Recommendation Systems I

Movies m

Nutzer n

	M1	M2	M3	M4
N1	5	3	–	1
N2	4	–	–	1
N3	1	1	–	5
N4	1	–	–	4
N5	–	1	5	4

z.B. Netflix, Movielens

Ratingmatrix $A = (a_{ij})$, $i = 1, \dots, n \leftarrow$ # Nutzer, $j = 1, \dots, m \leftarrow$ # Movies

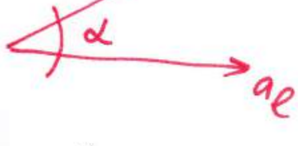
a_{ij}: Bewertung des j-ten Movie vom i-ten Nutzer

Frage: wie vervollständigt man die Ratingsmatrix A und gibt Empfehlungen ab?

① Ähnlichkeit von Nutzern / Movies

$$S(i,\ell) = \frac{\sum_{j \in \mathcal{U}(i) \cap \mathcal{U}(\ell)} a_{ij} \cdot a_{\ell j}}{\sqrt{\sum_{j \in \mathcal{U}(i) \cap \mathcal{U}(\ell)} a_{ij}^2} \cdot \sqrt{\sum_{j \in \mathcal{U}(i) \cap \mathcal{U}(\ell)} a_{\ell j}^2}} = \frac{a_i^T \cdot a_\ell}{\|a_i\| \cdot \|a_\ell\|}$$

$S(i,\ell)$: cosine similarity $= \cos\alpha$



$\mathcal{U}(i) := \{ j \mid a_{ij} \text{ existiert} \}$

j: Movies, für welche der i-te Nutzer Bewertung abgab.

Aber: $a_i \parallel a_\ell \Rightarrow \cos\alpha = 1$

↳ Mittelwert einbeziehen: $\bar{a}_i := \frac{1}{|\mathcal{U}(i)|} \sum_{j \in \mathcal{U}(i)} a_{ij}$

$|\mathcal{U}(i)|$ ← Elementenanzahl

$$S(i,\ell) = \frac{\sum_{j \in \mathcal{U}(i) \cap \mathcal{U}(\ell)} (a_{ij} - \bar{a}_i) \cdot (a_{\ell j} - \bar{a}_\ell)}{\sqrt{\sum_{j \in \mathcal{U}(i) \cap \mathcal{U}(\ell)} (a_{ij} - \bar{a}_i)^2} \cdot \sqrt{\sum_{j \in \mathcal{U}(i) \cap \mathcal{U}(\ell)} (a_{ij} - \bar{a}_i)^2}}$$

$S(i,\ell)$: Korrelation $= \frac{\operatorname{cov}(X_i, X_\ell)}{\sigma_{X_i} \cdot \sigma_{X_\ell}}$

k-Nearest Neighbor (kNN-Algorithm) $N_j(i) = \{$ k Nutzer mit größter Ähnlichkeit zu $i\}$
welche j bewertet haben

existiert

$$\underset{\text{geschätzt}}{a_{ij}} := \frac{\sum_{\ell \in N_j(i)} S(i,\ell) \cdot a_{\ell j}}{\sum_{\ell \in N(i)} |S(i,\ell)|}$$

- gewichtete Summe von Bewertungen aller k Nutzer, die dem Nutzer i am ähnlichsten sind.

Analog: Ähnlichkeit von Movies, da Nutzervektoren dünnbesetzt ("sparse") sind. ü

- wenig Aussagekraft, da dünnbesetzt
- suche nach verdeckten Mustern (z.B. Genre, Land, Schauspieler etc.)

② Singular Value Decomposition (SVD)
(Singulärwertzerlegung)

Frage: wie kann man Merkmale / Features identifizieren?
r

Nutzer bewerten (n) — charakterisieren Movies (m)

Sei zunächst A vollständig, d.h. alle Einträge seien vorhanden

$B := A^T \cdot A$ - symmetrisch

$m\overset{m}{\square} \quad m\overset{n}{\square}\; n\overset{m}{\square}$

- Eigenwerte und Eigenvektoren: $\quad \overbrace{B \cdot v_i = \lambda_i \cdot v_i}^{\text{Multiplikation}}$, $i = 1, \dots, r$ mit $\lambda_i \in \mathbb{R}$, $v_i \in \mathbb{R}^m$, mit $\lambda_i \neq 0$.

$B \cdot v_\ell = 0$, $\ell = r+1, \dots, m$

o.B.d.A. $\underbrace{\lambda_i \neq \lambda_j}_{\text{paarweise verschieden}}$ $\forall i,j = 1, \dots, r$

ⓤ $v_j^T \cdot B \cdot v_i = v_j^T \cdot \lambda_i \cdot v_i$
$\parallel$
$v_i^T \cdot B \cdot v_j = v_i^T \cdot \lambda_j \cdot v_j$

$\Rightarrow \quad v_i^T \cdot v_j = 0 \quad \Rightarrow$

- $v_1, \dots, v_m$ sind p.w. orthogonal (linear unabhängig)
- $\|v_i\|_2 = 1$ (o.B.d.A.) (normiert)

ⓥ $\underbrace{(Av_i)^T \cdot (Av_i)}_{\|Av_i\|_2^2} = v_i^T A^T A v_i = v_i^T \cdot B \cdot v_i = \lambda_i \underbrace{v_i^T v_i}_{=1} = \lambda_i \Rightarrow \lambda_i > 0$, $i = 1, \dots, r$ positiv

3

- **Rang:** $\underbrace{d_i}_{\neq 0} v_i = B \cdot v_i = A^T \cdot A \cdot v_i \Rightarrow v_i = A^T \left(A \cdot \frac{v_i}{d_i} \right) \in \text{Bild}(A^T)$, $i = 1, \dots, r$

$$\underset{v_i \text{ - l.u.}}{\Rightarrow} \quad \text{Rang}(A^T) = r \Rightarrow \text{Rang}(A) = r$$

Rang einer Matrix ist max. Anzahl l.u. Spalten/Zeilen

- **Orthogonalität**

$$\mathbb{R}^n \ni u_i := \frac{1}{\sqrt{d_i}} A \cdot v_i \ , \ i = 1, \dots, r \longleftarrow \text{orthogonal + normiert}$$

$$\Rightarrow u_i^T \cdot u_j = \frac{1}{\sqrt{d_i \cdot d_j}} (A v_i)^T \cdot A v_j = \frac{1}{\sqrt{d_i \cdot d_j}} v_i^T \cdot \underbrace{\underbrace{A^T \cdot A}_{=B} \cdot v_j}_{d_j \cdot v_j} = \sqrt{\frac{d_j}{d_i}} \cdot \underbrace{v_i^T \cdot v_j}_{= \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}}$$

für $i, j = 1, \dots, r$

- **Zerlegung:** "σ_i" - Singulärwerte

$$\sum_{i=1}^{r} \sqrt{d_i}\, u_i\, v_i^T = \sum_{i=1}^{r} A \cdot v_i \cdot v_i^T = \sum_{i=1}^{n} A \cdot v_i \cdot v_i^T = A$$

$$\| A v_e \|_2^2 = (A v_e)^T \cdot A v_e = v_e^T \underbrace{A^T \cdot A}_{B} \cdot v_e = 0 \Rightarrow A v_e = 0. \quad \underbrace{}_{=0} \quad e = r+1, \dots, m$$

$n \Big| \ 1 \overline{} m$

Rang 1 - Matrix $n \square^m$

$$\underset{n \times r}{\begin{bmatrix} | & & | \\ u_1 & \dots & u_r \\ | & & | \end{bmatrix}} \ \underset{r \times r}{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}} \ \underset{r \times m}{\begin{bmatrix} - v_1^T - \\ \vdots \\ - v_r^T - \end{bmatrix}} = \underset{n \times m}{\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}}$$

$$\mathcal{U} \cdot \Sigma \cdot V = A$$

(ii) • $\mathcal{U}^T \cdot \mathcal{U} = I_r$

• $V \cdot V^T = I_r$

Nutzer n × Feature r: u_{ik} — Bewertung des k-ten Feature vom i-ten Nutzer

Feature r × Feature r: σ_k — Gewichtung von Feature k

Feature r × Movie m: v_{kj} — Ausprägung des k-ten Feature in Movie j.

Ziel: Features nachahmen, wenn A nicht vollständig ist.
$\rightarrow$ Principal Component Analysis.