TSINGHUA UNIVERSITY

Online Dating Recommender Systems: The Split-complex Number Approach

Jérôme Kunegis, Gerd Gröner, Thomas Gottron
WeST – Institute for Web Science and Technologies
University of Koblenz–Landau

Present by *Feng Xie*

October 29, 2012

Outline

- Background
- Dating Recommendation Algorithm
- Evaluation
- Conclusion

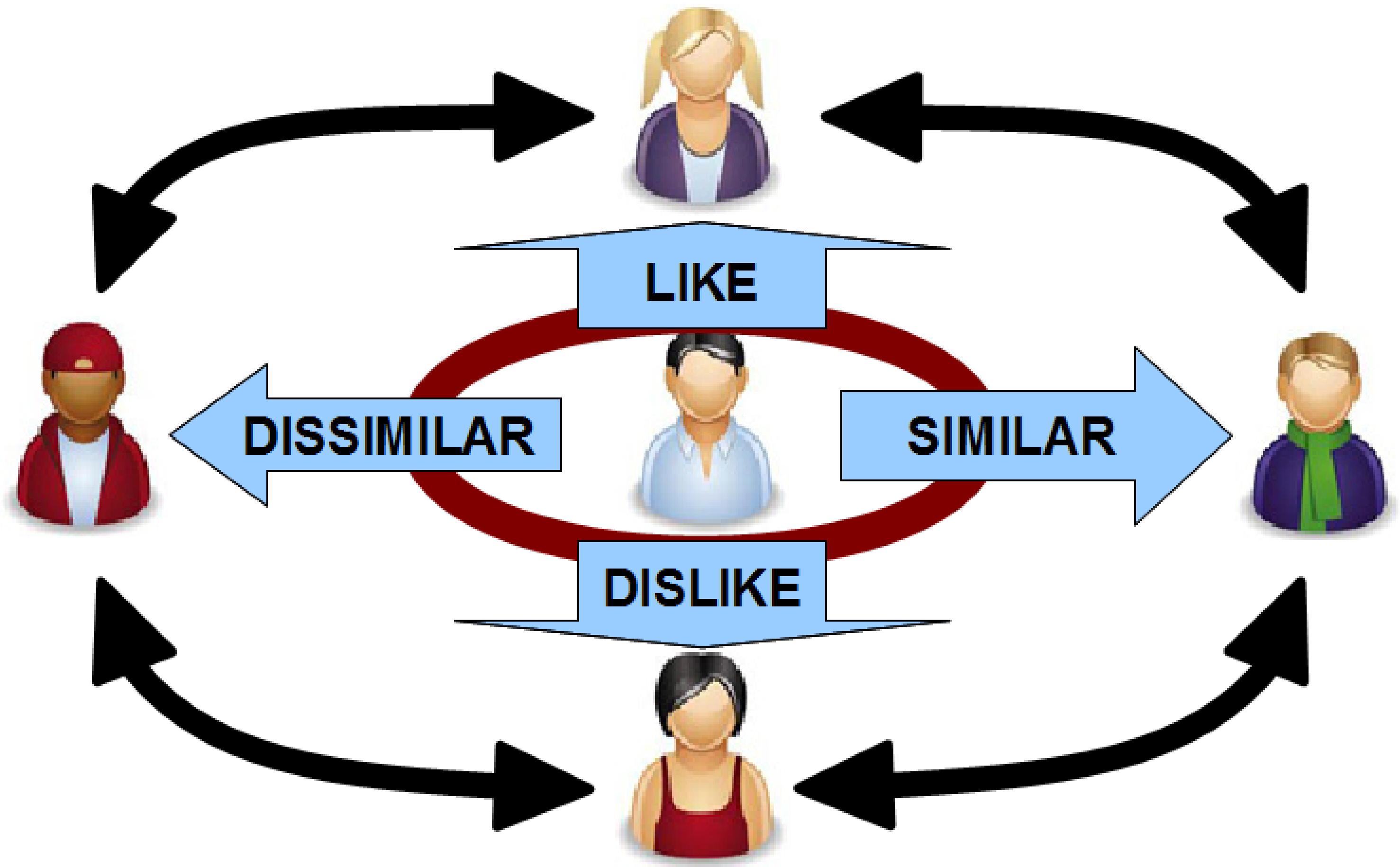
Recommendation Scenarios

- Asymmetric
  - Objects to user
- Symmetric
  - Users to user

Friend Recommendation

- Pure Friendship
  - similar: from your hometown, school, employer and more
- Dating
  - taste: interested in other gender
  - not recommend women to women nor men to men

Dating Recommendation

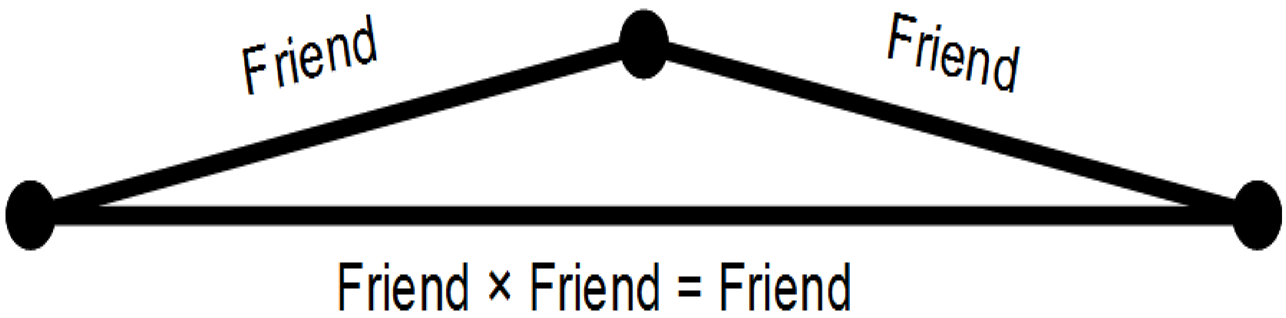


Contributions

- Contribution 1: Reducing the problem of dating recommendations to a link prediction problem
- Contribution 2: Split-complex number is used to model the particularities of online dating sites
- Contribution 3: Distinguish between the like/dislike relationships and the similar/dissimilar relationships

Link Prediction in Friend Recommendation

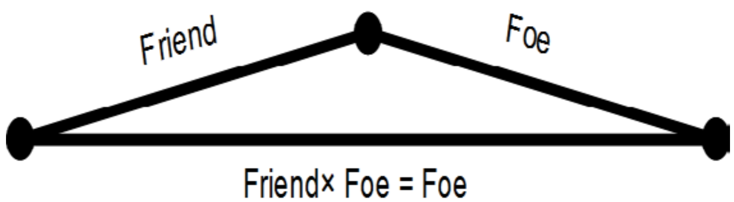
- Link Prediction
  - People are nodes, the known friendships are edges
  - Predict new links or lost links

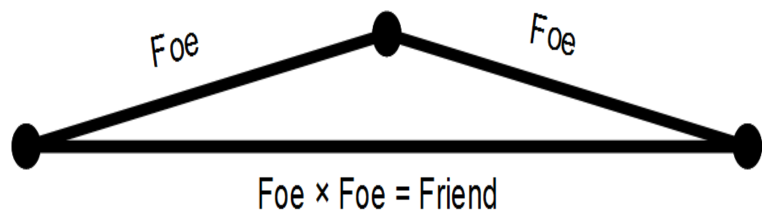


- Friends' friends are friends
- Only one relationship type
- All predicted edges are this type

Link Prediction in Friend and Enmity Recommendation

- People can tag each other as friends and foes





- Friends’ foes are foes, and foes’ foes are friends

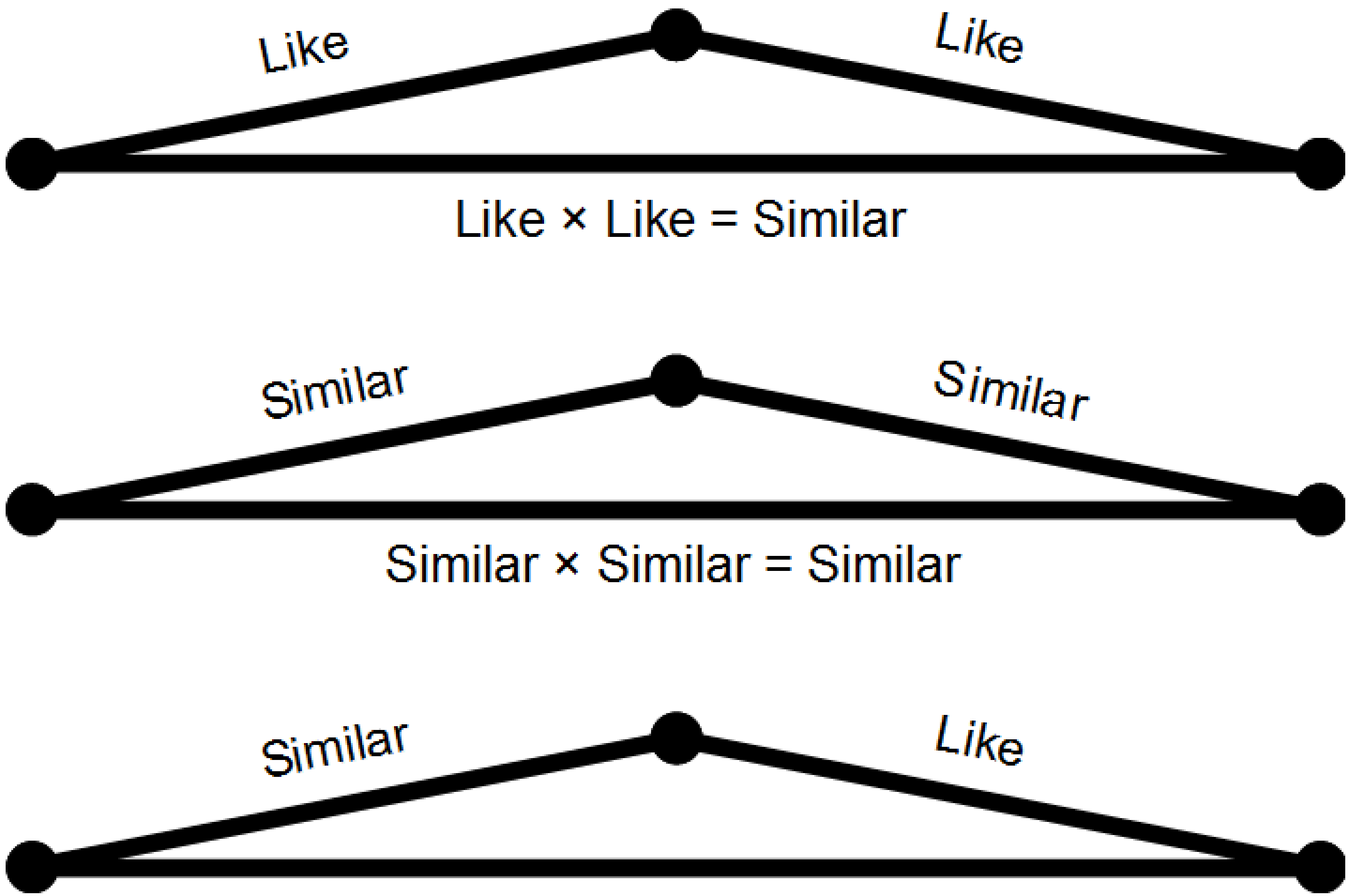
$$\text{Friend} = +1 \qquad \text{Foe} = -1$$

$$+1 \times +1 = +1$$
$$-1 \times +1 = -1$$
$$-1 \times -1 = +1$$

Link Prediction in Dating Recommendation

- e_{like} for *like* friendship, $e_{similar}$ for *similar* friendship
  - $e_{like} \cdot e_{like} = e_{similar}$
  - $e_{similar} \cdot e_{similar} = e_{similar}$
  - $e_{like} \cdot e_{similar} = e_{like}$
- A trivial solution $e_{like} = e_{similar} = 1$
- We want the relationships *like* and *similar* to be different
- A perfect solution $e_{similar} = 1$ and $e_{like} = j$, where $j^2=1$
- Analogous multiplication rules with *dislike* and *dissimilar* can then be derived by multiplying both sides with -1

Link Prediction in Dating Recommendation

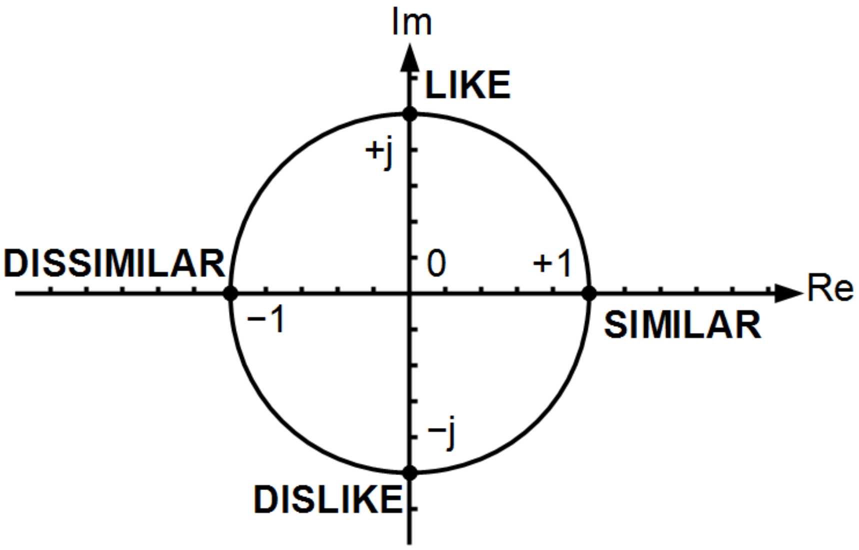


Split-complex Numbers

- Split-complex number is different from the complex number, every split-complex number has the form:

$$z = a + bj$$

$$j^2 = 1$$



Split-complex Numbers

$z = a + bj$

$j \times j = +1$

$(a + bj) + (c + dj) = (a + c) + (b + d)j$

$(a + bj) \times (c + dj) = (ac + bd) + (ad + bc)j$

Not a field: $(1 + j)(1 - j) = 0$

Dating Recommendation Algorithm

- Algebraic Graph Theory
  - Let G=(V, E) be an unweighted and undirected network, Its adjacency matrix can be defined as $\mathrm{A} \in \Re^{|V| \times |V|}$

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{when } \{i,j\} \in E, \\ 0 & \text{when } \{i,j\} \notin E. \end{cases}$$

  - The number of common neighbor of nodes i and j is given by the square of the adjacency matrix:

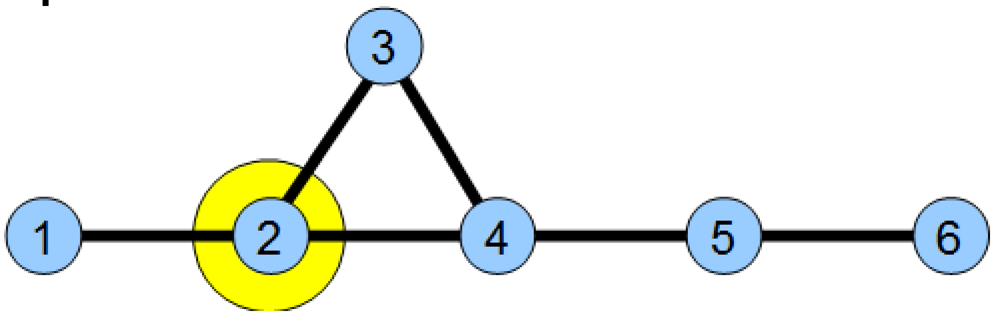
$$\mathrm{CN}(i,j) = (\mathbf{A}^2)_{ij}$$

  - Equivalently, the number of common neighbors of i and j can be interpreted as the number of paths of length two between i and j.

$$(\mathbf{A}^k)_{ij}$$

Dating Recommendation Algorithm

- An example



$$\mathbf{A} = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 0 \\ 4 & 0 & 1 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 0 & 1 & 0 & 1 \\ 6 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

$\mathbf{A}_{uv}$ = 1 when u and v are connected

$\mathbf{A}_{uv}$ = 0 when u and v are not connected

Dating Recommendation Algorithm

- The Powers of the adjacency matrix A can be used to implement a social recommender based on triangle closing:
  - A^2 implements the basic triangle closing recommender
  - Higher powers generalize A^k triangle closing to the closing of longer paths
- A possible combination

$$\exp(\mathbf{A}) = \mathbf{I} + \mathbf{A} + \frac{1}{2}\mathbf{A}^2 + \frac{1}{6}\mathbf{A}^3 + \ldots$$

Dating Recommendation Algorithm

- Multiple relationship types: *friend* and *foe*
  - The weights of edges are +1 or -1
  - The adjacency matrix is defined as:

$$\mathbf{A}_{ij} = \begin{cases} +1 & \text{when } \{i,j\} \in E \text{ and } \sigma(\{i,j\}) = +1, \\ -1 & \text{when } \{i,j\} \in E \text{ and } \sigma(\{i,j\}) = -1, \\ 0 & \text{when } \{i,j\} \notin E. \end{cases}$$

  - The powers A^k has the same preference
  - "The enemy of my enemy is my friend"
  - The matrix exponential of exp(A) can be used as a recommendation algorithm in networks with positive and negative edges

Dating Recommendation Algorithm

- Multiple relationship types: *like*, *dislike*, *similar* and *dissimilar*
  - Let $G=(V,E,w)$ be the directed and signed rating network
  - Each edge $(i, j) \in E$ is given a weight by the weight function w, with $w((i,j))$
  - $e_{like} = j$, $e_{dislike} = -j$

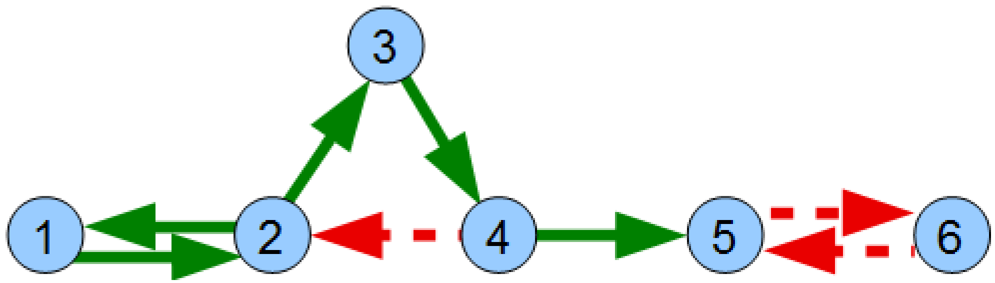
$$\mathbf{A}_{ij} = \begin{cases} w((i,j)) & \text{when } (i,j) \in E, \\ 0 & \text{when } (i,j) \notin E. \end{cases}$$

  - The split-complex adjacency matrix is $A_s = jA$
  - $A_s^k = j^k A^k$

$$j^k = \begin{cases} 1 & \text{when } n \text{ is even} \\ j & \text{when } n \text{ is odd} \end{cases}$$

Dating Recommendation Algorithm

- An example



$\mathbf{B}_{uv}$ = +j when u likes v

$\mathbf{B}_{uv}$ = −j when u dislikes v

$\mathbf{B}_{uv}$ = 0 when u and v are not connected

$$\mathbf{B} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left|\begin{array}{cccccc} & +j & & & & \\ +j & & +j & & & \\ & & & +j & & \\ & -j & & & +j & \\ & & & & & -j \\ & & & & -j & \end{array}\right| \end{array}$$

B = j**A**

Dating Recommendation Algorithm

- Any sun of the powers of A_s

$$
\begin{aligned}
p(\mathbf{A}_s) &= a\mathbf{I} + b\jmath\mathbf{A} + c\mathbf{A}^2 + d\jmath\mathbf{A}^3 + \ldots \\
&= (a\mathbf{I} + c\mathbf{A}^2 + \ldots) + \jmath(b\mathbf{A} + d\mathbf{A}^3 + \ldots).
\end{aligned}
$$

- We thus see that the even part of the power sum can be used to find similar persons, and the odd part to find liked persons

Dating Recommendation Algorithm

- Split-complex numbers as 2×2 matrices

$$a + bj \equiv \begin{vmatrix} a & b \\ b & a \end{vmatrix}$$

$$1 \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad j \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

$$\begin{vmatrix} a & b \\ b & a \end{vmatrix} + \begin{vmatrix} c & d \\ d & c \end{vmatrix} = \begin{vmatrix} a+c & b+d \\ b+d & a+c \end{vmatrix}$$

$$\begin{vmatrix} a & b \\ b & a \end{vmatrix} \times \begin{vmatrix} c & d \\ d & c \end{vmatrix} = \begin{vmatrix} ac+bd & ad+bc \\ ad+bc & ac+bd \end{vmatrix}$$

- A_s can be represented by the real matrix A_b

$$\mathbf{A}_s \equiv \mathbf{A}_b = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix}$$

$$\mathbf{A}_b^{2k} = \begin{bmatrix} (\mathbf{A}\mathbf{A}^T)^k & \mathbf{0} \\ \mathbf{0} & (\mathbf{A}^T\mathbf{A})^k \end{bmatrix}$$

$$\mathbf{A}_b^{2k+1} = \begin{bmatrix} \mathbf{0} & (\mathbf{A}\mathbf{A}^T)^k\mathbf{A} \\ (\mathbf{A}^T\mathbf{A})^k\mathbf{A}^T & \mathbf{0} \end{bmatrix}$$

Dating Recommendation Algorithm

- Representation as the bipartite double cover

- If the graph G has the real adjacency matrix A, then its bipartite double cover has the adjacency matrix

$$\mathrm{A_b} = \begin{bmatrix} 0 & \mathrm{A} \\ \mathrm{A^T} & 0 \end{bmatrix}$$

- This shows that counting paths in a graph with edge weights $\pm j$ is equivalent to counting paths in the bipartite double cover of the graph with edge weights ± 1

Dating Recommendation Algorithm

- Dating Recommender Functions
  - *Polynomials*

$$p(\mathbf{A}) = a\mathbf{A} + b\mathbf{A}^3 + c\mathbf{A}^5 + \dots$$

  - *Hyperbolic sine*

$$\sinh(\mathbf{A}) = \mathbf{A} + \frac{1}{6}\mathbf{A}^3 + \frac{1}{120}\mathbf{A}^5 + \dots$$

  - *Newman kernel*

$$(\mathbf{I} - \alpha\mathbf{A})^{-1} = \mathbf{I} + \alpha\mathbf{A} + \alpha^2\mathbf{A}^2 + \alpha^3\mathbf{A}^3 + \dots$$

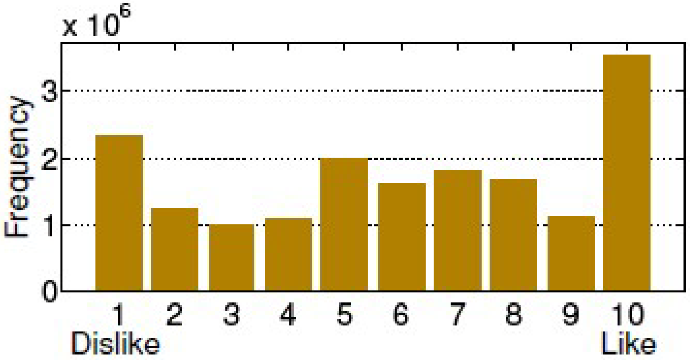
$$\alpha\mathbf{A}(\mathbf{I} - \alpha^2\mathbf{A}^2)^{-1} = \alpha\mathbf{A} + \alpha^3\mathbf{A}^3 + \alpha^5\mathbf{A} + \dots$$

Evaluation

- Dataset
  - Libimseti.cz (“Do you like me”-Czech dating site)
  - Directed network, edges represent ratings on a scale from 1 (dislike) to 10 (like)

Table 1: Demographics and distribution of ratings in Libimseti.cz.

Gender	Count	Rating counts Unknown	Male	Female	Total
Unknown	83,164	366,180	891,550	445,115	1,702,845
Male	76,441	937,684	682,710	3,232,064	4,852,458
Female	61,365	2,460,765	7,099,688	1,243,590	10,804,043
All	220,970	3,764,629	8,673,948	4,920,769	17,359,346



Evaluation

- Converting the dataset to positive and negative values
  - Overall mean rating

$$\mu = |E|^{-1} \sum_{(i,j)\in E} r_{ij}$$

  - The real adjacency matrix

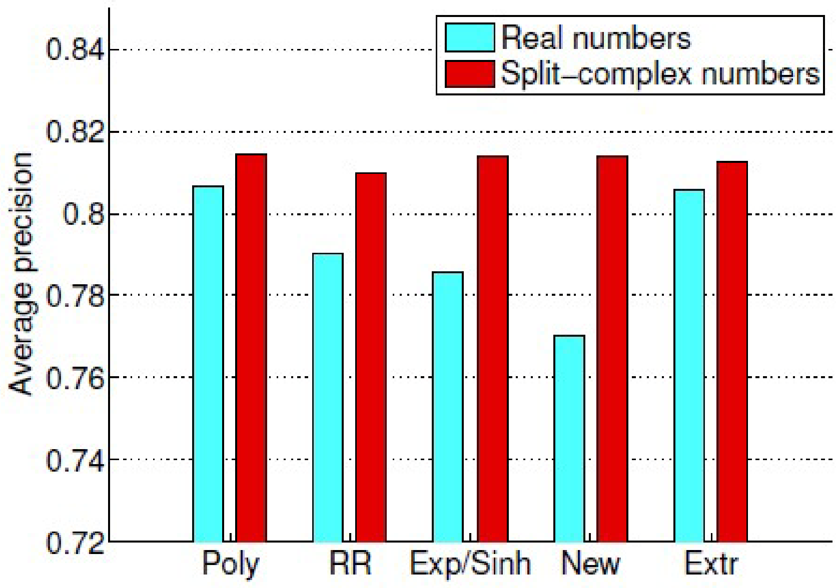
$$\mathbf{A}_{ij} = \begin{cases} r_{ij} - \mu & \text{when } (i,j) \in E, \\ 0 & \text{when } (i,j) \notin E. \end{cases}$$

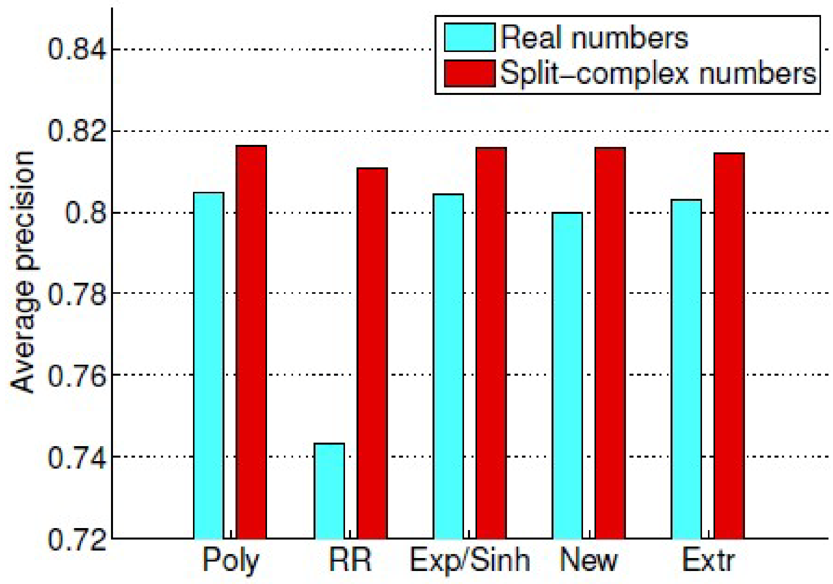
Evaluation

- Test set: random set containing 25% of all edges
- Training set: the rest of the network
- Comparison methods
  - (Poly) The best nonnegative polynomial
  - (RR) Rank reduction
  - (Exp/Sinh) The matrix exponential and hyperbolic sine
  - (New) The (odd) Newman kernel
  - (Extr) Spectral extrapolation

Evaluation

- Experimental method
  - Recommendation with Unknown Gender
  - Recommendation with Known Gender

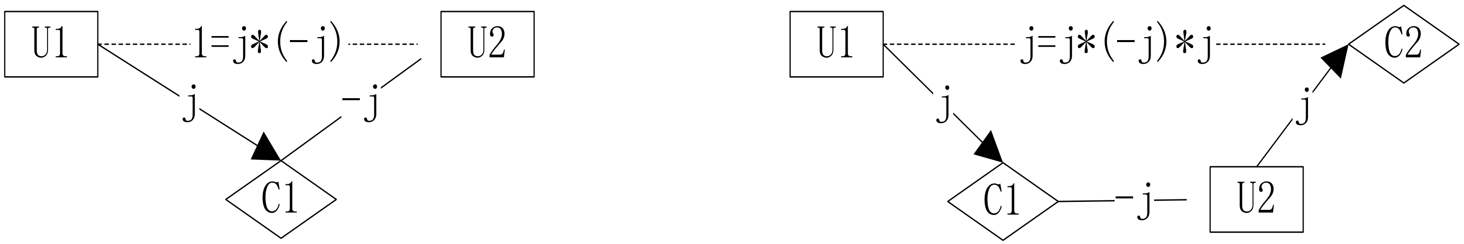




Conclusion

- For all recommender systems, our model based on the split-complex numbers performs better than the model using only real numbers. This validates our model that distinguishes the relationship types *like*, *dislike*, *similar* and *dissimilar*.
- The description of our method has been restricted to the case of heterosexual relationships, i.e., men liking women and women liking men. However, the method can in fact be generalized to homosexual relationships: Our first experiment in which we ignored the gender of the rater showed that the method is independent of genders, and thus can be generalized to any types of sexual relationships.

Discussion



(a) (b)

$$A = \begin{array}{c c} & \begin{array}{cc} U & \quad C \end{array} \\ \begin{array}{c} U \\ C \end{array} & \begin{bmatrix} 0 & A_x \\ -A_x{}^T & 0 \end{bmatrix} \end{array} = \begin{bmatrix} 0 & jA_b \\ -jA_b{}^T & 0 \end{bmatrix}$$

Discussion

$$A^{2k} = \begin{bmatrix} \left(A_b A_b{}^T\right)^k & 0 \\ 0 & \left(A_b{}^T A_b\right)^k \end{bmatrix}$$

$$A^{2k+1} = j\begin{bmatrix} 0 & \left(A_b A_b{}^T\right)^k A_b \\ \left(A_b{}^T A_b\right)^k A_b{}^T & 0 \end{bmatrix}$$

$$\exp(A) = I + A + \frac{1}{2}A^2 + \frac{1}{6}A^3 + \cdots$$

Q&A

Thank you!