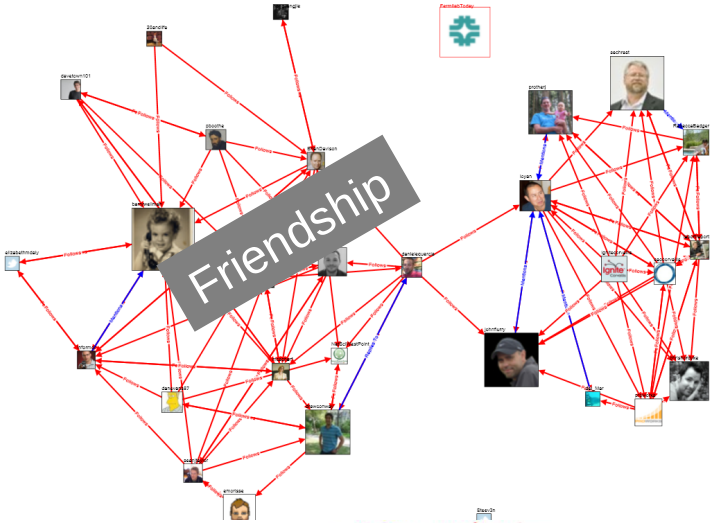
On the Spectral Evolution of Large Networks

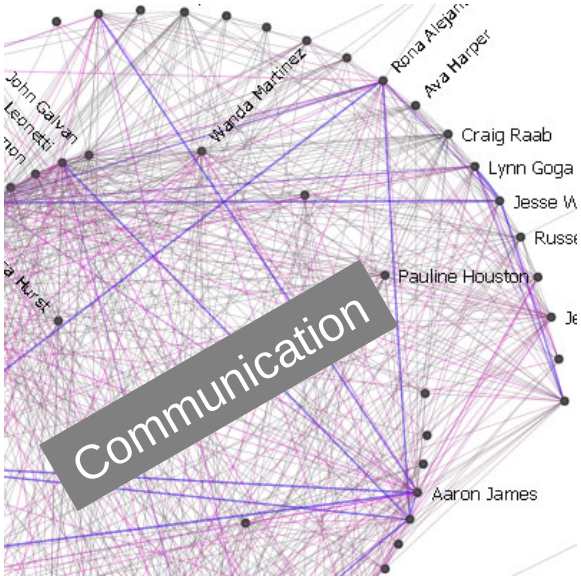
Jérôme Kunegis

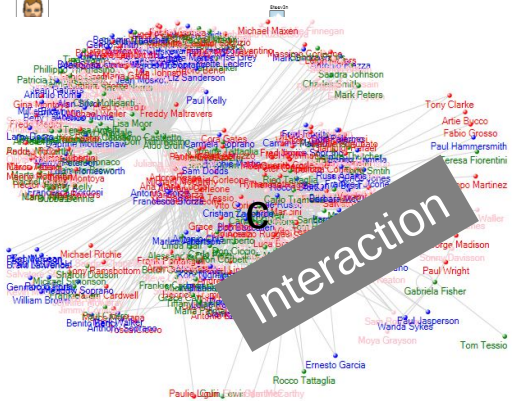
Committee: Prof. Dr. Staab
Prof. Dr. Bauckhage
Prof. Dr. Obermayer

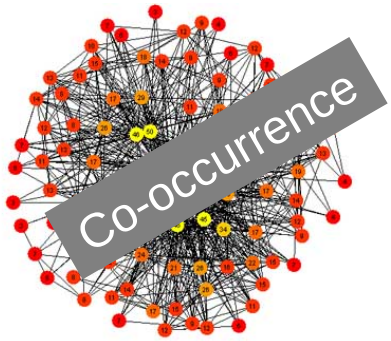
Networks

…are everywhere



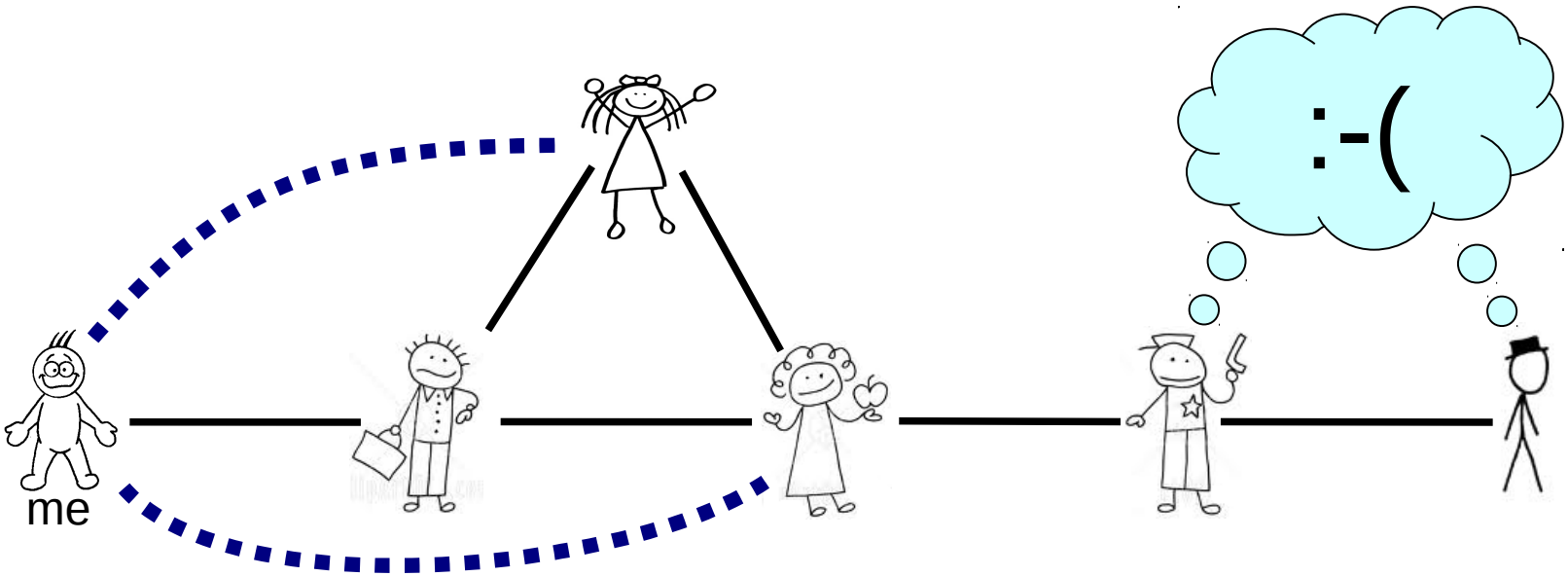






How can we exploit the information contained in them?

Use Networks for Recommendation



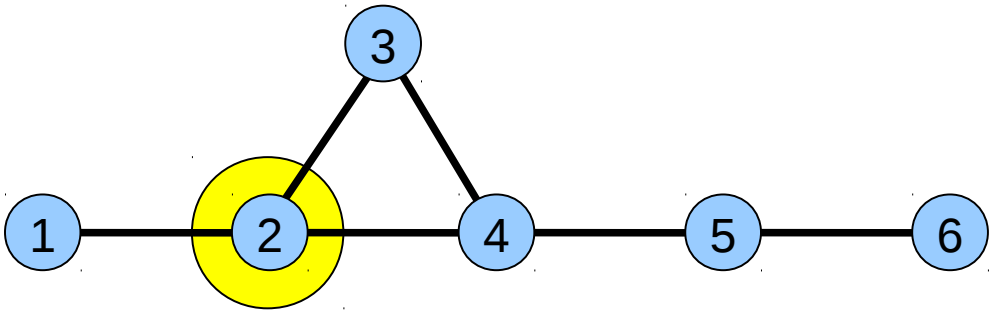
- Idea: Predict who a person will connect with
- Facebook's algorithm: find friends-of-friends

→ Problem: Rest of the network is ignored!

1. **Algebraic Link Prediction**
2. Spectral Transformations
3. Learning Link Prediction
4. Special Cases

Take into account the whole network

Algebraic Graph Theory



Represent a network
by an adjacency matrix **A**:

$\mathbf{A}_{ij}$ = 1 when *i* and *j* are connected

$\mathbf{A}_{ij}$ = 0 when *i* and *j* are not connected

A is square and symmetric.

A =

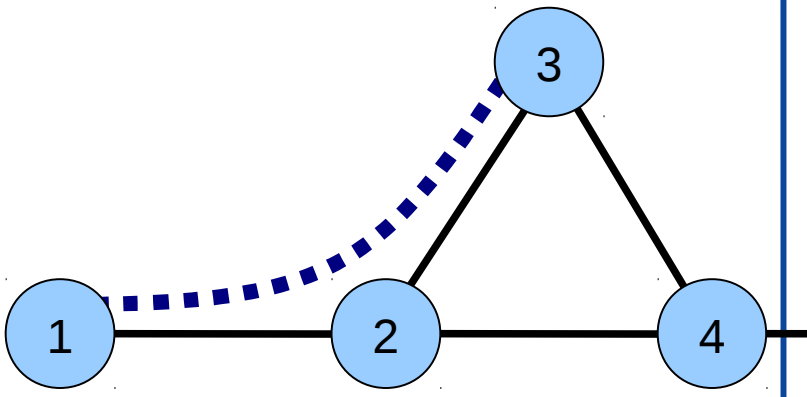
	1	2	3	4	5	6
1	0	1	0	0	0	0
2	1	0	1	1	0	0
3	0	1	0	1	0	0
4	0	1	1	0	1	0
5	0	0	0	1	0	1
6	0	0	0	0	1	0

Baseline: Friend of a Friend Model

Count the number of ways a person can be found as the friend of a friend.

Consider the matrix product $\mathbf{A}\,\mathbf{A} = \mathbf{A}^2$

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}^2 = \begin{vmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 3 & 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 1 & 1 & 0 \\ 1 & 1 & 1 & 3 & 0 & 1 \\ 0 & 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{vmatrix}$$



Eigenvalue Decomposition

Write the matrix **A** as a product:

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T$$

where

U are the eigenvectors $\quad \mathbf{U}^T\mathbf{U} = \mathbf{I}$

Λ are the eigenvalues $\quad \mathbf{\Lambda}_{ij} = 0$ when $i \neq j$

Computing A^2

Use the eigenvalue decomposition $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T$

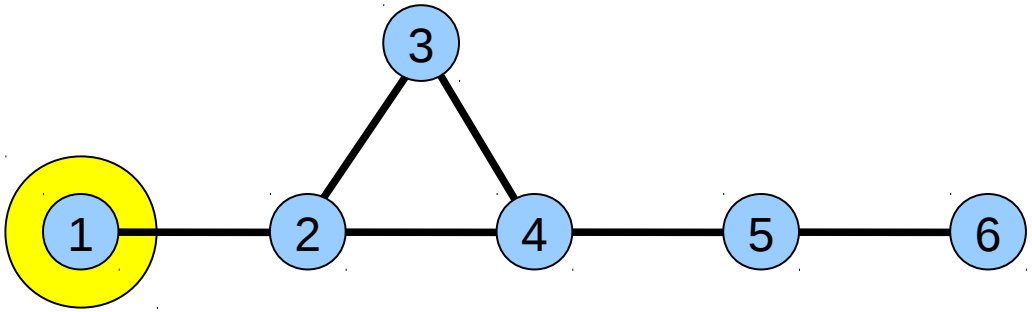
$$\mathbf{A}^2 = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T\,\mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T = \mathbf{U}\,\mathbf{\Lambda}^2\,\mathbf{U}^T$$

Exploit $\mathbf{U}$ and $\mathbf{\Lambda}$:

- $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ because $\mathbf{U}$ contains eigenvectors
- $(\mathbf{\Lambda}^2)_{ii} = \mathbf{\Lambda}_{ii}^2$ because $\mathbf{\Lambda}$ contains eigenvalues

Result: Just square all eigenvalues!

Friend of a Friend of a Friend



Compute the number of friends-of-friends-of-friends:

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}^3 = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{vmatrix} 0 & 3 & 1 & 1 & 1 & 0 \\ 3 & 2 & 4 & 5 & 1 & 1 \\ 1 & 4 & 2 & 4 & 1 & 1 \\ 1 & 5 & 4 & 2 & 4 & 0 \\ 1 & 1 & 1 & 4 & 0 & 2 \\ 0 & 1 & 1 & 0 & 2 & 0 \end{vmatrix} \end{array} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

$$\mathbf{A^3 = U \Lambda U^T U \Lambda U^T U \Lambda U^T = U \Lambda^3 U^T}$$

Power Sums

$\mathbf{A}^n$

= Number of paths of length n

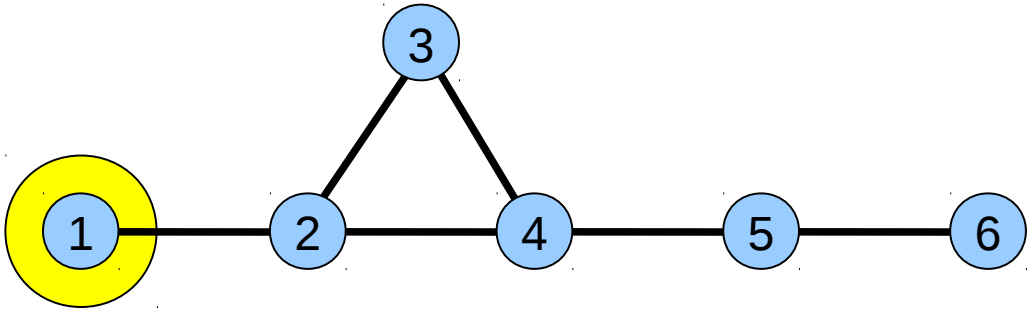
$a\,\mathbf{A}^2 + b\,\mathbf{A}^3 + c\,\mathbf{A}^4 + \ldots$

= Number of paths, weighted by path length

→ New edges more likely to appear when there are **many paths** already

→ When $a > b > c > \cdots > 0$, **short paths** are weighted more

Matrix Exponential



The matrix exponential can be written as a power sum with decreasing coefficients:

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

$$\exp \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix} = \begin{array}{c} \begin{array}{cccccc} ① & ② & ③ & ④ & ⑤ & ⑥ \end{array} \\ \begin{vmatrix} 1.66 & 1.72 & 0.93 & 0.98 & 0.28 & 0.06 \\ 1.72 & 3.57 & 2.70 & 2.92 & 1.04 & 0.28 \\ 0.93 & 2.70 & 2.86 & 2.71 & 0.99 & 0.27 \\ 0.98 & 2.93 & 2.71 & 3.62 & 1.94 & 0.71 \\ 0.28 & 1.04 & 0.99 & 1.94 & 2.31 & 1.39 \\ 0.06 & 0.28 & 0.27 & 0.71 & 1.39 & 1.59 \end{vmatrix} \begin{array}{c} ① \\ ② \\ ③ \\ ④ \\ ⑤ \\ ⑥ \end{array} \end{array}$$

Example: To ①, recommend ④, ③, ⑤, then ⑥

Computing Power Sums

Let p(**A**) be a power sum:

$$\begin{aligned} p(\mathbf{A}) &= a\,\mathbf{A}^2 + b\,\mathbf{A}^3 + c\,\mathbf{A}^4 + \cdots \\ &= a\,\mathbf{U}\,\mathbf{\Lambda}^2\,\mathbf{U}^T + b\,\mathbf{U}\,\mathbf{\Lambda}^3\,\mathbf{U}^T + c\,\mathbf{U}\,\mathbf{\Lambda}^4\,\mathbf{U}^T + \cdots \\ &= \mathbf{U}\,(a\,\mathbf{\Lambda}^2 + b\,\mathbf{\Lambda}^3 + c\,\mathbf{\Lambda}^4 + \cdots)\,\mathbf{U}^T \\ &= \mathbf{U}\,p(\mathbf{\Lambda})\,\mathbf{U}^T \end{aligned}$$

Therefore:

Power sums change only the eigenvalues!

Outline

1. Algebraic Link Prediction
2. **Spectral Transformations**
3. Learning Link Prediction
4. Special Cases

Why does it work?

Spectral Transformations

$\mathbf{A}^2 = \mathbf{U}\,\boldsymbol{\Lambda}^2\,\mathbf{U}^T$ Friend of a friend

$\mathbf{A}^3 = \mathbf{U}\,\boldsymbol{\Lambda}^3\,\mathbf{U}^T$ Friend of a friend of a friend

$\exp(\mathbf{A}) = \mathbf{U}\,\exp(\boldsymbol{\Lambda})\,\mathbf{U}^T$ Matrix exponential

…are link prediction functions!

Looking at Real Facebook Data

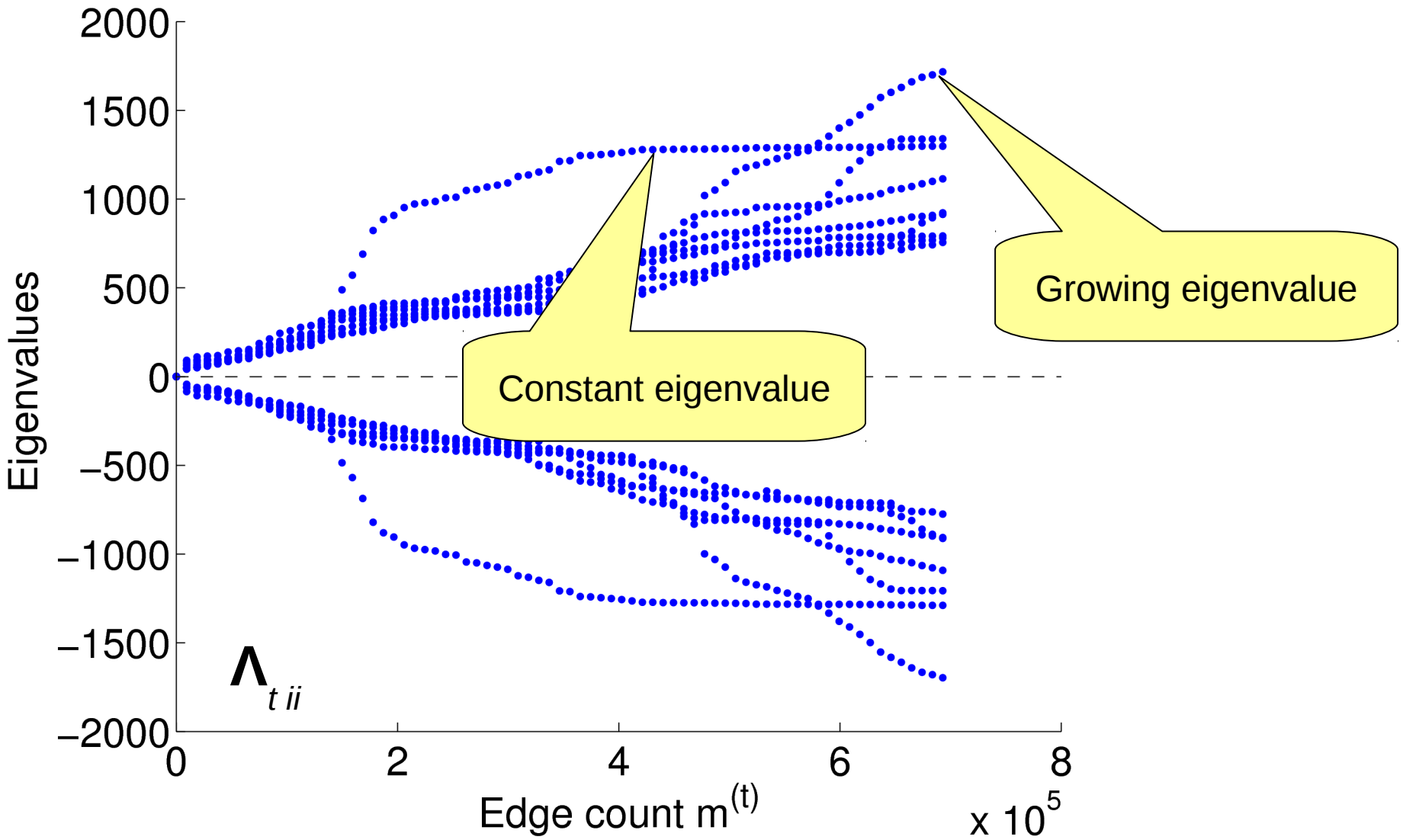
Dataset: Facebook New Orleans (Viswanath 2009)

63,731 persons
1,545,686 friendship links with **creation dates**

Let $\mathbf{A}_t$ be the adjacency matrix at time t ($1 \leq t \leq n = 75$)
$\mathbf{A}_t$ contains all edges formed before time t

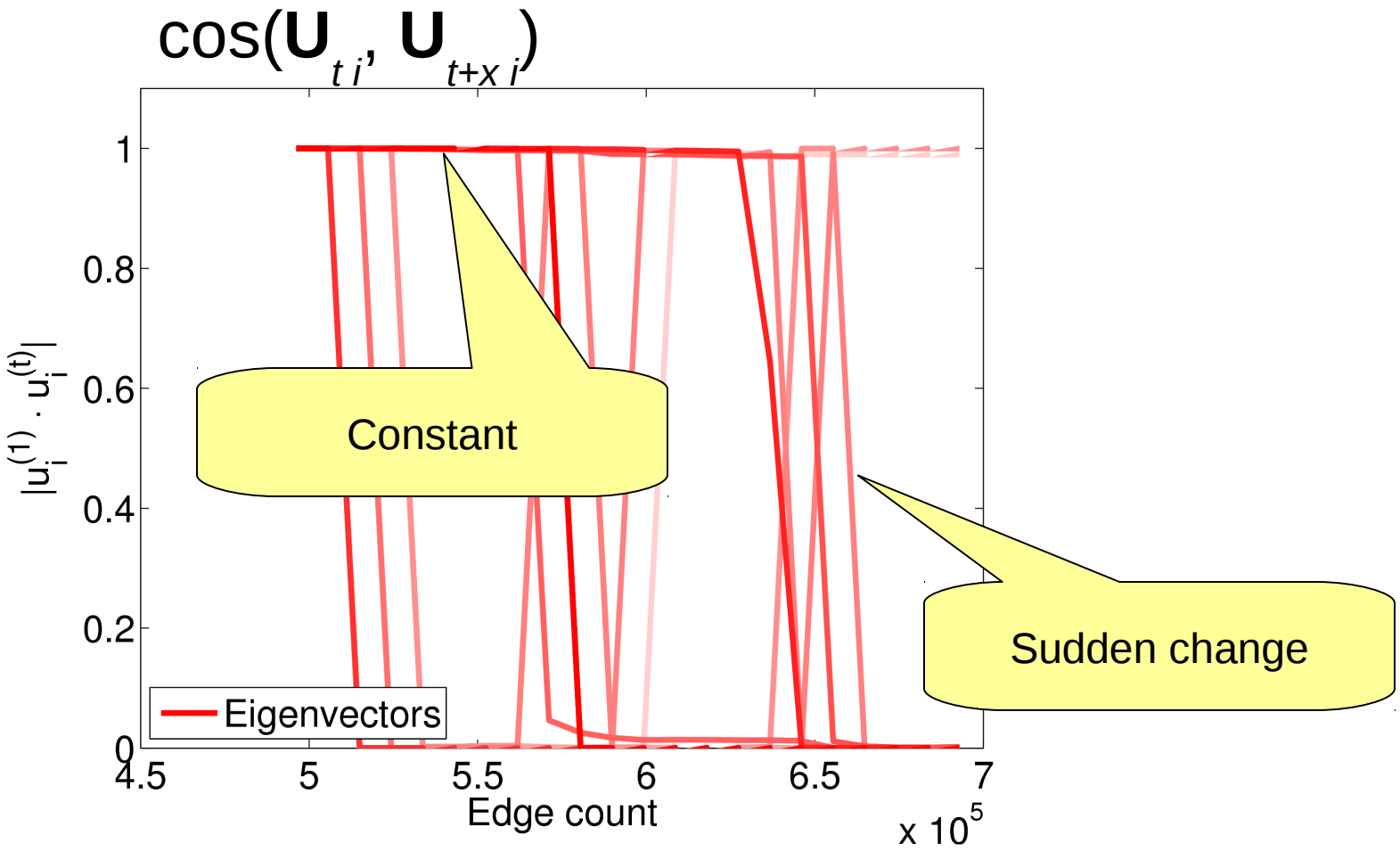
Compute all eigenvalue decompositions $\mathbf{A}_t = \mathbf{U}_t \mathbf{\Lambda}_t \mathbf{U}_t^{\mathsf{T}}$

Evolution of Eigenvalues



Eigenvector Evolution

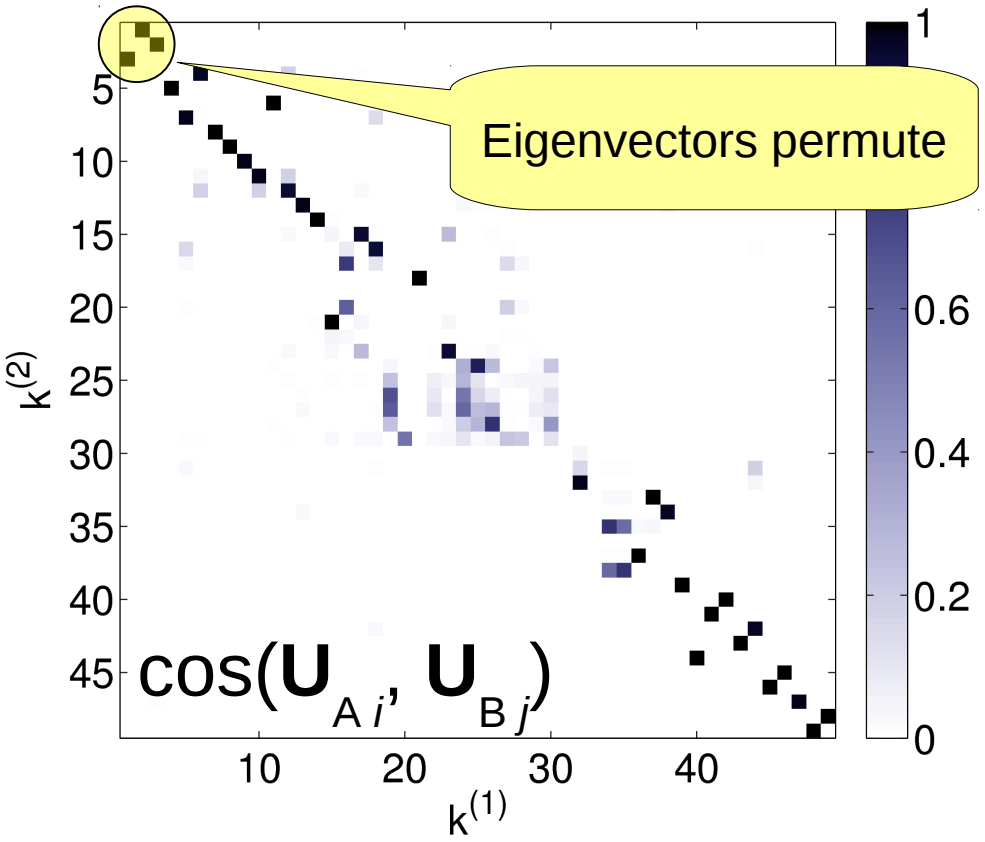
Cosine similarity between $\mathbf{U}_{t\,i}$ and $\mathbf{U}_{t+x\,i}$



Eigenvector Permutation

Split: old edges $\mathbf{A} = \mathbf{U}_A \mathbf{\Lambda}_A \mathbf{U}_A^T$

new edges $\mathbf{B} = \mathbf{U}_B \mathbf{\Lambda}_B \mathbf{U}_B^T$



Diagonality Test

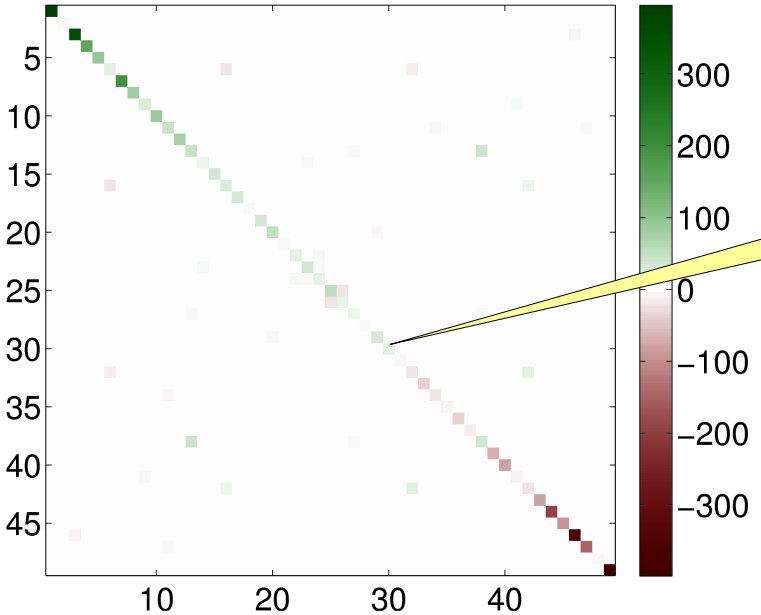
Diagonalize $\mathbf{A}_B$ using the eigenvectors of $\mathbf{A}_A$

$$\mathbf{A}_A = \mathbf{U}_A \mathbf{\Lambda}_A \mathbf{U}_A^T$$

$$\mathbf{A}_B = \mathbf{U}_A \mathbf{X} \mathbf{U}_A^T$$

$$\Rightarrow \mathbf{X} = \mathbf{U}_A^T \mathbf{A}_B \mathbf{U}_A$$

$\mathbf{X} =$



Nearly diagonal!

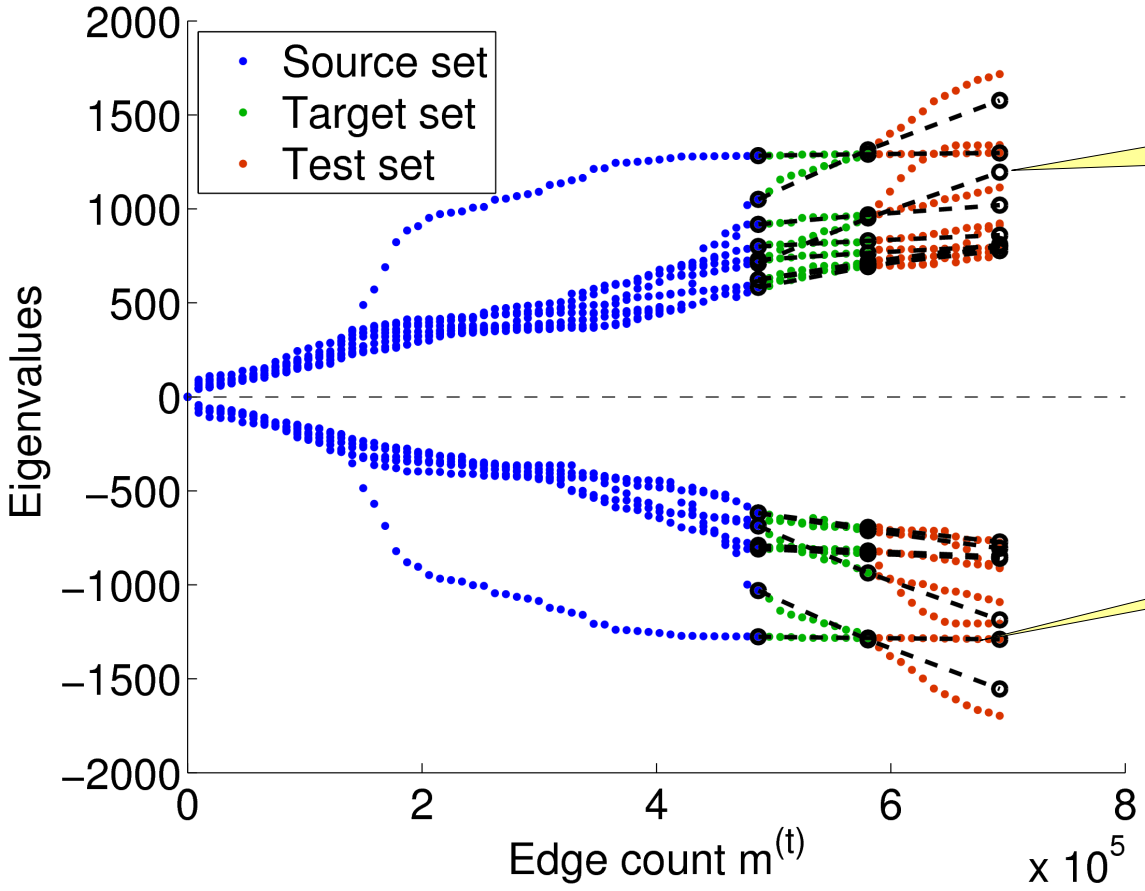
Outline

1. Algebraic Link Prediction
2. Spectral Transformations
3. **Learning Link Prediction**
   - Learning by Extrapolation
   - Learning by Curve Fitting
4. Special Cases

What spectral transformation is best?

Learning by Extrapolation

Extrapolate the growth of the spectrum



Potential problem: overfitting

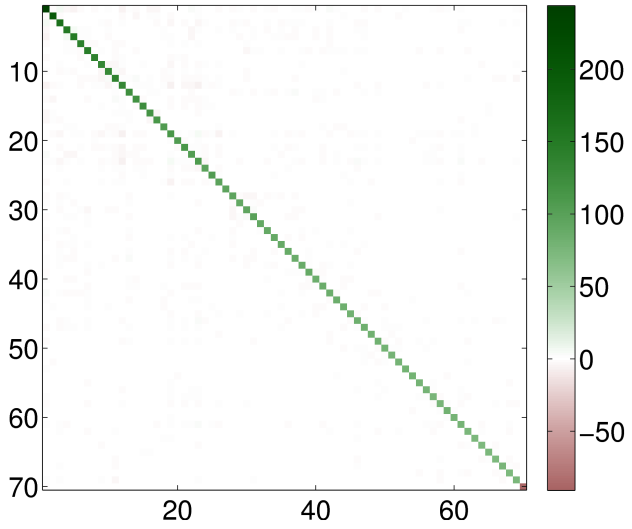
Good when growth is irregular

Learning by Curve Fitting

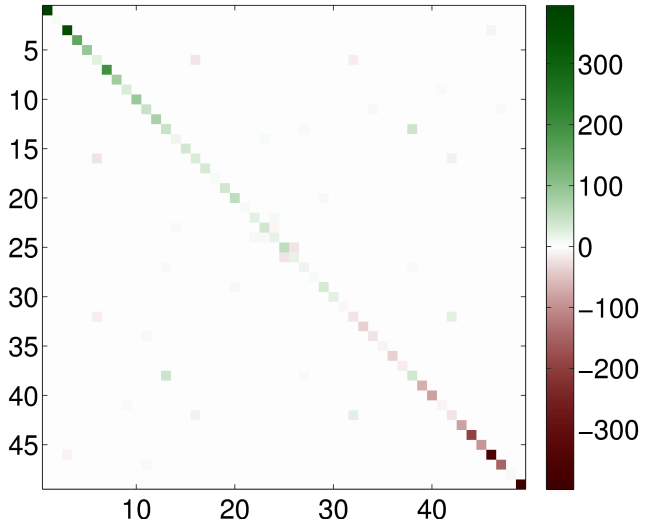
$$\mathbf{A}_A \xrightarrow{f} \mathbf{A}_B$$

$$\mathbf{U}_A \mathbf{\Lambda}_A \mathbf{U}_A^T \longrightarrow \mathbf{A}_B$$

$$\mathbf{\Lambda}_A \longrightarrow \mathbf{U}_A^T \mathbf{A}_B \mathbf{U}_A$$

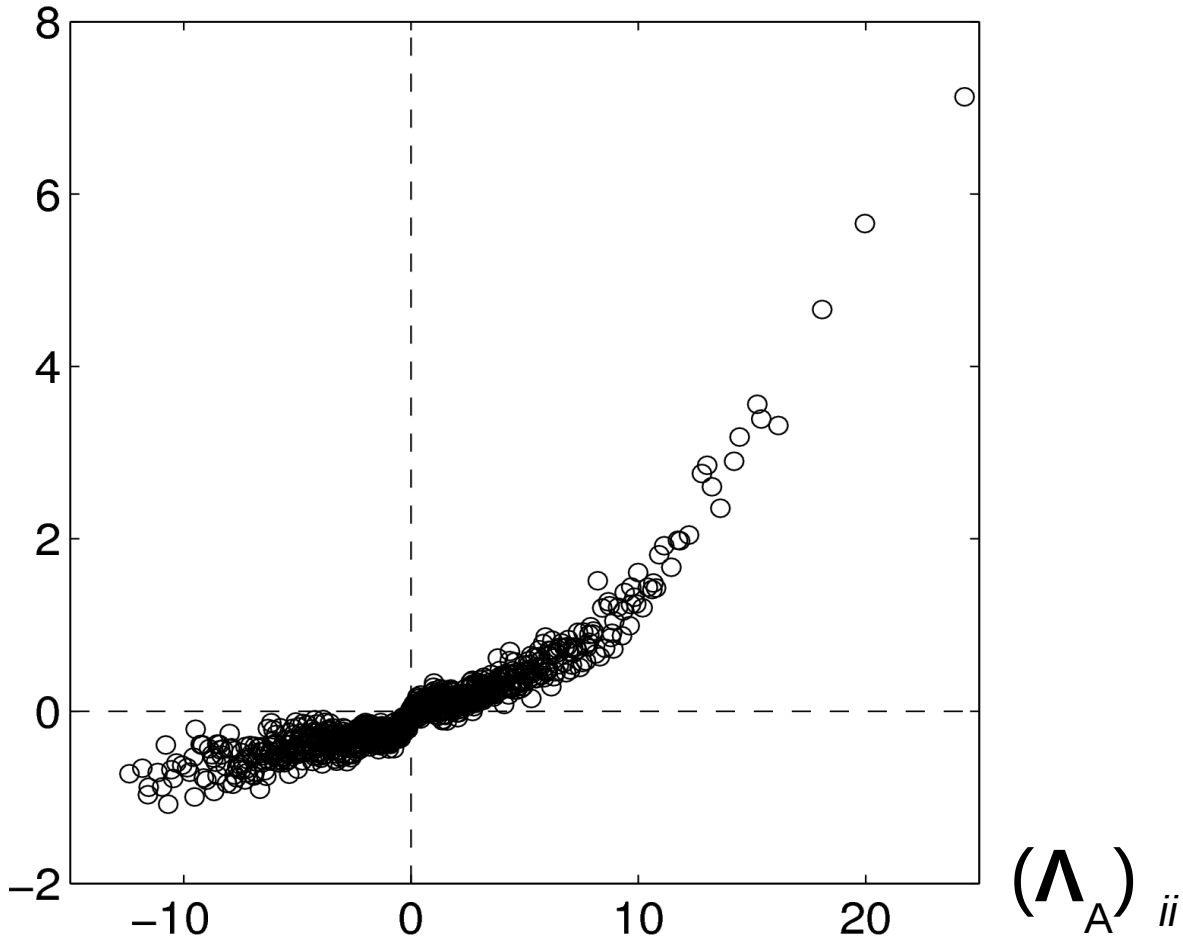


f



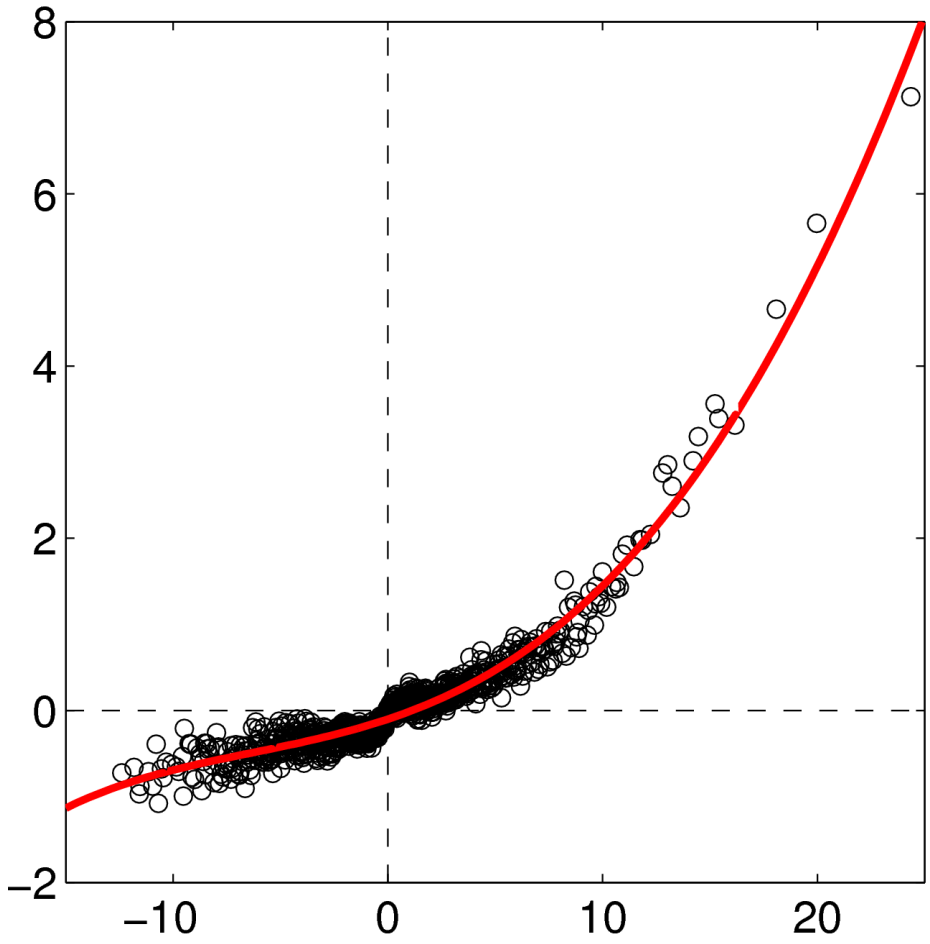
Curve Fitting

$(U_A^T A_B U_A)_{ii}$

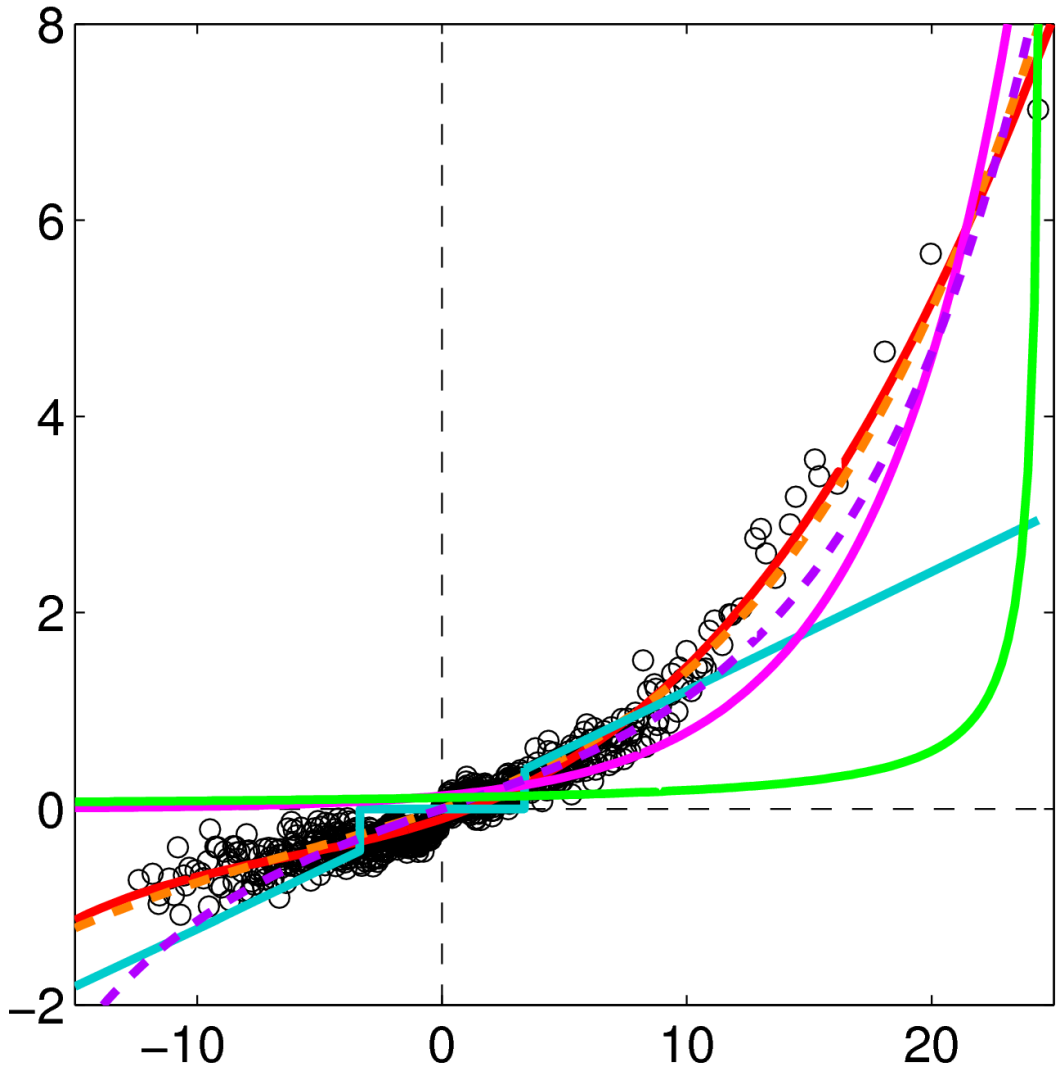


Polynomial Curve Fitting

Fit a polynomial $a + b\,x + c\,x^2 + d\,x^3 + e\,x^4$



Other Curves



Polynomial
$a + b\,x + c\,x^2 + d\,x^3 + e\,x^4$

Nonnegative Polynomial
$a + \ldots + h\,x^7$ $\quad a, \ldots, h \geq 0$

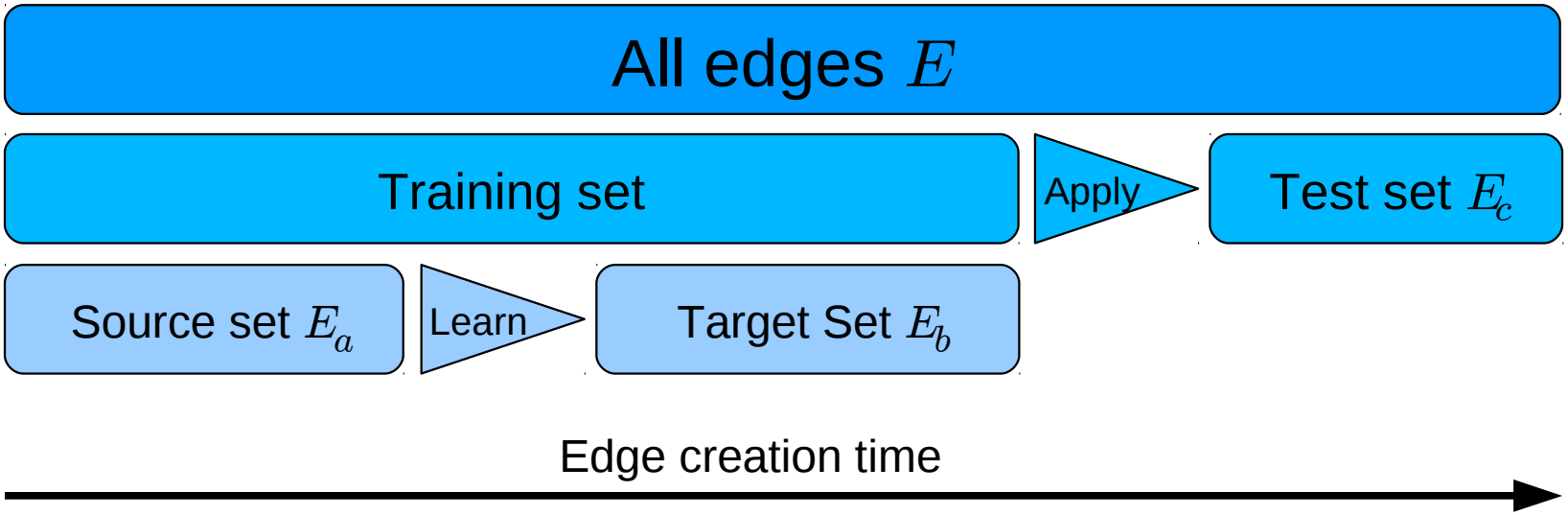
Von Neumann kernel
$b\,/\,(1 - a\,x)$

Matrix exponential
$b \exp(a\,x)$

Hyperbolic sine
$b \sinh(a\,x)$

Rank reduction
$a\,x$ if $|x| \geq x_0$, 0 otherwise

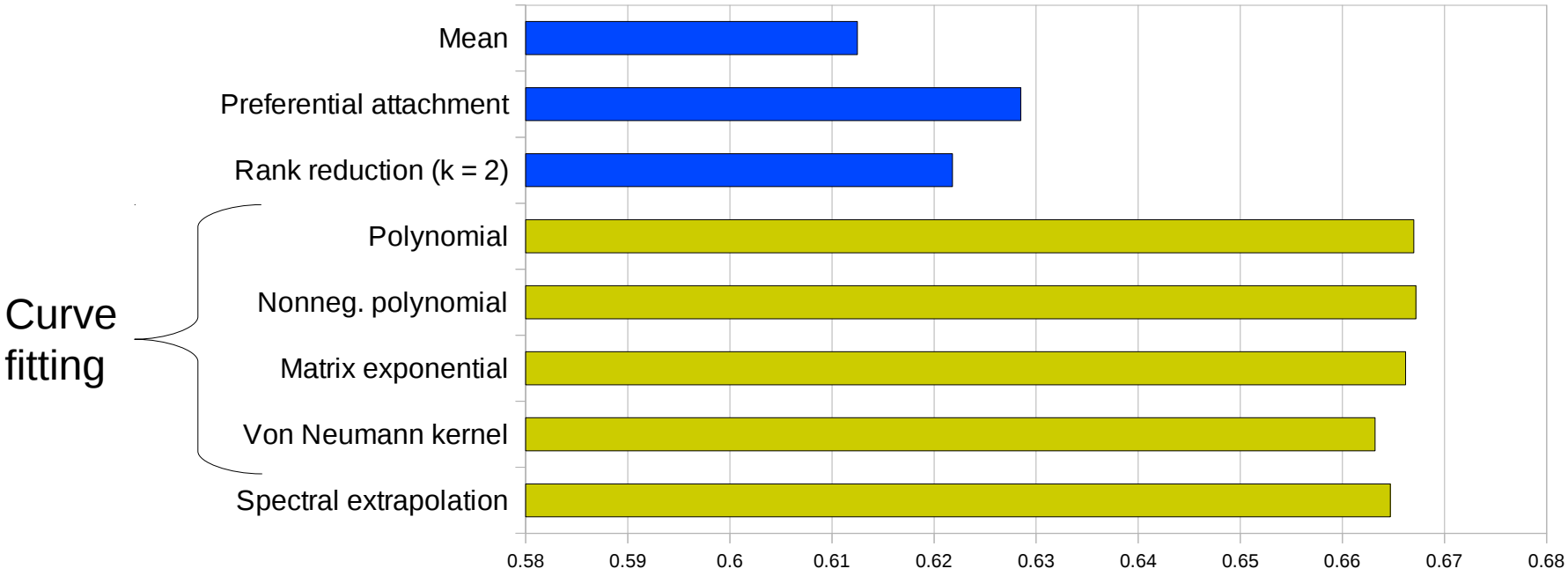
Evaluation Methodology



3-way split of edge set by edge creation times

Experiments

Precision of link prediction (1 = perfect)

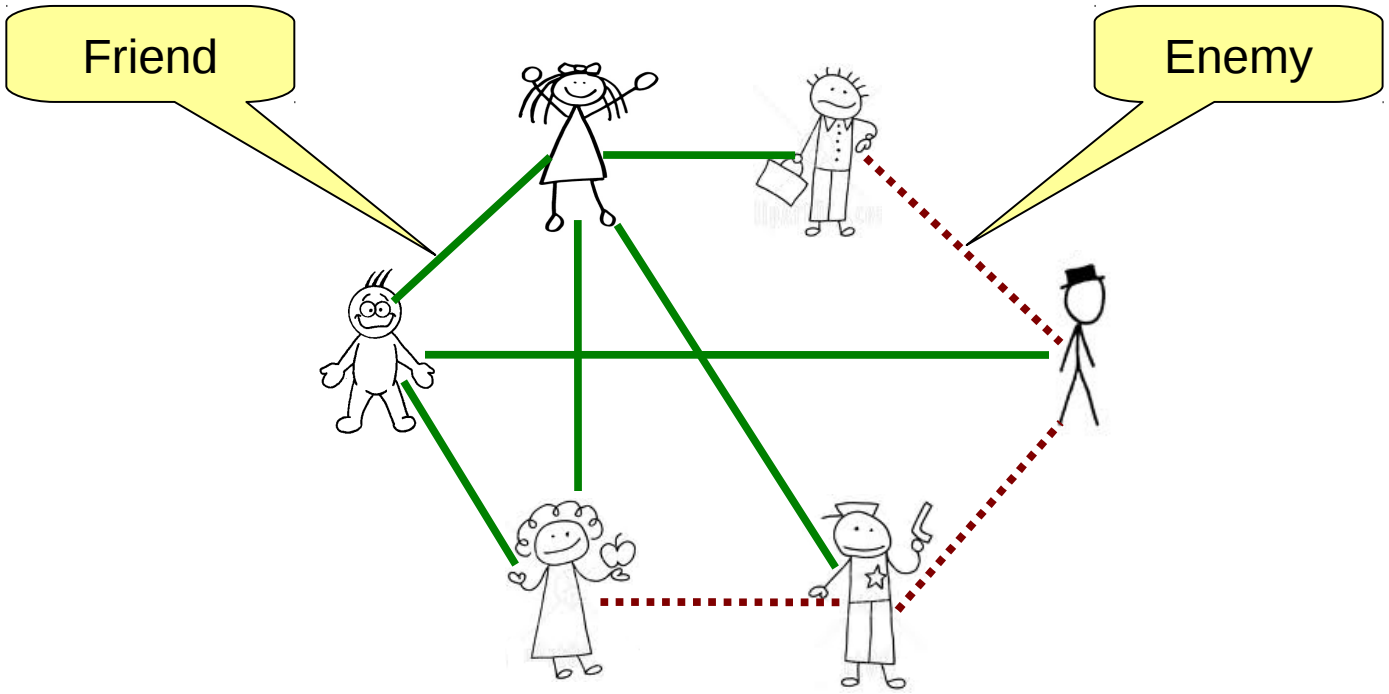


Outline

1. Algebraic Link Prediction
2. Spectral Transformations
3. Learning Link Prediction
4. **Special Cases**
   - Signed networks
   - Directed networks
   - Bipartite networks

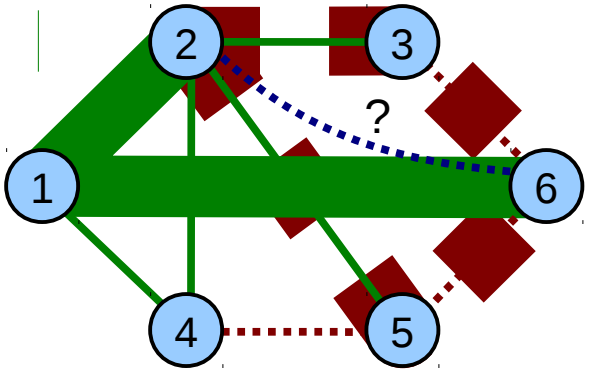
Not all networks are simple

Signed Networks



Two link types: positive and negative

Signed Networks: Algebraic Analysis



$$\mathbf{A} = \begin{vmatrix} 0 & +1 & 0 & +1 & 0 & +1 \\ +1 & 0 & +1 & +1 & +1 & 0 \\ 0 & +1 & 0 & 0 & 0 & -1 \\ +1 & +1 & 0 & 0 & -1 & 0 \\ 0 & +1 & 0 & -1 & 0 & -1 \\ +1 & 0 & -1 & 0 & -1 & 0 \end{vmatrix}$$

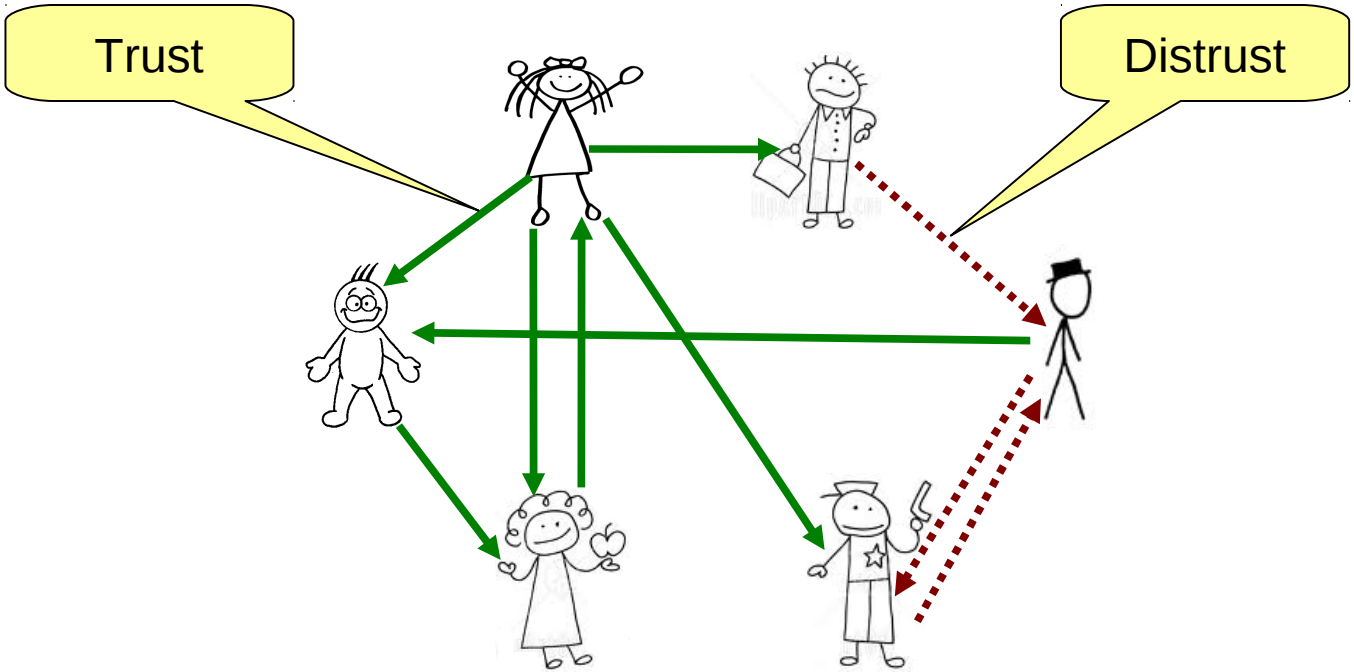
The square $\mathbf{A}^2$ counts common friends / enemies

$$(\mathbf{A}^2)_{ij} = \sum_k \mathbf{A}_{ik}\,\mathbf{A}_{jk}$$

$$(\mathbf{A}^2)_{26} = (+1)(+1) + (+1)(-1) + (+1)(-1) = -1$$

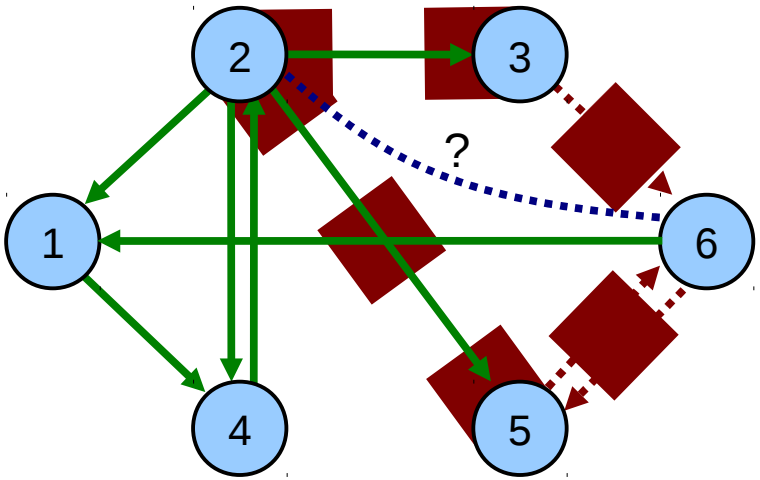
‘The enemy of my enemy is my friend’

Directed Networks



Links have a direction

Directed Networks: Algebraic Analysis



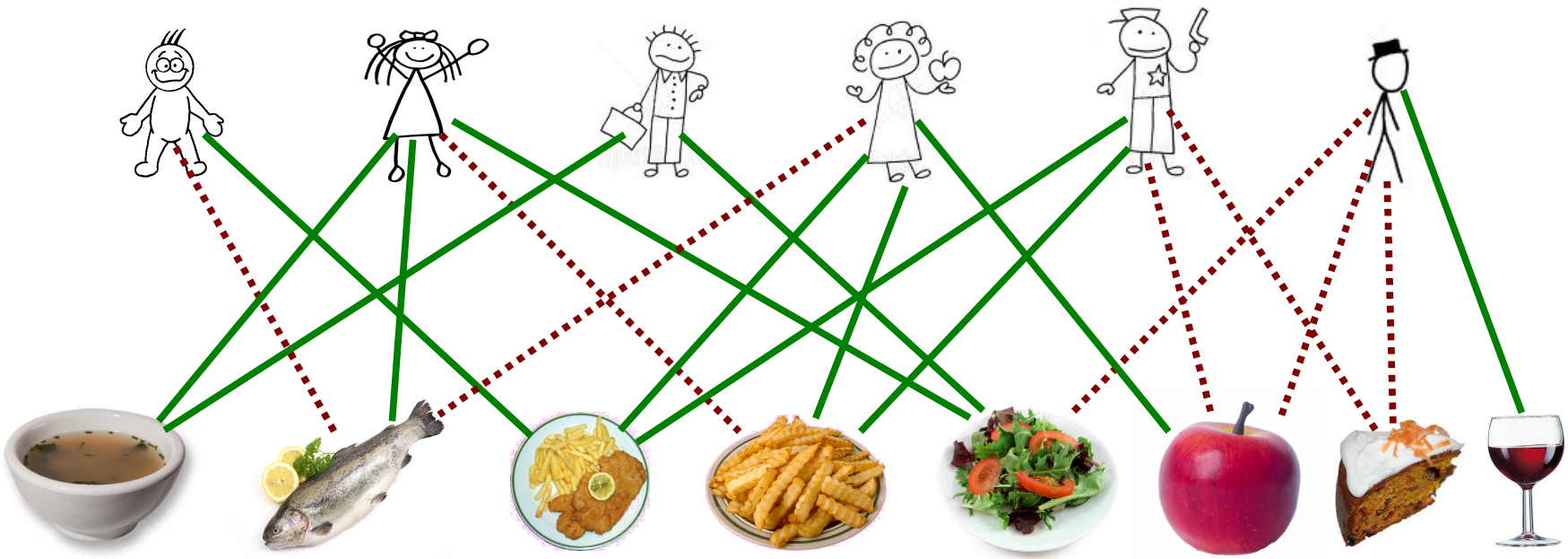
$$\mathbf{A} = \begin{vmatrix} 0 & 0 & 0 & +1 & 0 & 0 \\ +1 & 0 & +1 & +1 & +1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ +1 & 0 & 0 & 0 & -1 & 0 \end{vmatrix}$$

The square $\mathbf{A}^2$ counts directed paths

$$(\mathbf{A}^2)_{ij} = \sum_k \mathbf{A}_{ik}\,\mathbf{A}_{kj}$$

$$(\mathbf{A}^2)_{26} = (+1)(-1) + (+1)(-1) = -2$$

Bipartite Networks



Links between two different node types

Bipartite Networks: Spectral Analysis

Rectangular matrix **B** =

$$\begin{vmatrix} 0 & -1 & +1 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & 0 & 0 & +1 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & +1 & 0 & 0 & 0 \\ 0 & -1 & +1 & +1 & 0 & 1 & 0 & 0 \\ 0 & 0 & +1 & +1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & +1 \end{vmatrix}$$

Singular value decomposition of **B**:

$$\mathbf{B} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{T}$$

Left and right eigenvectors are different

Conclusion

Can the eigenvalue decomposition model the growth of networks?
Yes, because only eigenvalues change, not eigenvectors.

Is there an explanation for constant eigenvectors?
There are several: Graph kernels, triangle closing, the sum-over-paths model, rank reduction, etc.

Can we exploit this in recommender systems?
Yes, by learning the spectral transformation for a given dataset.

Is this scalable?
Yes, because we can reduce the learning problem to a one dimensional curve fitting problem.

Selected Publications

The Slashdot Zoo: Mining a social network with negative edges
J. Kunegis, A. Lommatzsch and C. Bauckhage
In Proc. World Wide Web Conf., pp. 741–750, 2009.

Learning spectral graph transformations for link prediction
J. Kunegis and A. Lommatzsch
In Proc. Int. Conf. on Machine Learning, pp. 561–568, 2009.

Spectral analysis of signed graphs for clustering, prediction and visualization
J. Kunegis, S. Schmidt, A. Lommatzsch and J. Lerner
In Proc. SIAM Int. Conf. on Data Mining, pp. 559–570, 2010.

Network growth and the spectral evolution model
J. Kunegis, D. Fay and C. Bauckhage
In Proc. Conf. on Information and Knowledge Management, pp. 739–748, 2010.

References

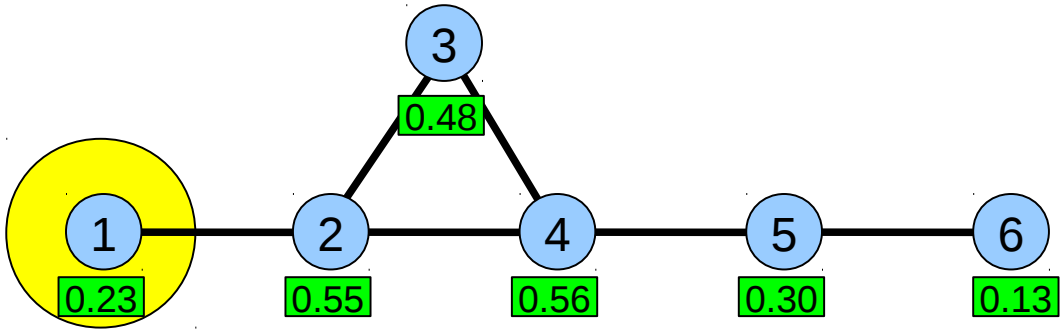
B. Viswanath, A. Mislove, M. Cha, K. P. Gummadi, On the evolution of user interaction in Facebook. In Proc. Workshop on Online Social Networks, pp. 37–42, 2009.

Acknowledgements

Backup: Other Findings

- **Evaluation on 114 network datasets**
- **Control tests**: Spectral evolution is not observed in random graph growth models
- **Spectral transformations of the Laplacian matrix**
- **The Laplacian matrix with negative edges**

Backup: Eigenvalue Decomposition: Example



$$A = U \Lambda U^T$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$U = \begin{pmatrix} 0.25 & -0.43 & -0.17 & 0.76 & 0.30 & 0.23 \\ -0.44 & 0.59 & 0.10 & 0.21 & 0.33 & 0.55 \\ -0.10 & -0.56 & 0.51 & -0.39 & 0.18 & 0.48 \\ 0.61 & 0.18 & -0.41 & -0.32 & -0.12 & 0.56 \\ -0.52 & -0.28 & -0.37 & 0.09 & -0.64 & 0.30 \\ 0.30 & 0.20 & 0.63 & 0.34 & -0.59 & 0.13 \end{pmatrix}$$

Eigenvector

$$\Lambda = \begin{pmatrix} -1.74 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.37 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.59 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.27 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.33 \end{pmatrix}$$

Eigenvalue

$$U^T = \begin{pmatrix} 0.25 & -0.43 & -0.17 & 0.76 & 0.30 & 0.23 \\ -0.44 & 0.59 & 0.10 & 0.21 & 0.33 & 0.55 \\ -0.10 & -0.56 & 0.51 & -0.39 & 0.18 & 0.48 \\ 0.61 & 0.18 & -0.41 & -0.32 & -0.12 & 0.56 \\ -0.52 & -0.28 & -0.37 & 0.09 & -0.64 & 0.30 \\ 0.30 & 0.20 & 0.63 & 0.34 & -0.59 & 0.13 \end{pmatrix}$$

U contains eigenvectors, **Λ** contains eigenvalues