Recommender System - Graphen

Markus Deutschl

Ziel

- Recommender System
- Ähnlichkeiten von Filmen
- Graphenmodell
- Distanzfunktion
- Vorschläge anhand des Graphen

Graphdatenbanken

- Datenmodell: Graph
- Stark vernetzte Daten
- Flexible Schemata
- Praktische Abfragemöglichkeiten

Distanzberechnung

- Basis: Genres
- Modifikationen notwendig
- Rating
- Land
- Regisseur
- Jahr

Basisdistanz - Euklid

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

p …….. Vektor 1

q …….. Vektor 1

n …….. Dimensionsanzahl

Rating-Faktor

$$ratingFactor(m_1, m_2) = 1 + \left|\frac{rating(m_1) - rating(m_2)}{5}\right|$$

m_1 ..... Movie 1

m_2 ..... Movie 2

Ratingdistanz

$$ratingDistance(m_1, m_2) = 100 * (1 + \frac{euclid(m_1, m_2)}{100}) * ratingFactor(m_1, m_2)$$

Jahr-Faktor

$$yearFactor(m_1, m_2) = 1 + \left|\frac{year(m_1) - year(m_2)}{200}\right|$$

Regisseur-Faktor

$$directorMatch(m_1, m_2) = \begin{cases} 0{,}75 \; if \; same \; director \\ 0 \end{cases}$$

Land-Faktor

$$countryMatch(m_1, m_2) = \begin{cases} 0{,}25 \; if \; same \; country \\ 0 \end{cases}$$

Eigene Distanzfunktion

$$MovLibDistance(m_1, m_2) = \frac{100 * \left(1 + \frac{euclid(m_1, m_2)}{100}\right) * \ yearFactor(m_1, m_2)}{\left(1 + directorMatch(m_1, m_2) + countryMatch(m_1, m_2)\right) * \frac{1}{ratingFactor(m_1, m_2)}}$$

Verifikation

- Problem: Dimensionsanzahl
- 2D durch Berechnungen
- Lösung: Dimensionsreduktion
  - Embedding
  - PCA

Minimum Volume Embedding

- Embedding
- Dimensionsreduktion
- Effizienter als kPCA
- Sparse data

MVE-Algorithmus

Algorithm 1 Minimum Volume Embedding

Input: a Gram matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, sparse connectivity $\mathbf{A} \in \mathbb{B}^{n \times n}$, and parameters d, β, κ.

1: Initialize $\mathbf{K} \leftarrow \mathbf{W}$

2: $\mathcal{K} \leftarrow \{\mathbf{K} \succeq 0, \sum_{ij} K_{ij} = 0, K_{ii} + K_{jj} - 2K_{ij} = W_{ii} + W_{jj} - 2W_{ij} \ \forall_{i,j} \text{ s.t. } A_{ij} = 1\}$

3: **repeat**

4: Solve for the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and eigenvalues and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ of $\mathbf{K}$ using an SVD.

5: $\mathbf{B} \leftarrow \beta \sum_{i=1}^{d} \mathbf{v}_i \mathbf{v}_i^\top - \sum_{i=1}^{n} \mathbf{v}_i \mathbf{v}_i^\top$

6: $\mathbf{K} \leftarrow \hat{\mathbf{K}}$

7: $\hat{\mathbf{K}} \leftarrow \text{argmax}_{\mathbf{K} \in \mathcal{K}} \text{tr}(\mathbf{K}\mathbf{B})$ {Found via SDP}

8: **until** $\|\mathbf{K} - \hat{\mathbf{K}}\| \leq \kappa$

9: Perform SVD on $\hat{\mathbf{K}}$ to compute the d leading eigenvectors $\hat{\mathbf{v}}_i$ and corresponding eigenvalues $\hat{\lambda}_i$, and set $\mathbf{y}_i \leftarrow \sqrt{\hat{\lambda}_i} \hat{\mathbf{v}}_i$ for $i = 1, \ldots, d$

10: **return** $\mathbf{y}_1, \ldots, \mathbf{y}_d$

Nächste Schritte

- Implementierung in MatLab
- Visualisierung
- Anpassungen
- Metrik