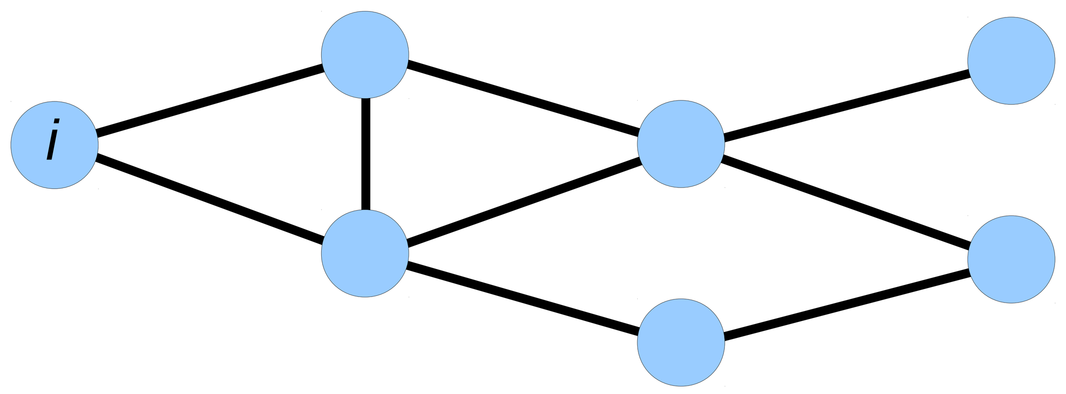
Jérôme Kunegis
Fachbereich 4, Institute WeST – Web Science and Technologies
Prof. Steffen Staab, Prof. Klaus Obermayer, Prof. Christian Bauckhage
kunegis@uni-koblenz.de

UNIVERSITÄT KOBLENZ · LANDAU

On the Spectral Evolution of Large Networks

EXAMPLE: RECOMMEND FRIENDS ON FACEBOOK



Consider a network of friends connected by friendship links.

- Given a person *i*, find new friends *j* for that person (recommendation problem)
- Equivalent problem: Find edges (*i*, *j*) that will appear in the future (link prediction problem)

EIGENVALUE DECOMPOSITION

Use the adjacency matrix **A**:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Compute the eigenvalue decomposition:

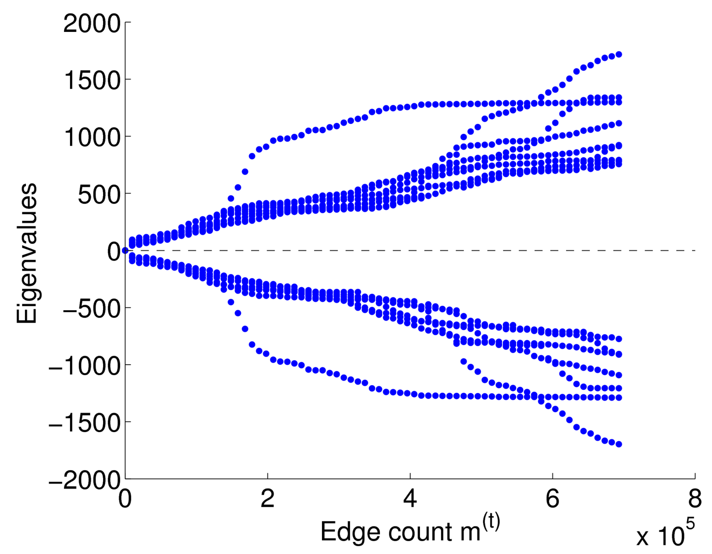
$$\mathbf{A} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$$

where **U** is an orthogonal matrix (i.e. $\mathbf{U}^T \mathbf{U} = \mathbf{I}$) and $\mathbf{\Lambda}$ is a diagonal matrix.
U contains the eigenvectors and $\mathbf{\Lambda}$ the eigenvalues of **A**.

EVOLUTION OF THE EIGENVALUE DECOMPOSITION

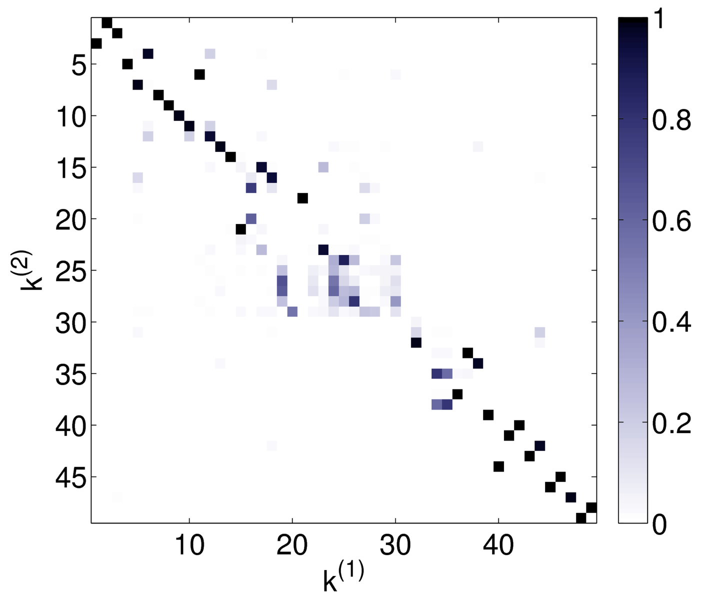
How does the eigenvalue decomposition of Facebook change over time?

Let $\mathbf{A}_{(t)} = \mathbf{U}_{(t)} \mathbf{\Lambda}_{(t)} \mathbf{U}_{(t)}^T$ be the adjacency matrix of the Facebook network at time *t* (going from 1 to *n*)



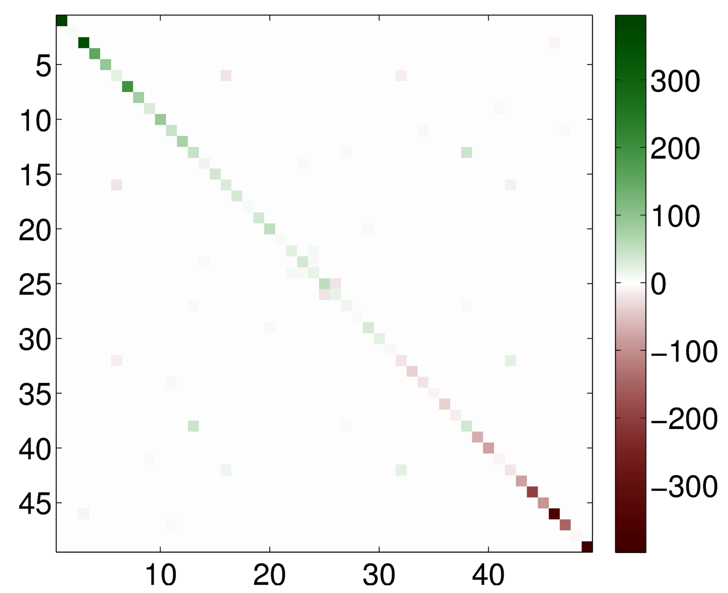
Eigenvalues grow

$(\mathbf{\Lambda}_{(t)})_{ii}$ vs *t*



Eigenvectors permute!

$\mathbf{U}_{(n)}^T \mathbf{U}_{(1)}$



Express $\mathbf{A}_{(n)}$ using $\mathbf{U}_{(1)}$

$$\mathbf{A}_{(n)} = \mathbf{U}_{(1)} \mathbf{\Delta} \mathbf{U}_{(1)}$$
$$\mathbf{\Delta} = \mathbf{U}_{(1)}^T \mathbf{A}_{(n)} \mathbf{U}_{(1)}$$

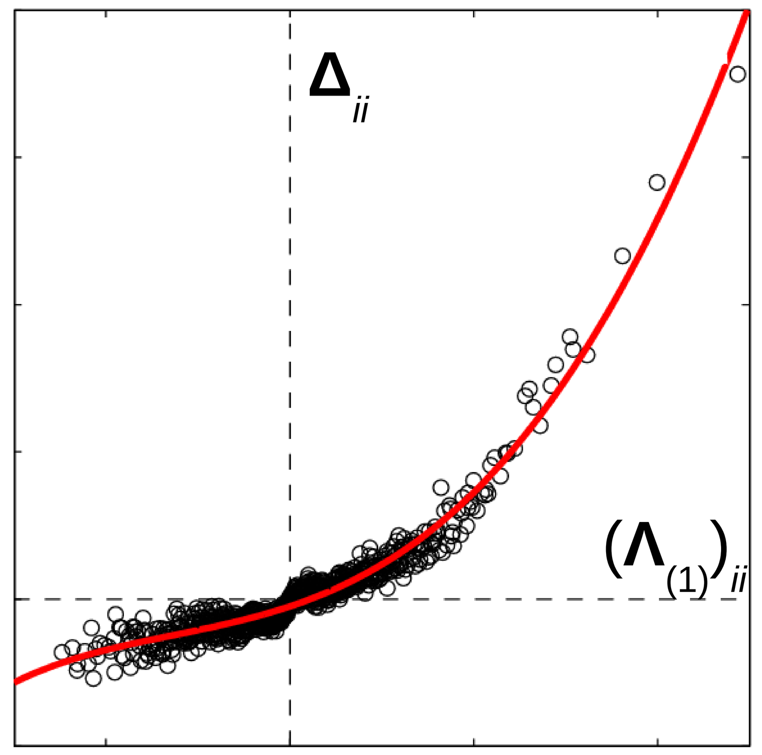
$\mathbf{\Delta}$ is diagonal!

PREDICTING NETWORK EVOLUTION

Replace $\mathbf{\Lambda}_{(1)}$ by the diagonal elements of

$$\mathbf{\Delta} = \mathbf{U}_{(1)}^T \mathbf{A}_{(n)} \mathbf{U}_{(1)}$$

Learn this mapping by curve fitting:



Polynomial curve fitting corresponds to

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

Note: A polynomial of **A** is a weighted sum of path counts!

REFERENCES

J. Kunegis, A. Lommatzsch, *Learning spectral graph transformations for link prediction*, ICML 2009.
J. Kunegis, D. Fay, C. Bauckhage, *Network growth and the spectral evolution model*, CIKM 2010.