Item-based Recommendation via Matrix Factorization and Green's Function (**Some ideas during my ongoing study**)

Wang Dingyan
CSE, CUHK

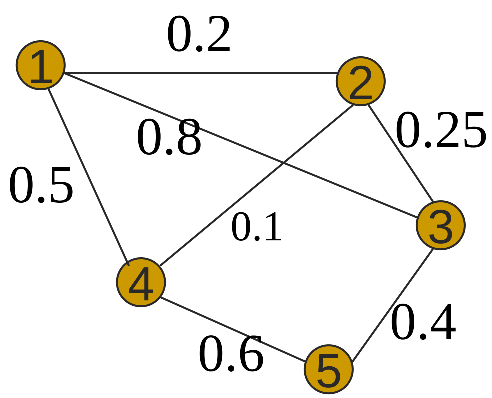
香港中文大學
The Chinese University of Hong Kong

Outline

- Previous Work
- Similarity Computation
- Matrix Factorization
- Motivations
- Recommendation using Green’s Function &Matrix Factorization
- Discussion

Previous Work

- Green's Function
  - Given a weighted graph G=(V,E),



$$W=\begin{pmatrix} 1 & 0.2 & 0.8 & 0.5 & 0 \\ 0.2 & 1 & 0.25 & 0.1 & 0 \\ 0.8 & 0.25 & 1 & 0 & 0.4 \\ 0.5 & 0.1 & 0 & 1 & 0.6 \\ 0 & 0 & 0.4 & 0.6 & 1 \end{pmatrix}$$

$$D=\begin{pmatrix} 2.5 & 0 & 0 & 0 & 0 \\ 0 & 1.55 & 0 & 0 & 0 \\ 0 & 0 & 2.45 & 0 & 0 \\ 0 & 0 & 0 & 2.2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

  - The Graph Laplacian matrix L= D-W.

Previous Work

- Green's Function
  - Defined as the inverse of L = D-W with zero-mode discarded.

$$Lv_k = \lambda_k v_k, \quad 0 = \lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$$

$$G^* = L_+^{-1} = \frac{1}{(D-W)_+} = \sum_{i=2}^{n} \frac{v_i v_i^T}{\lambda_i}$$

discard $\lambda_1 = 0$

Previous Work

- Item-based Recommendation
  - To calculate unknown rating by averaging rating of similar items by test users
  - User-item $M \times N$ matrix R,

    R_{pq} : u_p rates i_q
  - Item Graph G=(V,E)

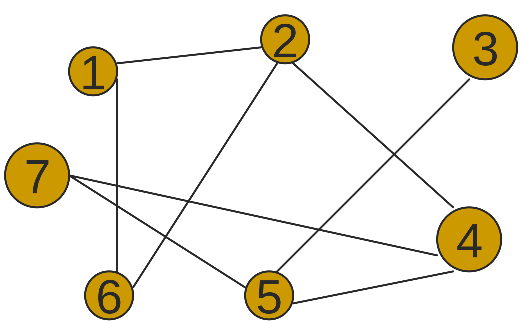
    typical similarity: *cosine similarity, PCC…*

Previous Work

- Recommendation with Green's Function

$$R_0 = \begin{pmatrix} 2 & 3 & 8 & 5 & 0 & 1 & 0 \\ 1 & 0 & 0 & 5 & 0 & 0 & 2 \\ 0 & 2 & 7 & 4 & 7 & 3 & 0 \\ 2 & 4 & 6 & 6 & 8 & 0 & 0 \\ 0 & 1 & 5 & 0 & 5 & 0 & 8 \\ 3 & 2 & 7 & 9 & 0 & 0 & 0 \\ 3 & 6 & 0 & 0 & 0 & 4 & 0 \\ 4 & 5 & 6 & 0 & 0 & 5 & 8 \end{pmatrix}$$

$$R^T = G R_0^{\ T}$$



$$G = \frac{1}{(D-W)_+}$$

Similarity Computation

- Similarity Computation :
  - *Cosine* similarity

$$Sim(i,j) = \frac{\| r_i \| \bullet \| r_j \|}{r_i \bullet r_j}$$

  - *Pearson Correlation Coefficient* (PCC)

$$Sim(i,j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i) \bullet (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i)^2} \bullet \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \bar{r}_j)^2}}$$

Similarity Computation

- Disadvantages:
  - Data Sparsity
  - One-sided: only rating-related

Matrix Factorization

- Matrix Factorization:
  - User-item matrix decomposition
  - Dimensionality reduction

$$A = UV$$

$$\widehat{r_{ij}} = \sum_{l=1}^{k} u_{il} v_{jl}$$

user-item matrix A: m*n

user factor matrix U: m*k

item factor matrix V: k*n

Matrix Factorization

- Matrix Factorization Methods:
  - RMF
  - MMMF
  - Non-negative MF
  - Probabilistic MF…

$$\min L(U,V)$$

Loss function : L(U,V)

Gradient descent algorithm

Matrix Factorization

- Matrix Factorization:
  - Solve the data sparsity problem
- Disadvantages:
  - Information Loss
    - Discard some matrix information

Motivation

- To construct a more accurate item graph
- To solve data sparsity problem
- To balance robustness and accuracy

Recommendation with Green’s Function & Matrix Factorization.

Recommendation with Green's Function & Matrix Factorization

- Given a user-item matrix ,
  then use Non-negative MF

$$R_0 = \begin{pmatrix} 2 & 3 & 8 & 5 & 0 & 1 & 0 \\ 1 & 0 & 0 & 5 & 0 & 0 & 2 \\ 0 & 2 & 7 & 4 & 7 & 3 & 0 \\ 2 & 4 & 6 & 6 & 8 & 0 & 0 \\ 0 & 1 & 5 & 0 & 5 & 0 & 8 \\ 3 & 2 & 7 & 9 & 0 & 0 & 0 \\ 3 & 6 & 0 & 0 & 0 & 4 & 0 \\ 4 & 5 & 6 & 0 & 0 & 5 & 8 \end{pmatrix} \Rightarrow R_0 \approx U_{8\times k} V_{k\times 7} \,(1 \le k < 8)$$

Recommendation with Green's Function & Matrix Factorization

- Similarity computation:

$$V = \begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & v_{ij} & \vdots \\ v_{k1} & \cdots & v_{kn} \end{pmatrix} \Rightarrow Sim_1(i, j) = \frac{\| v_i \| \bullet \| v_j \|}{v_i \bullet v_j}$$

A feature vector for item 1

Recommendation with Green's Function & Matrix Factorization

- Similarity computation:

$$Sim_2(i, j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i) \bullet (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \bar{r}_i)^2} \bullet \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \bar{r}_j)^2}}$$

$$Sim(i, j) = \lambda Sim_1(i, j) + (1 - \lambda) Sim_2(i, j)$$

$$\lambda \in [0,1]$$

Recommendation with Green's Function & Matrix Factorization

- Construct Graph:
  - According to Sim(i,j), construct a weighted item graph $G=(v,e)$
  - Given a threshold $\varepsilon (0<\varepsilon<1)$, if sim(I,j)< ε the two items are not connected.
  - W(i,j)=Sim(i,j)

Recommendation with Green's Function & Matrix Factorization

- Recommendation with Green's Function

$$G = \frac{1}{(D - W)_+}$$

$$\Downarrow$$

$$R^T = G R_0^{\ T}$$

Recommendation with Green's Function & Matrix Factorization

- Sorry, the experiment is on-going.
- Disadvantages:
  - Time consuming
  - Accumulated loss
- Extension:
  - Iterative training
  - Incorporate with additional information: social network, confidence level, implicit feedback…..

Discussion and Suggestion

Any Suggestion?

Any Inspiration?

Thank You!