderer Anwender auszuwerten.
  - Es sei denn, diese sind irgendwie in den Inhaltsmerkmalen enthalten.

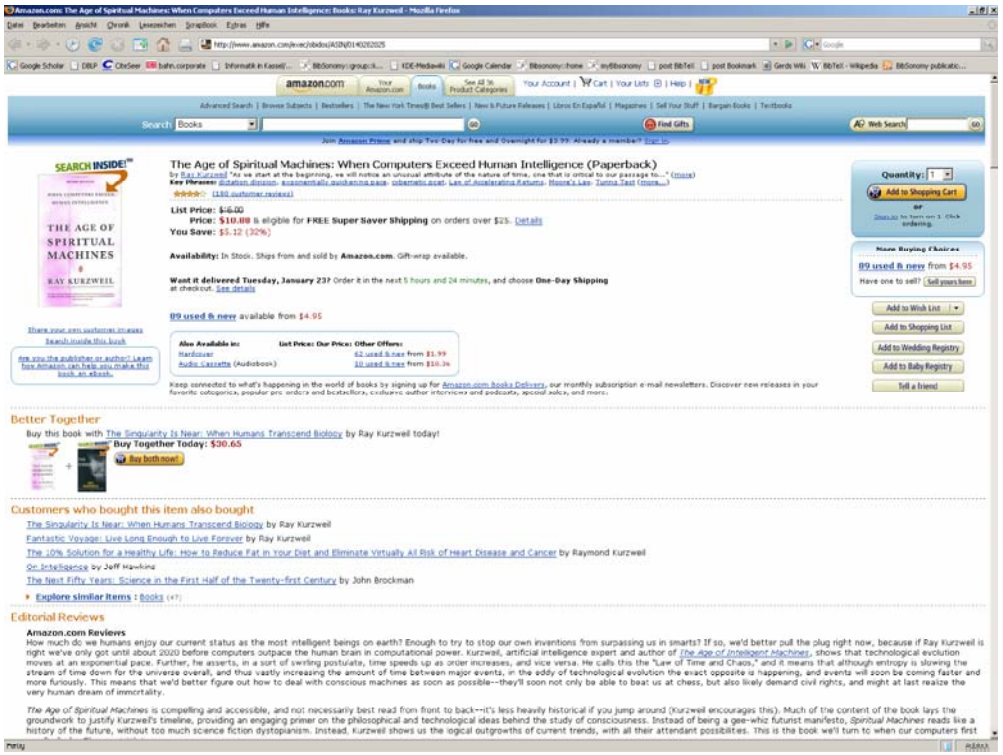
LIBRA

Learning Intelligent Book Recommending Agent

- Inhaltsbasierender Recommender für Bücher, der Informationen über Titel verwendet, die von Amazon extrahiert wurden.
- Verwendet Informations-Exktration aus dem Web, um Text in Feldern zu organisieren:
  - Autor
  - Titel
  - Redaktionnelle Reviews
  - Stellungnahmen von Kunden
  - Themenbezeichnungen
  - Zugehörige Autoren
  - Zugehörige Titel

LIBRA System





Bsp.: von Amazon extrahierte Informationen

Titel: <The Age of Spiritual Machines: When Computers Exceed Human Intelligence>
Autor: <Ray Kurzweil>
Preis: <11.96>
Datum der Veröffentlichung: <Januar 2000>
ISBN: <0140282025>
zugehörige Titel: <Titel: <Robot: Mere Machine or Transcendent Mind>
Autor: <Hans Moravec> >
…
Reviews: <Autor: <Amazon.com Reviews> Text: <How much do we humans…> >
…
Stellungnahmen: <Stars: <4> Autor: <Stephen A. Haines> Text:<Kurzweil has …> >
…
zugehörige Autoren: <Hans P. Moravec> <K. Eric Drexler>…
Betreffs: <Science/Mathematics> <Computers> <Artificial Intelligence> …

Libra: Inhaltsinformationen

- Libra verwendet diese Information, um die folgenden Slots mit "bags of words" zu füllen:
  - Autor
  - Titel
  - Beschreibung (Stellungnahmen und Kommentare)
  - Betreffs
  - verwandte Titel
  - verwandte Autoren

Libra: Übersicht

- Anwender bewertet ausgewählte Titel auf einer Skala von 1 bis 10.
- Libra verwendet einen Naive-Bayes-Text-Kategorisierungs-Algorithmus, um daraus ein Profil zu erlernen.
  - Bewertung 6–10: positiv
  - Bewertung 1–5: negativ
- Von den anderen Büchern werden diejenigen als Empfehlungen klassifiziert, bei denen die errechnete Wahrscheinlichkeit für eine positive Einschätzung groß genug ist.
- Der Anwender kann auch explizite positive/negative Schlüsselwörter liefern, die dann höher gewichtet werden.

Bayesian Kategorisierung in LIBRA

- Das Model ist verallgemeinert, um einen **Vektor** von "bags of words" zu erzeugen (ein "bag" für jeden Slot).
  - Instanzen desselben Worts in verschiedenen Slots werden als separate Merkmale behandelt:
    - "Chrichton" bei Autor – "Chrichton" in der Beschreibung
- Trainingsbeispiele werden als positiv oder negativ *gewichtete* Beispiele bei der Schätzung der bedingten Wahrscheinlichkeitsparameter behandelt:
  - Gegeben ist ein Beispiel mit Bewertung $1 \le r \le 10$:
    positive Wahrscheinlichkeit: $(r-1)/9$ (w_r)
    negative Wahrscheinlichkeit: $(10-r)/9$

Wenn ein Wort in n Trainingsbeispielen mit Rating r vorkommt:

- für die positive Klasse wird dann $n*w_r$ gezählt

Implementierung

- Stopwörter wurden von allen “bags of words” entfernt.
- Buchtitel und Autoren werden auch zu den Slots “verwandte Titel” bzw. “verwandte Autoren” hinzugefügt.
- Alle Wahrscheinlichkeiten werden wegen der kleinen Datenbasis mit der Laplace-Schätzung geglättet.
- Lisp-Implementierung ist ziemlich effizient:
  - Training: 20 Datensätze in 0.4 s, 840 in 11.5 s
  - Test: 200 Bücher pro Sekunde

Erläuterung von Profilen und Empfehlungen

- Stärke des Auftretens von Wort w_k in Slot s_j:

$$\text{strength}(w_k, s_j) = \log \frac{P(w_k \mid \text{positive}, s_j)}{P(w_k \mid \text{negative}, \text{s}_\text{j})}$$

Experimentelle Daten

- Bücher aus verschiedenen Genres von Amazon.
- Titel, die zumindest eine Stellungnahme oder einen Kommentar haben, wurden ausgewählt.
- Datensätze:
  - Literaturfiktion: 3,061 titles
  - Mysterium: 7,285 titles
  - Wissenschaft: 3,813 titles
  - Science Fiction: 3,813 titles

Bewertete Daten

- Vier Anwender bewerteten zufällige Beispiele innerhalb eines Genres, indem sie die Amazon-Seiten hinsichtlich der Titel durchsahen:
  - LIT1 936 titles
  - LIT2 935 titles
  - MYST 500 titles
  - SCI 500 titles
  - SF 500 titles

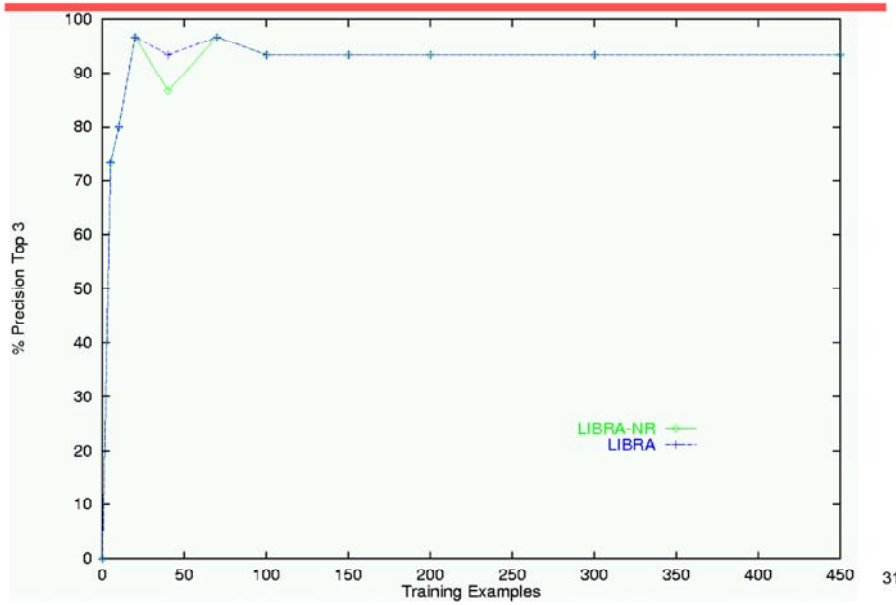
Experimentelle Methode

- 10-fache Kreuz-Validierung, um Lernkurven zu erzeugen.
- Auf unabhängigen Testdaten wurde gemessen:
  - Präzision der Top 3: % der besten 3, die positiv sind
  - Bewertung der Top 3: Durchschnittsbewertung, die den besten 3 zugeordnet ist
  - Klassifizierungs-Korrelation: Spearman's, r_s, zwischen vollständigen Klassifizierungen von System und Anwender.
- Test ohne die Slots "verwandter Autor" und "verwandter Titel" (LIBRA-NR).
  - Testet den Einfluss von Informationen, die durch den kollaborativen Ansatz von Amazon erzeugt wurden.

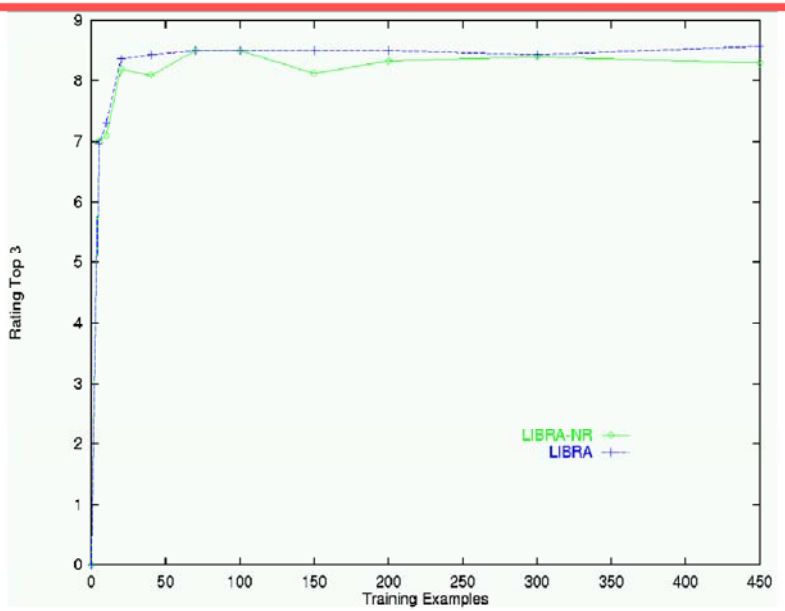
Experimentelle Ergebnisse: Zusammenfassung

- Präzision der Top 3 ist nach nur 20 Beispielen mit um die 90 % ziemlich konsistent.
- Bewertung der Top 3 ist nach nur 20 Beispielen mit über 8 ziemlich konsistent.
- Alle Ergebnisse sind nach nur 5 Beispielen immer bedeutend besser als eine zufällige Auswahl.
- Klassifizierungs-Korrelation liegt nach nur 10 Beispielen generell über 0.3 (gemäßigt).
- Klassifizierungs-Korrelation liegt nach nur 40 Beispielen generell über 0.6 (hoch).

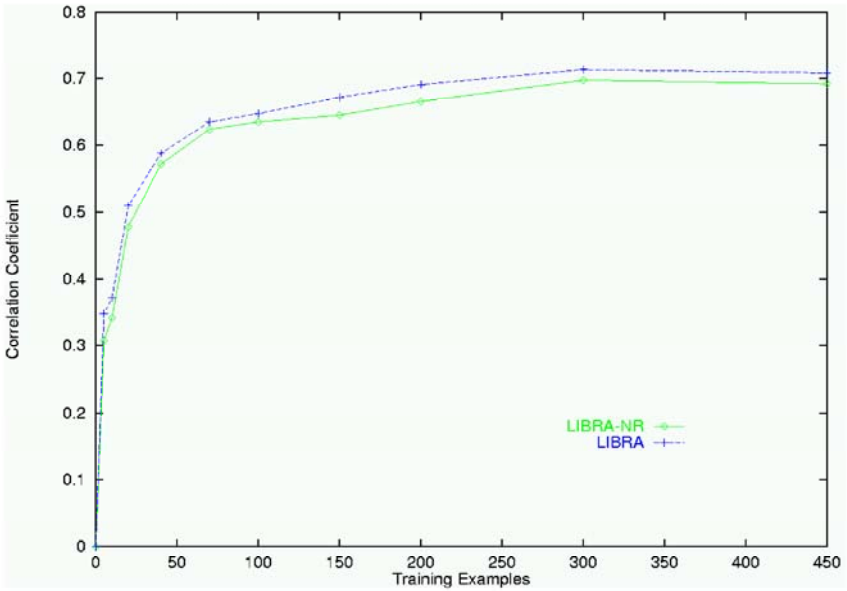
Präzision der Top 3 für "Science"



Bewertung der Top 3 für "Science"



Klassifizierungs-Korrelation für “Science”



Anwenderstudie

Nutzer

- wurden gebeten, Libra zu verwenden und Empfehlungen anzufragen.
- wurden zu mehreren Feedback-Runden ermutigt.
- bewerteten alle Bücher der endgültigen Empfehlungs-Liste.
- wählten zwei Bücher zum Kauf.
- schickten nach Lesen der Auswahl Stellungnahmen zurück.
- vervollständigten Fragebogen über das System.

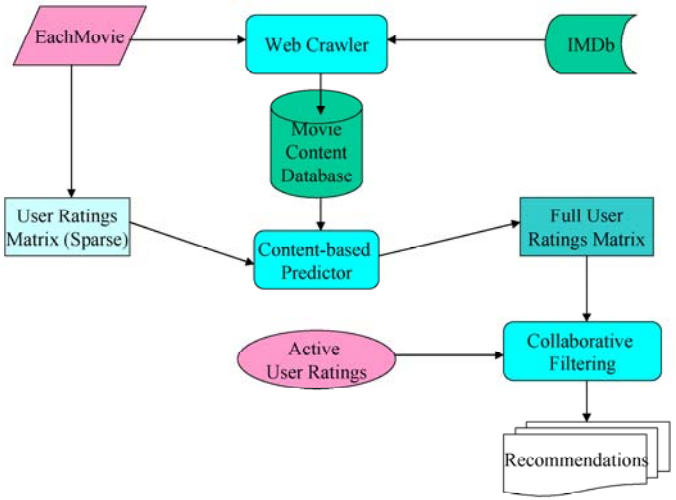
Verbinden von Inhalt und Kollaboration

- Inhaltsbasierte und kollaborative Methoden haben komplementäre Stärken und Schwächen.
- Kombiniere Methoden, um das Beste von beiden zu erhalten.
- Es gibt verschiedene hybride Ansätze:
  - Wende beide Methoden an und verknüpfe die Empfehlungen.
  - Verwende kollaborative Daten als Inhalt.
  - Verwende inhaltsbasierte Empfehlungen als weiteren Nutzer, dessen Verhalten in weitere Vorhersagen einfließt.
  - **Verwende einen inhaltsbasierten Empfehler, um kollaborative Daten zu vervollständigen.**

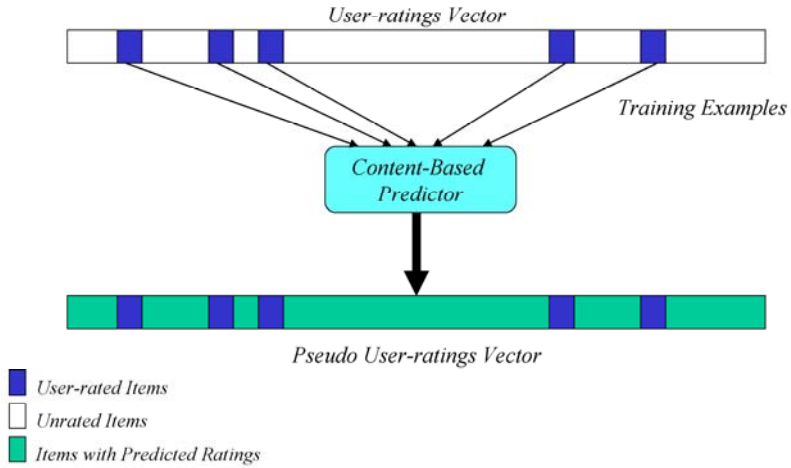
Anwendung: Movies

- *EachMovie* Datensatz [Compaq Research Labs]
  - enthält Anwenderbewertungen für Filme auf einer Skala von 0 bis 5.
  - 72,916 Anwender (mit durchschn. 39 Bewertungen).
  - 1,628 Filme.
  - Spärliche besetzte Anwender-Bewertungsmatrix – (zu 2.6% besetzt).
- Internet Movie Database (*IMDb*)
  - Wurde für die Titel aus *EachMovie* gecrawlt.
- Wesentliche Filminformationen:
  - Titel, Direktor, Rollenbesetzung, Genre, etc.
- Populäre Meinungen:
  - Nutzerkommentare, Zeitungs- und Newsgroup-Berichte, etc.

Content-Boosted kollaboratives Filtern



Content-Boosted CF - I



Content-Boosted CF - II

- Berechne Pseudo-Anwenderbewertungsmatrix
  - Volle Matrix – approximiert die konkreten Nutzerbewertungen
- Führe CF aus
  - Unter Verwendung von Pearson-Korr.-Koeeffizient zwischen den Pseudo-Anwender-Bewertungsvektoren

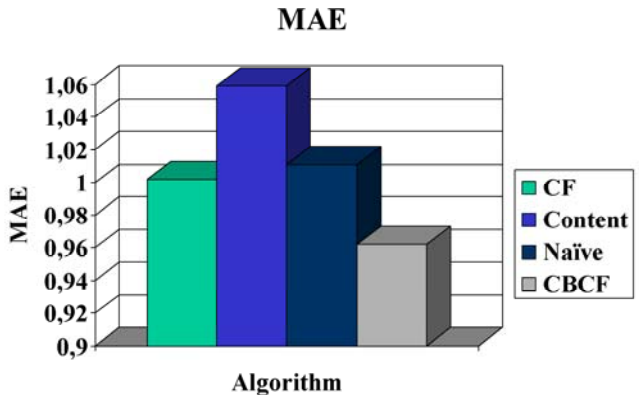
Experimentelle Evaluierung

- Teilmenge von *EachMovie* wurde verwandt:
  - 7,893 Anwender; 299,997 Bewertungen
- Testmenge: 10% der Anwender wurden zufällig ausgewählt.
  - Sie bewerteten je mindestens 40 Filme.
  - Training auf der verbleibenden Menge.
- "Hold-out"-Menge: 25% der Objekte für jeden Testanwender.
  - Vorhersage der Bewertung für jedes Objekt in der "Hold-out"-Menge.
- Vergleich CBCF mit anderen Ansätzen:
  - Reines CF
  - Rein inhaltsbasiert
  - hybrider Naïve Bayes (Durchschnitts-CF und inhaltsbasierende Vorhersagen)

Maße

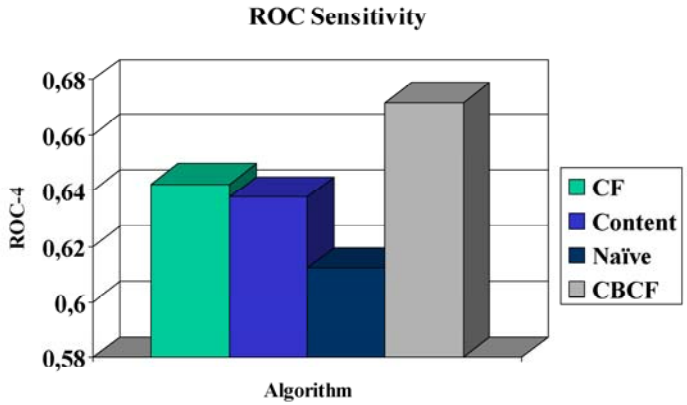
- Mittlerer absoluter Fehler (MAE)
  - Vergleicht die Vorhersagen mit Anwenderbewertungen
- ROC Empfindlichkeit [Herlocker 99]
  - Wie gut helfen die Vorhersagen den Anwendern, eine gute Auswahl zu treffen?
  - Bewertungen ≥ 4 als "gut" betrachtet; < 4 als "schlecht" betrachtet
- Gepaarter T-Test stellt die statistische Signifikanz fest.

Ergebnisse - I



CBCF ist bedeutend besser (4% über CF) bei ($p < 0.001$)

Ergebnis - II



CBCF übertrifft Rest (5% Verbesserung über CF)

Aktives Lernen

- Wird verwendet, um die Anzahl der Trainingsbeispiele zu verringern.
- System fordert Bewertungen für spezifische Objekte, von denen es vermutet, am meisten zu lernen.
- Verschiedene existierende Methoden:
  - Uncertainty Sampling
  - Komitee-basiertes Sampling

Halbüberwachtes Lernen
(weakly supervised, Bootstrapping)

- Verwende die große Anzahl ungekennzeichneter Beispiele, um das Lernen von einer kleinen Menge von gekennzeichneten Beispielen zu unterstützen.
- Einige neue Methoden:
  - Halbüberwachtes EM (Erwartungsmaximierung)
  - Ko-Training
  - Transduktive SVM's

Schlussfolgerungen

- Empfehlungen und Personalisierung sind wichtige Ansätze zur Bekämpfung der Informations-Überfrachtung.
- Machinelles Lernen ist ein wichtiger Teil der Systeme zur Lösung diese Aufgaben.
- Kollaboratives Filtern hat Probleme.
- Inhaltsbasierende Methoden sprechen diese Probleme an (haben aber eigene Probleme).
- Das Beste ist, beides zu integrieren.