Diffusion:
a new approach to recommender systems

Yi-Cheng Zhang, Matúš Medo, Jie Ren,
Tao Zhou, Tao Li, and Fan Yang

University of Fribourg Switzerland, Renmin University of China, USTC China

ECCS'07 Satellite Workshop
Dresden, October 4, 2007

The playground

- available data:
  - ratings from M users for N movies
  - integer scale: 1 (very bad), 2, 3, 4, 5 (perfect)

The playground

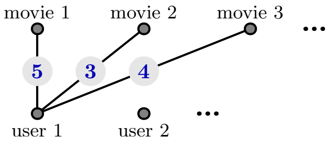
- available data:
  - ratings from M users for N movies
  - integer scale: 1 (very bad), 2, 3, 4, 5 (perfect)
- both M and N can be very large
  - thus we need an efficient recommendation method

The playground

- available data:
  - ratings from M users for N movies
  - integer scale: 1 (very bad), 2, 3, 4, 5 (perfect)
- both M and N can be very large
  - thus we need an efficient recommendation method
- prediction methods already available:
  - **average rating**
    (recommendation by overall “quality”)
  - **correlation based**
    (recommendation by the user-user similarity)
  - . . .

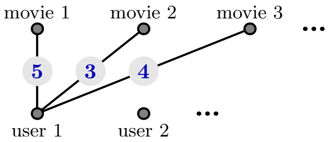
Representations of the data

- available ratings: weighted bipartite graph

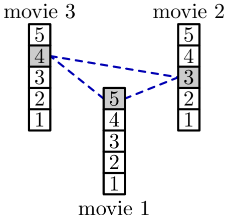


Representations of the data

- available ratings: weighted bipartite graph

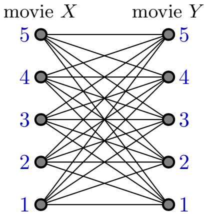


- movie-to-movie projection of the data



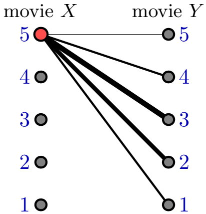
Diffusion in the movie-to-movie network

- opinions expressed by a particular user spread over the network



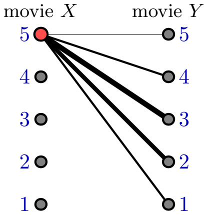
Diffusion in the movie-to-movie network

- opinions expressed by a particular user spread over the network



Diffusion in the movie-to-movie network

- opinions expressed by a particular user spread over the network



- summary:
  - **key process**: diffusion on a complex network
  - **underlying network**: movie-to-movie projection
  - **boundary condition**: ratings given by one user

Mathematical formalism

- the rating of user *i* for movie α we represent as a 5-dimensional vector

$$1 \rightarrow \boldsymbol{v}_{i\alpha} = \{1, 0, 0, 0, 0\}^{\mathsf{T}}, \ldots, \; 5 \rightarrow \boldsymbol{v}_{i\alpha} = \{0, 0, 0, 0, 1\}^{\mathsf{T}}$$

- the link between movies α and β is a 5×5 matrix

$$\mathsf{W}_{\alpha\beta} = \sum_{i=1}^{M} \lambda_i \boldsymbol{v}_{i\alpha} \boldsymbol{v}_{i\beta}^{\mathsf{T}}$$

- here λ_i is the weight of each contribution of user *i*

Mathematical formalism

- the rating of user i for movie α we represent as a 5-dimensional vector

$$1 \rightarrow \boldsymbol{v}_{i\alpha} = \{1,0,0,0,0\}^{\mathsf{T}}, \ldots, \; 5 \rightarrow \boldsymbol{v}_{i\alpha} = \{0,0,0,0,1\}^{\mathsf{T}}$$

- the link between movies α and β is a 5×5 matrix

$$\mathsf{W}_{\alpha\beta} = \sum_{i=1}^{M} \lambda_i \boldsymbol{v}_{i\alpha} \boldsymbol{v}_{i\beta}^{\mathsf{T}}$$

- here λ_i is the weight of each contribution of user i
  - for a user who has rated k_i movies we use $\lambda_i = 1/(k_i - 1)$

The diffusion process

- matrices $W_{\alpha\beta}$ form the overall symmetric matrix W ($5N \times 5N$)
- by the column normalization of W we obtain Ω which describes the diffusion process on the movie-to-movie network

The diffusion process

- matrices $\mathrm{W}_{\alpha\beta}$ form the overall symmetric matrix W ($5N \times 5N$)
- by the column normalization of W we obtain Ω which describes the diffusion process on the movie-to-movie network
- Ω acts on the $5N$-dimensional "state" vector $\boldsymbol{h}$
  - elements 1-5 correspond to movie 1
  - elements 6-10 correspond to movie 2
  - ...

Boundary conditions

- for a given user we set $\boldsymbol{h}_0$ according to the expressed ratings

$$\boldsymbol{h}_0 = \{\underbrace{1,0,0,0,0}_{\text{1 for movie 1}},\underbrace{0,0,0,1,0}_{\text{4 for movie 2}},\dots\}^{\mathsf{T}}.$$

  - if the rating has not been given yet, we set only zeros for this movie

- then we iterate using the equation

$$\boldsymbol{h}_{n+1} = \hat{\Omega}\boldsymbol{h}_n$$

  - $\hat{\Omega}$ differs from Ω in the fact that it **preserves expressed opinions** of the given user

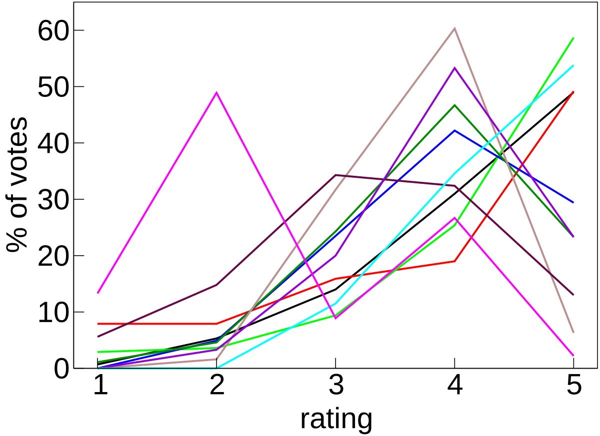
Interpretation of the results

- after $n_{\max}$ iterations we stop
- e.g. for the movie X (not rated yet by the given user) we obtain $\{0.1, 0.1, 0.4, 0.3, 0.0\}^{\mathrm{T}}$
- the standard weighted average transforms to a scalar value

$$\hat{v} = \frac{0.1 \times 1 + 0.1 \times 2 + 0.4 \times 3 + 0.3 \times 4}{0.1 + 0.1 + 0.3 + 0.4} = \mathbf{3.0}$$

  - this is the prediction for movie X

The polarization problem



Solution to the polarization problem

- we unify the ratings by the linear transformation

$$v_{i\alpha} \rightarrow M_i + (v_{i\alpha} - \mu_i)\,\frac{S_i}{\sigma_i}$$

  - μ_i and σ_i is the avearage rating and the standard deviation of ratings for user i
  - M_i and S_i is the avearage rating and the standard deviation of ratings for the whole society
(only movies rated by user i taken into account)

Summary of the algorithm

$W_{\alpha\beta}$

Summary of the algorithm

$$\mathrm{W}_{\alpha\beta}$$
$$\downarrow$$
$$\mathrm{W}$$

Summary of the algorithm

$$W_{\alpha\beta}$$
$$\downarrow$$
$$W$$
$$\downarrow$$
$$\Omega$$

Summary of the algorithm

$$\mathsf{W}_{\alpha\beta}$$
$$\downarrow$$
$$\mathsf{W}$$
$$\downarrow$$
$$\Omega$$
$$\downarrow$$
$$\hat{\Omega}$$

Summary of the algorithm

$$\mathsf{W}_{\alpha\beta}$$
$$\downarrow$$
$$\mathsf{W}$$
$$\downarrow$$
$$\Omega$$
$$\downarrow$$
$$\hat{\Omega}$$
$$\downarrow$$
$$\hat{\Omega}^n \boldsymbol{h}_0$$

Summary of the algorithm

unification

$$\downarrow$$

$$W_{\alpha\beta}$$

$$\downarrow$$

$$W$$

$$\downarrow$$

$$\Omega$$

$$\downarrow$$

$$\hat{\Omega}$$

$$\downarrow$$

$$\hat{\Omega}^n \boldsymbol{h}_0$$

$$\downarrow$$

repersonalization

Numerical tests

- data of GroupLens project (`www.grouplens.org`)
    - 943 users, 1682 movies, 100 000 ratings
    - sparsity around 6%

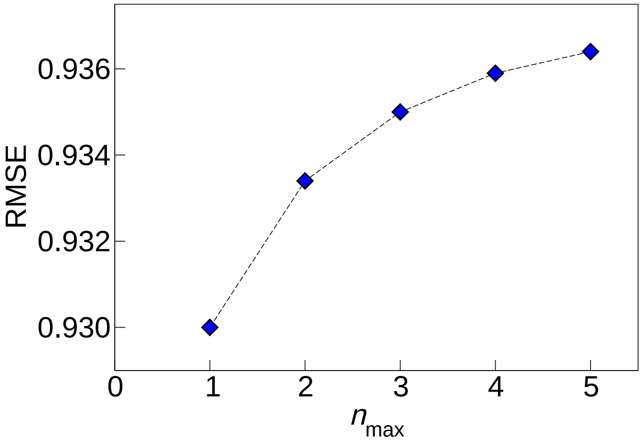
Numerical tests

- data of GroupLens project (`www.grouplens.org`)
  - 943 users, 1682 movies, 100 000 ratings
  - sparsity around 6%
- randomly chosen $n = 10\,000$ ratings we move to probe $\mathcal{P}$
- the remaining 90 000 ratings we use to predict the deleted
- two standard error measures of the prediction performance

$$\mathrm{RMSE} = \left[\frac{1}{n}\sum_{(i,\alpha)\in\mathcal{P}}(v_{i\alpha} - \hat{v}_{i\alpha})^2\right]^{1/2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{(i,\alpha)\in\mathcal{P}}|v_{i\alpha} - \hat{v}_{i\alpha}|$$

Performance against the number of iterations



Comparison with standard methods

RMSE values for the tested methods

method	no unification	with unification
movie average	1.18	1.01
correlation-based	1.09	1.09
diffusion-based	1.00	0.93

Comparison with standard methods

RMSE values for the tested methods

method	no unification	with unification
movie average	1.18	1.01
correlation-based	1.09	1.09
diffusion-based	1.00	0.93

- observations:
  - movie average is the simplest method but with unification it performs well

Comparison with standard methods

RMSE values for the tested methods

method	no unification	with unification
movie average	1.18	1.01
correlation-based	1.09	1.09
diffusion-based	1.00	0.93

- observations:
  - movie average is the simplest method but with unification it performs well
  - substraction of the user average μ_i is a part of correlation-based methods and thus unification does not improve performance here

Comparison with standard methods

RMSE values for the tested methods

method	no unification	with unification
movie average	1.18	1.01
correlation-based	1.09	1.09
diffusion-based	1.00	0.93

- observations:
  - movie average is the simplest method but with unification it performs well
  - substraction of the user average μ_i is a part of correlation-based methods and thus unification does not improve performance here
  - the diffusion-based method outperforms the other two

Thank You for Your attention