INTELLIGENTE DATENANALYSE IN MATLAB

Collaborative Filtering

Literatur

- Benjamin Marlin: Collaborative Filtering – A Machine Learning Perspective.
- Thomas Hofmann: Collaborative Filtering with Privacy via Factor Analysis.
- Robert Bell et. al: Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights.
- Markus Weimer at al.: CoFiRank.

Überblick

- Schritte der Datenanalyse:

Daten-vorverarbeitung → Problemanalyse → Problemlösung → Anwendung der Lösung

Daten-vorverarbeitung	Problemanalyse	Problemlösung	Anwendung der Lösung
Aggregation und Selektion von Daten. Integration und Säuberung der Daten. Feature-Extraktion.	Bestimmen von gegeb./gesuchten Größen. Wahl des Performanzmaß/ Zielkriteriums. Modellraum und Modellannahmen.	Algorithmen für das Optimieren des Zielkriteriums finden. Implementieren der Algorithmen.	Modell-Selektion & -Anpassung. Training & Evaluation des Modells auf gegebenen Daten. Vorhersage für neue Daten.

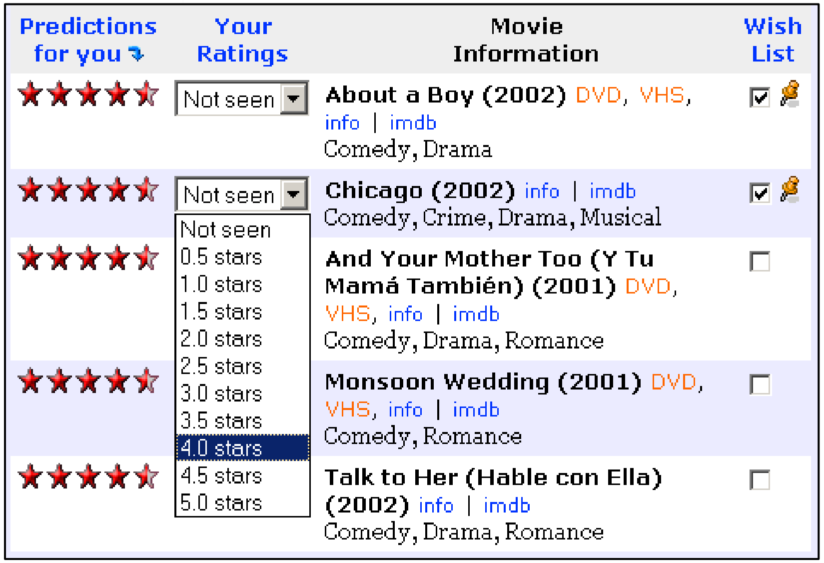
Problemstellung

- Gegeben: Trainingsdaten mit <u>mehreren</u> (teilweise) bekannten nutzerabhängigen Sortierungen.
- Eingabe: Menge von Instanzen (mit/ohne Attribute), Nutzer (mit/ohne Attribute) und nutzerspezifische Instanz-Bewertungen.
- Ausgabe: Sortierung der Instanzen <u>pro Nutzer</u>.
  - Collaborative Filtering: Nur Instanz-Bewertungen teilweise bekannt.
  - Content-based Filtering: Eigenschaften von Instanzen und Nutzern bekannt; Präferenzen teilweise bekannt.

Collaborative Filtering

Motivation

- Beispiel:
  - Empfehlungssysteme (*Recommendation Systems*) für Filme, Musik, Restaurants, Nachrichten, Bücher usw.



Film	Nutzer 1	Nutzer 2	Nutzer 3
About a Boy	5		4
Chicago	4	2	?
And Your M...			5
Monsoon Wedding	4	1	?
Talk to Her			4
Titanic			?
The Bourne Identity	2	2	?
SAW	1	5	?
Se7en		4	1
Earth			5
Stuart Little		1	?

Trainingsdaten

Zielgröße

Collaborative Filtering

Bewertungen: Erhebung

- Explizit:
  - Bewertung einer Instanz auf einer Skala bzw. mehreren Skalen (z.B. bzgl. Qualität, Preis-Leistungsverhältnis).
  - Textbeitrag (z.B. Buch-Rezension, Restaurant- & Hotel-beschreibung).
- Implizit:
  - Kauf eines Artikels auf einer Webseite.
  - Anklicken, Verweilen, Wechseln einer Webseite.
  - Standortinformationen bei der Auswahl (z.B. Telefon-, GPS- oder WLAN-Position).

Collaborative Filtering

Bewertungen: Eigenschaften

- Umfang der Daten:
  - Sehr wenig Bewertungen, z.B.
    - Einzelner Kunde eines Web-Shops bewertet nur Bruchteil des Angebotes.
    - Durchschnittliche „Besetzung“ einer Filmdatenbank beträgt nur ca. 2-5%.
  - Sehr viele Nutzer und Instanzen.

Skalierbarkeit bzgl. Laufzeit und Speicherverbrauch

- Qualität der Daten:
  - Nicht-zufälliges Fehlen von Bewertungen.
  - Unterschiedliche Verteilung der Scores verschiedener Benutzer.
  - Fehlerhafte/unfaire Bewertungen.
  - Zeitliche Abhängigkeiten von Bewertungen (selbstverstärkend).

Robustheit und Generalisierungsfähigkeit

Collaborative Filtering

Ansatz

- Gegeben: Bewertungsmatix **Y** wobei Instanz $i = 1 \ldots m$ von Nutzer $j = 1 \ldots n$ mit $Y_{ij} \in [Y_{\min}, Y_{\max}]$ bewertet wurde und zahlreiche Bewertungen Y_{ij} unbekannt sind.
- Gesucht: Rating Scores $R_{ij} \in \mathbb{R}$ für jeden Nutzer und jede Instanz.
- Ziel: Modell zur Vorhersage unbekannter Rating Scores.
  - Rating bzw. Ranking für bekannte Y_{ij} und zugehörige vorhergesagte Scores R_{ij} sollen möglichst gleich sein.
  - Hohe Generalisierungsfähigkeit/Robustheit.

Collaborative Filtering

Ansatz

- Modellierung als nutzerspezifische…
  - Rating-Vorhersage: Vorhersage des Scores (Klassifikations- bzw. Regressions-Problem).
  - Ranking-Vorhersage: Vorhersage der Reihenfolge (Ranking-Problem).
- Lösungsansätze:
  - *Ähnlichkeitsbasiert* (memory-based): Ähnliche Nutzer bewerten ähnlich und/oder ähnliche Instanzen werden ähnlich bewertet.
  - *Modellbasiert*: Jede Instanz hat versteckte Eigenschaften welche jeder Nutzer unterschiedlich bewertet; Eigenschaften und Bewertungen gleichzeitig aus Daten schätzen.

Ähnlichkeitsbasiertes Filtern

Nutzer-Nutzer-Ähnlichkeit

- Annahmen:
  - Ähnlichkeit U_{ij} zwischen Nutzer i und Nutzer j entspricht Ähnlichkeit gemeinsamer Bewertungen.

Film	Nutzer 1	Nutzer 2	Nutzer 3
About a Boy	5	3	4
Chicago	4	2	
And Your M...			5
Monsoon Wedding	4	1	4
Talk to Her	1		4

- Nächste-Nachbarn-Modell:
  - Unbekannte Bewertung = bekannte Bewertungen gewichtet mit Nutzer-Ähnlichkeit:

$$R_{ij} = \sum_{k \in B_j} \frac{U_{kj}}{\sum_{p \in B_j} U_{pj}} Y_{ik}$$

B_j = Indexmenge $\{k\}$ für welche Y_{kj} bekannt

	Nutzer 1	Nutzer 2	Nutzer 3
Nutzer 1	1,0	0,2	0,5
Nutzer 2	0,2	1,0	0,3
Nutzer 3	0,5	0,3	1,0

$$R_{23} = \frac{0,5 \cdot 4 + 0,3 \cdot 2}{0,5 + 0,3} = 3,25$$

Ähnlichkeitsbasiertes Filtern

Instanz-Instanz-Ähnlichkeit

- Annahmen:
  - Ähnlichkeit V_{ij} zwischen Instanz i und Instanz j entspricht Ähnlichkeit gemeinsamer Bewertungen.

Film	Nutzer 1	Nutzer 2	Nutzer 3
About a Boy	5	3	4
Chicago	4	2	
And Your M…			5
Monsoon Wedding	4	1	4
Talk to Her	1		4

- Nächste-Nachbarn-Modell:
  - Unbekannte Bewertung = bekannte Bewertungen gewichtet mit Instanz-Ähnlichkeit:

$$R_{ij} = \sum_{k \in B_i} \frac{V_{ik}}{\sum_{p \in B_i} V_{ip}} Y_{kj}$$

B_i = Indexmenge $\{k\}$ für welche Y_{ik} bekannt

	Film 1	Film 2	Film 3	Film 4	Film 5
Film 1	1,0	0,8	0,7	0,9	0,4
Film 2	0,8	1,0	0,1	0,9	0,2
Film 3	0,7	0,1	1,0	0,8	0,8
Film 4	0,9	0,9	0,8	1,0	0,6
Film 5	0,4	0,2	0,8	0,6	1,0

$$R_{23} = \frac{0{,}8 \cdot 4 + 0{,}1 \cdot 5 + 0{,}9 \cdot 4 + 0{,}2 \cdot 4}{0{,}8 + 0{,}1 + 0{,}9 + 0{,}2} = 4{,}05$$

Ähnlichkeitsbasiertes Filtern

Lösungsidee

- Allgemeiner Ansatz:
  - Bestimmung eines Kernels **K** für Nutzer-Nutzer- und/oder Instanz-Instanz-Ähnlichkeit.
  - Bestimmung von Gewichten **W**, so dass

$$R_{ij} = \sum_{k=1}^{n} K_{ik} W_{kj} \quad \Rightarrow \quad \mathbf{R} = \mathbf{KW}$$

  - Beispiel Nächste-Nachbarn-Modell (bspw. Instanz-Instanz):

$$R_{ij} = \sum_{k \in B_i} \left(\frac{V_{ik}}{\sum_{p \in B_i} V_{ip}} \right) Y_{kj}$$

Ähnlichkeit K_{ik} zwischen Instanz i und Instanz k

Gewichtungsfaktor W_{kj} für Instanz k und Nutzer j

Ähnlichkeitsbasiertes Filtern

Bestimmung des Ähnlichkeitskernel

- *1. Möglichkeit*: Empirische Korrelation (beispielhaft bzgl. Nutzern).

$$K_{ij} = \frac{1}{|B_{ij}|-1} \sum_{k \in B_{ij}} \frac{Y_{ik} - \mu_i}{\sigma_i} \frac{Y_{jk} - \mu_j}{\sigma_j}$$

B_{ij} = Indexmenge $\{k\}$ für welche Y_{ik} und Y_{jk} bekannt

mit mittlerem Nutzer-Rating (empirischer Mittelwert)

$$\mu_i = \frac{1}{|B_i|} \sum_{k \in B_i} Y_{ik}$$

B_i = Indexmenge $\{k\}$ für welche Y_{ik} bekannt

und mittlerer Abweichung (empirische Varianz).

$$\sigma_i^2 = \frac{1}{|B_i|-1} \sum_{k \in B_i} (Y_{ik} - \mu_i)^2$$

Ähnlichkeitsbasiertes Filtern

Bestimmung des Ähnlichkeitskernel

- 2. *Möglichkeit*: Für gegebene Gewichtematrix **W**, Ähnlichkeitsfaktoren aus Daten lernen.
- Minimierung einer (regularisierten) Verlustfunktion wie z.B. die mittlere quadratische Abweichung zwischen gegebenen und vorhergesagtem Rating:

Gesuchte Ähnlichkeiten K_{ik}

Echtes Rating Y_{ij}

$$\arg\min_{\mathbf{K}} \sum_{i=1}^{n} \sum_{j \in B_i} l\left(\sum_{k=1}^{n} K_{ik} W_{kj}, Y_{ij} \right) + \Omega(\mathbf{K})$$

Feste Gewichte W_{kj}

Regularisierer

Ähnlichkeitsbasiertes Filtern

Bestimmung der Gewichtematrix

- *1. Möglichkeit*: Nächste-Nachbarn-Ansatz bspw. mit Mittelwert-Korrektur (beispielhaft bzgl. Instanzen).

$$W_{ij} = \mu_j + \sum_{k \in B_i} \left(\frac{V_{ik}}{\sum_{p \in B_i} V_{ip}} \right) (Y_{kj} - \mu_k)$$

- *2. Möglichkeit*: Für gegebenen Ähnlichkeitskernel **K**, Gewichtematrix aus Daten lernen:

$$\arg\min_{\mathbf{W}} \sum_{i=1}^{n} \sum_{j \in B_i} l\left(\sum_{k=1}^{n} K_{ik} W_{kj}, Y_{ij} \right) + \Omega(\mathbf{W})$$

Feste Ähnlichkeiten K_{ik}

Gesuchte Gewichte W_{kj}

Ähnlichkeitsbasiertes Filtern
Bestimmung der Gewichtematrix

- Verlustfunktion:
  - Squared Loss (Regressions-Problem).
  - Relaxiertes NDCG-Loss (Ranking-Problem).
- Regularisierer:
  - Sparse Lösung bevorzugt, z.B. $\Omega(\mathbf{W}) = \sum_{i,j} |W_{ij}|$ bzw. $\Omega(\mathbf{K}) = \sum_{i,j} |K_{ij}|$.
- Kombination aus Instanz-Instanz- und Nutzer-Nutzer-Ähnlichkeit:
  - Annahme: Ähnliche Instanzen werden von ähnlichen Nutzern nahezu gleich bewertet.

Modellbasiertes Filtern

Annahmen

- Annahmen:
  - Jede Instanz (z.B. Film, Produkt) hat versteckte Eigenschaften, d.h. Vektor $\mathbf{x}_i$ ($i = 1 \ldots m$) mit k unbekannten Attribut-Belegungen.
  - Jeder Nutzer hat versteckte Interessen, d.h. Vektor $\mathbf{w}_j$ ($j = 1 \ldots n$) mit k unbekannten Attribut-Bewertungen (Nutzer-Profil).
  - Rating R_{ij} eines Nutzers j für Instanz i ist Skalarprodukt aus Attribut-Belegung $\mathbf{x}_i$ und Attribut-Bewertung $\mathbf{w}_j$: $R_{ij} = \mathbf{x}_i^{\mathrm{T}} \mathbf{w}_j$.
- Ziel:
  - Vektoren $\mathbf{x}_i$ und $\mathbf{w}_j$ für alle Instanzen und Nutzer aus Daten Schätzen sodass R_{ij} möglichst gleich den bekannten Y_{ij}.
  - Hohe Generalisierungsfähigkeit, d.h. Vorhersage basierend auf möglichst wenig Attributen (kleines k).

Modellbasiertes Filtern

Beispiel

- Beispiel für (versteckte) Attribut-Belegungen:

Film	Drama	Action	Dialoge
About a Boy	0,6	0,1	1,0
Chicago	0,7	0,2	0,7
Terminator	0,1	0,9	0,1
Matrix	0,2	1,0	0,3

- Beispiel für (versteckte) Attribut-Bewertungen:

Nutzer	Mag Drama	Mag Action	Mag Dialoge
Nutzer 1	0,5	4,5	1,0
Nutzer 2	3,5	0,0	2,5
Nutzer 3	1,5	2,0	1,0

- Beispiel-Ratings:

Film	Nutzer 1	Nutzer 2	Nutzer 3
About a Boy	1,75	4,60	2,10
Chicago	1,95	4,20	2,15
Terminator	4,20	0,60	2,05
Matrix	4,90	1,45	2,60

$$= 0.6 \cdot 1.5 + 0.1 \cdot 2.0 + 1.0 \cdot 1.0$$

Modellbasiertes Filtern

Lösungsidee

- Gleichzeitiges Schätzen von allen Vektoren $\mathbf{x}_i$ und $\mathbf{w}_i$

$$
\begin{array}{cc}
 & \begin{bmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_n \\ | & & | \end{bmatrix} \\
\begin{bmatrix} - & \mathbf{x}_1 & - \\ & \vdots & \\ - & \mathbf{x}_m & - \end{bmatrix} & \begin{bmatrix} R_{11} & \cdots & R_{1n} \\ \vdots & \ddots & \vdots \\ R_{m1} & \cdots & R_{mn} \end{bmatrix}
\end{array}
\Rightarrow \mathbf{X}^{\mathrm{T}}\mathbf{W} = \mathbf{R}
$$

Gesuchte Instanz-Attribute

Gesuchte Attribut-Bewertungen

mit dem Ziel den Abstand (Verlust) zwischen vorher-gesagtem Rating **R** und echtem Rating **Y** zu minimieren:

$$\arg\min_{\mathbf{X},\mathbf{W}} \sum_{i=1}^{m} \sum_{j \in B_i} l\left(\sum_{l=1}^{k} x_{il} w_{lj}, Y_{ij} \right) + \Omega(\mathbf{X}, \mathbf{W})$$

Modellbasiertes Filtern

Annahme: Alle Ratings bekannt

- Betrachten zunächst vereinfachende Annahmen:
  - Alle Ratings **Y** bekannt.
  - Squared Loss.
  - Keine Regularisierung.
- Optimale Lösung:

Matrix-Faktorisierung

$$\underset{\mathbf{X},\mathbf{W}}{\arg\min} \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{x}_i^{\mathrm{T}} \mathbf{w}_j - Y_{ij})^2 \quad \Rightarrow \quad \mathbf{X}^{\mathrm{T}} \mathbf{W} = \mathbf{Y}$$

- Optimierungsproblem besitz unendlich viele Lösungen!
- Eindeutige Lösung durch Regularisierung.

Modellbasiertes Filtern

Annahme: Alle Ratings bekannt

- Welchen Regularisierer verwenden?
  - Anzahl versteckter Attribute (Rang von **R**) soll klein sein.
- Optimale Lösung:

$$\mathbf{U} \underset{i=1\ldots\min(m,n)}{diag} (\sigma_i)\mathbf{V}^{\mathrm{T}} = \mathbf{Y} \quad \Rightarrow \quad \mathbf{X} = \mathbf{U}^{\mathrm{T}},\ \mathbf{W} = \underset{i=1\ldots k}{diag}(\sigma_i)\mathbf{V}^{\mathrm{T}} \quad \Rightarrow \quad \mathbf{R} = \mathbf{U}\underset{i=1\ldots k}{diag}(\sigma_i)\mathbf{V}^{\mathrm{T}}$$

Singulärwertzerlegung mit Singulärwerten σ_i

Rang von $\mathbf{R}$ ist k

- Spalte von **U** ist Koordinatenachse des Attribute-Raums.
- Zeile von **U** ist eine Instanz (z.B. Film) in diesem Raum.

Modellbasiertes Filtern

Annahme: Nicht alle Ratings bekannt

- Problem: Nicht alle Ratings bekannt.
- Idee: Minimiere Rang von x und w direkt.

$$\arg\min_{\mathbf{X},\mathbf{W}} \sum_{i=1}^{m} \sum_{j \in B_i} l\left(\sum_{l=1}^{k} x_{il} w_{lj}, Y_{ij}\right) + \frac{\lambda}{2}(rk(\mathbf{X}) + rk(\mathbf{W}))$$

  - Rang-Funktion ist nicht konvex!
  - Statt Minimierung des Rangs (Anzahl Singulärwerte $\neq 0$), Minimierung der Summe der quadrierten Singulärwerte = (quadrierte) Frobenius-Norm.

$$\arg\min_{\mathbf{X},\mathbf{W}} \sum_{i=1}^{m} \sum_{j \in B_i} l\left(\sum_{l=1}^{k} x_{il} w_{lj}, Y_{ij}\right) + \frac{\lambda}{2}\|\mathbf{X}\|_F^2 + \frac{\lambda}{2}\|\mathbf{W}\|_F^2$$

Modellbasiertes Filtern

Annahme: Nicht alle Ratings bekannt

- Für Frobenius-Norm einer Matrix gilt:

$$\|\mathbf{X}\|_F^2 = \sum_{i=1}^{\min(m,n)} \sigma_i^2 = tr(\mathbf{X}\mathbf{X}^{\mathrm{T}}) = \sum_{i=1}^{m} \|\mathbf{x}_i\|^2$$

- (Ein mögliches) konvexes Optimierungsproblem für modellbasiertes Collaborative Filtering:

$$\boxed{\arg\min_{\mathbf{x},\mathbf{w}} \sum_{i=1}^{m} \sum_{j \in B_i} l(\mathbf{x}_i^{\mathrm{T}} \mathbf{w}_j, Y_{ij}) + \frac{\lambda_x}{2} \sum_{i=1}^{m} \|\mathbf{x}_i\|^2 + \frac{\lambda_w}{2} \sum_{j=1}^{n} \|\mathbf{w}_j\|^2}$$

- Lösen beispielsweise durch abwechselnde Optimierung bzgl. x und w.

Modellbasiertes Filtern

l_2-Regularized Collaborative Filtering

- Algorithmus:

RegCoFilter(*Ratings* **Y**, *Anzahl versteckter Attribute k*)

Setze *l* = 0, $\forall j\ \mathbf{w}_j^0 = \mathbf{0}$ und wähle zufällig $\forall i\ \mathbf{x}_i^0 \in \mathbb{R}^k$

DO

FOR *j* = 1…*n*

$$\mathbf{w}_j^{l+1} = \arg\min_{\mathbf{w}_j} \sum_{i \in B_j} l(\mathbf{w}_j^{\mathrm{T}} \mathbf{x}_i^l, Y_{ij}) + \frac{\lambda_w}{2} \|\mathbf{w}_j\|^2$$

Lösen bspw. mit RegERM und Trainingsbeispielen $\{\mathbf{x}_i^l, Y_{ij} \mid i \in B_j\}$.

FOR *i* = 1…*m*

$$\mathbf{x}_i^{l+1} = \arg\min_{\mathbf{x}_i} \sum_{j \in B_i} l(\mathbf{x}_i^{\mathrm{T}} \mathbf{w}_j^{l+1}, Y_{ij}) + \frac{\lambda_x}{2} \|\mathbf{x}_i\|^2$$

Lösen bspw. mit RegERM und Trainingsbeispielen $\{\mathbf{w}_j^{l+1}, Y_{ij} \mid j \in B_i\}$.

$l = l + 1$

WHILE $\frac{1}{m} \sum_{i=1}^{m} \|\mathbf{x}_i^l - \mathbf{x}_i^{l-1}\| + \frac{1}{n} \sum_{j=1}^{n} \|\mathbf{w}_j^l - \mathbf{w}_j^{l-1}\| > \varepsilon$

$$\mathbf{R} = \begin{bmatrix} \mathbf{x}_1^l & \cdots & \mathbf{x}_m^l \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{w}_1^l & \cdots & \mathbf{w}_n^l \end{bmatrix}$$

RETURN **R**

Modellbasiertes Filtern

Erweiterungen

- Ranking-Loss verwenden (z.B. CoFiRank).
- Kombination mit Ähnlichkeitsbasierten Verfahren (z.B. Jointly Derived Neighborhood).
- Spezielle Regularisierer welche bspw. Ähnlichkeits-kernel berücksichtigen.
- Approximationen für hohe Skalierbarkeit.
- Normierung der gegebenen Daten (z.B. mittlere Bewertung jedes Nutzers auf 0 skalieren).

Zusammenfassung

- Ziel: Vorhersage von Nutzerbewertungen bzw. nutzerspezifischen Rankings.
  - Vorhersage basierend ausschließlich auf bekannten Bewertungen/Rankings anderer Nutzer.
- Ähnlichkeitsbasierter Ansatz:
  - Nächste-Nachbarn-Modell mit Ähnlichkeiten zweier Nutzer und/oder Instanzen.
- Modellbasierter Ansatz:
  - Versteckte Instanz-Attribute und nutzerspezifischen Attribut-Bewertungen gleichzeitig aus Daten schätzen.