Item-based Collaborative Filtering

Initial implementation

Martin Krüger, Sebastian Kölle

12.05.2011

Seminar Collaborative Filtering

Projektplan

Implementierung

Ideen

Wdh.: Item-based Collaborative Filtering

Vorberechnung

Erstelle eine Item-Item Matrix, berechne dabei die Ähnlichkeit jedes Item-Paares unter Verwendung eines Ähnlichkeitsmaßes (*Cosinus-based*, *Correlation-based* oder *Adjusted Cosine similarity*).

$$s_{ij} = \frac{\sum_{u \in U} (r_{ui} - \overline{r}_u)(r_{uj} - \overline{r}_u)}{\sqrt{\sum_{u \in U} (r_{ui} - \overline{r}_u)^2} \cdot \sqrt{\sum_{u \in U} (r_{uj} - \overline{r}_u)^2}}$$

Vorhersage

Gegeben: User **u**, Item **i**. Gesucht: Rating $\mathbf{r_{ui}}$

1. Finde die **K** zu **i** ähnlichsten Nachbarn **N(i;u)**, die von **u** bewertet wurden.
2. Berechne den gewichteten Mittelwert auf Basis der Ähnlichkeiten oder berechne das Rating mit einem Regressionsmodell.

$$r_{ui} = \frac{\sum_{j \in N(i;u)} s_{ij} r_{uj}}{\sum_{j \in N(i;u)} |s_{ij}|}$$

[1, S. 6 ff]

Herausforderungen für den KDD Cup

- Großer Speicherbedarf: mindestens 364 GB für vollständige Speicherung der Item-Item- und Â-Matrix (ein Byte pro Paar – sehr optimistisch …)
  - Sampling für die Entwicklung
  - Space-time-tradeoff
- Berücksichtigung der Hierarchie
  - Reduzierung des Speicheraufwandes: Vergleich ausschließlich Items gleichen Genres?
  - Verbesserung der Vorhersagen: Macht z.B. die gute Bewertung eines Albums durch einen Nutzer es wahrscheinlicher, dass er auch die einzelnen Titel gut bewertet?
- Herausfinden der optimalen Parameterwerte für die gegebenen Daten
  - Testumgebung, die das systematische Ausprobieren mit verschiedenen Samples erlaubt

Herausforderungen für den KDD Cup

- Großer Speicherbedarf:
  - **Sampling für die Entwicklung**
  - **Space-time-tradeoff**

  + platzsparende Repräsentation der Matrizen

    - Keine Vorberechnung der gesamten Item-Item Matrix
    - Für Vorhersage zu einem Item für einen User mit N bewerteten Items eine N:1 Matrix berechnen.
- Berücksichtigung der Hierarchie
  - Reduzierung des Speicheraufwandes: Vergleich ausschließlich Items gleichen Genres?
  - **Verbesserung der Vorhersagen: Macht z.B. die gute Bewertung eines Albums durch einen Nutzer es wahrscheinlicher, dass er auch die einzelnen Titel gut bewertet?**
- Herausfinden der optimalen Parameterwerte für die gegebenen Daten
  - Testumgebung, die das systematische Ausprobieren mit verschiedenen Samples

 = implementiert  = Idee

Meilensteine

12. Mai	loader completely implemented and tested initial implementation presentation
15. Mai	basic item-item algorithm implementation finished first submission with complete dataset
26. Mai	global effects identified & removed *maybe:* split by item type *maybe:* neighborhood relationship model implemented
9. Juni	implemented hierarchy-based algorithm integration of different algorithms intermediate presentation
23. Juni	implementation finished parameters tweaked for optimal result final presentation
30. Juni	submission deadline

Projektplan

Implementierung

- Effiziente Speicherung der User-Item Matrix
- Sampling

Ideen

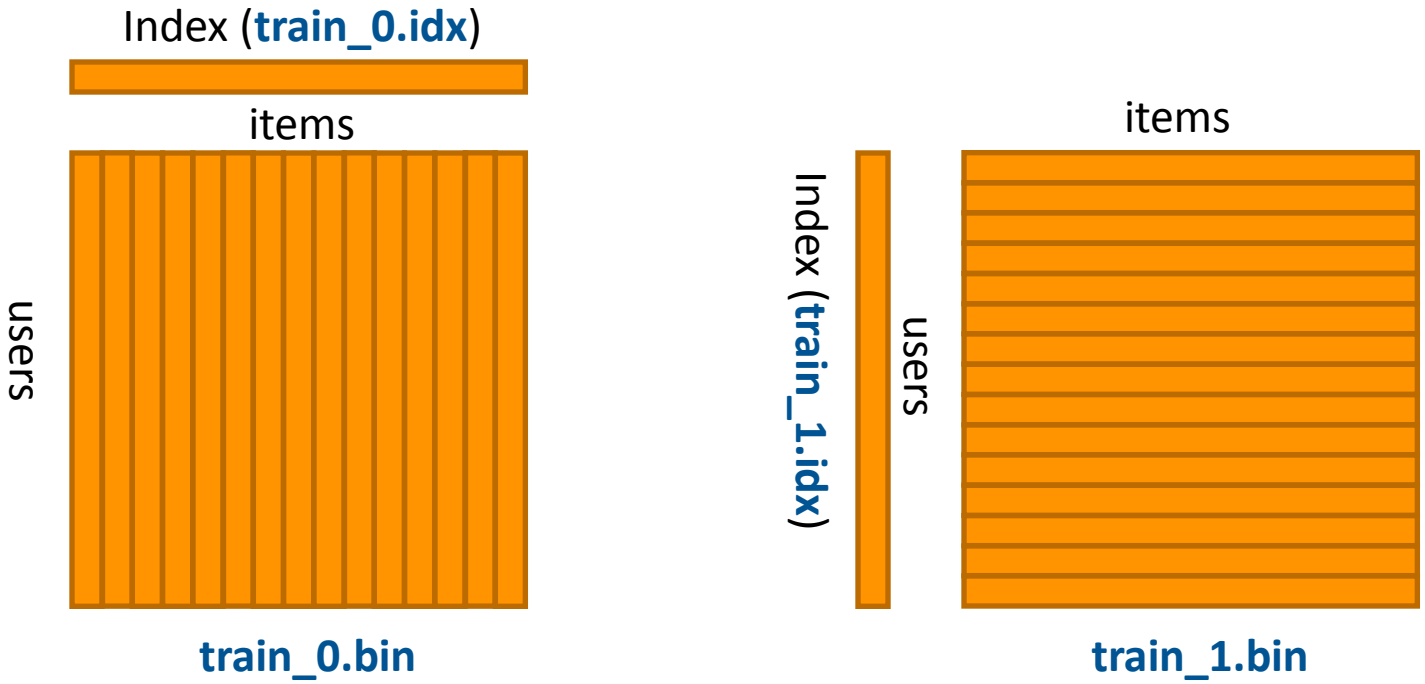
Effiziente Speicherung der User-Item Matrix

- **Problem**: Größe der User-Item-Matrix
  - 1.000.990 Users x 624.961 Items x 1 Byte ≈ 582,62 GB
- Benötigte **Funktionalität**:
  - Alle Ratings eines Users ermitteln
  - Alle Ratings eines Items ermitteln
  - Gezielt ein einzelnes Rating ermitteln

  O(1)

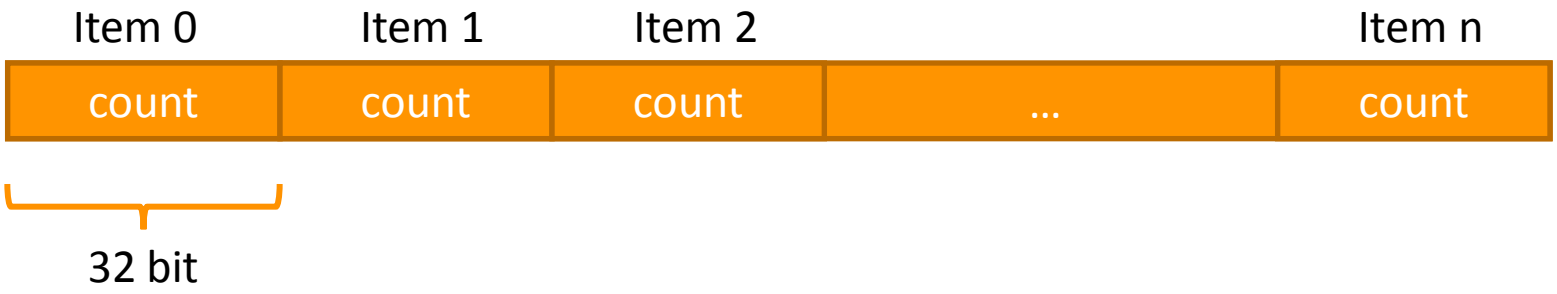
  O(log (N/U))
- Aber: größtenteils gefüllt mit **Nullwerten**
  - Nur 0,04% der Matrixelemente haben einen Wert
- **Idee**:
  - Speichere nur die tatsächlich ‚gefüllten' Elemente
  - Verwende Indizes für effizienten Zugriff

Datenformat

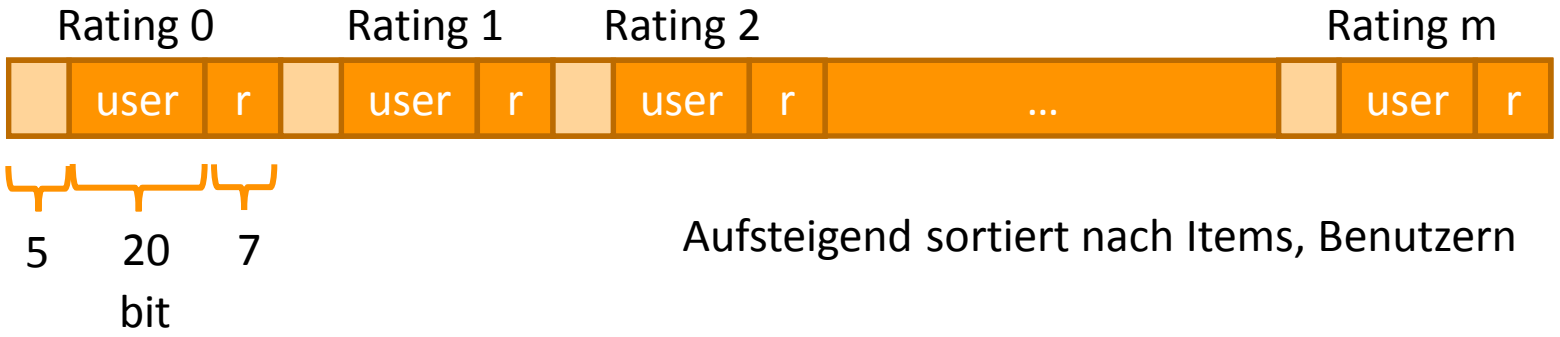


Matrix train_0 auf der Festplatte

train_0.idx
(2,4 MB)

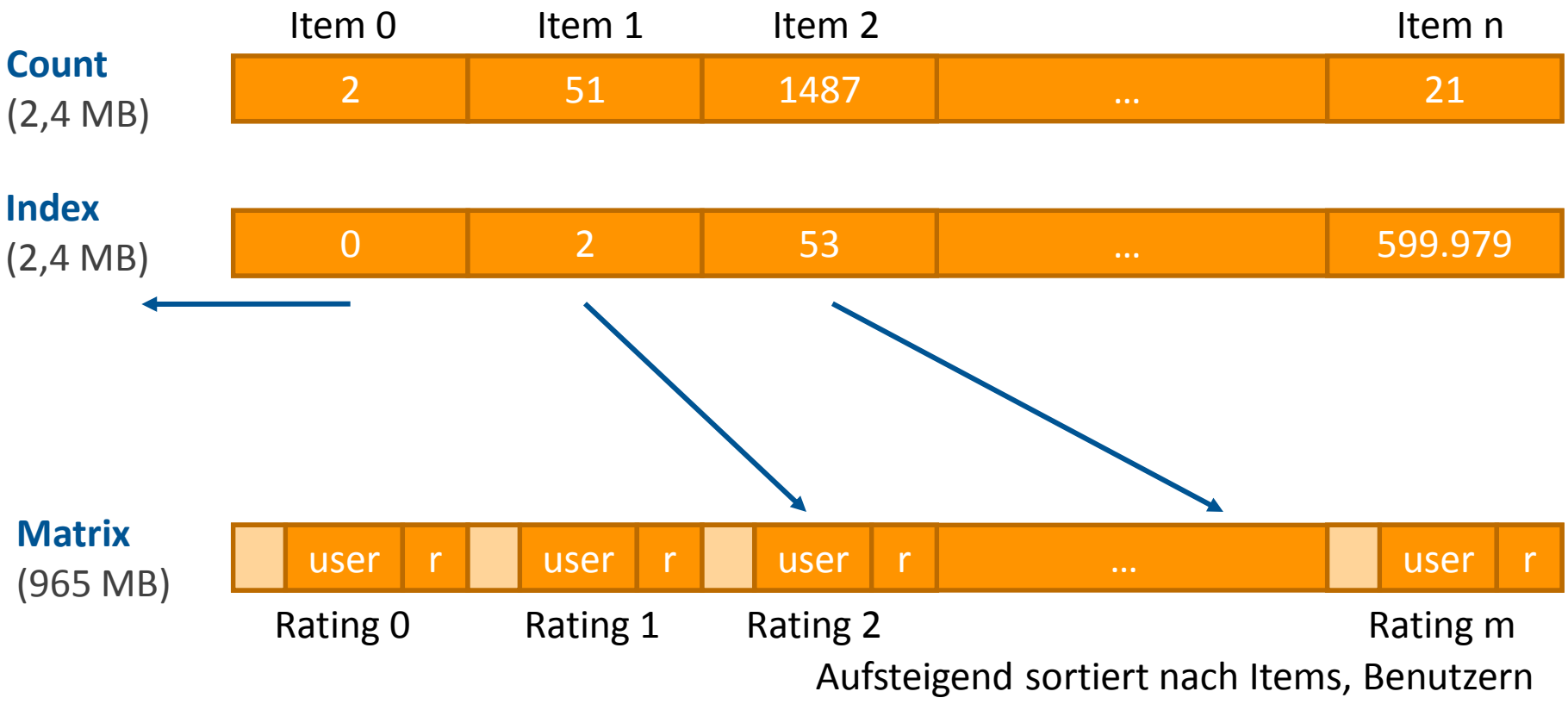


train_0.bin
(965 MB)



Aufsteigend sortiert nach Items, Benutzern

Matrix train_0 im Hauptspeicher



Benchmark

- Für **N = 10.000.000** zufällige User
  - Ermittle jeweils alle Ratings des Users
  - Für jedes Rating:
    - Lade alle Ratings des bewerteten Films
  - Für jeden User: ein Byte als Ergebnis auf die Festplatte schreiben
- Lenovo Thinkpad, Intel Core i7, 4 GB RAM
- **Laufzeit:** 05:03min davon 00:23 Laden der Matrizen

Sampling

1. Wähle zufällig N User
2. Suche alle von den Usern bewerteten Items (Validation+Training)
3. Suche alle für die Items relevanten Ratings/User (Training)

→ Neues Trainingsset = Trainingsset nur mit relevanten Ratings und Usern, die mind. ein relevantes Rating abgegeben haben (+ Durchschnittsbewertung der User)

→ Neues Validationset = Validationset nur mit den N Usern.

Projektplan

Implementierung

Ideen

Kombination mehrerer Verfahren

“Predictive accuracy is substantially improved when blending multiple predictors. Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a single technique. Consequently, our solution is an ensemble of many methods.“

[Bell, R. M., Koren Y. and Volinsky C.:
The BellKor solution to the Netflix Prize, 2007]

Kombination mehrerer Verfahren

- Ermittlung **verschiedener Prognosen** für das gesuchte Rating, z. B. für ein Album
  - Durchschnitt der Bewertungen des Users für alle im Album enthaltenen **Titel**
  - Durchschnittliche Bewertung aller gewerteten **Genres** des Albums
  - Bewertung des Künstlers
  - Durchschnittliche Bewertung aller **Alben** des gleichen Künstlers
  - **Wahrscheinlichkeitsverteilung** der Bewertung des jeweiligen Users
  - Bewertung durch **item-based CF** Algorithmus
  - …
- **Shrinkage** auf globalen Durchschnitt

Kombination mehrerer Verfahren

- Unterteilung der Items in verschiedene **Gruppen**
  - Z. B. nach **Typ** (Song, Album, Artist, …) und **Support**

- Für jede Gruppe:
  - Bestimmung der optimalen Gewichte durch **lineare Regression**