Learning Spectral Graph Transformations for Link Prediction

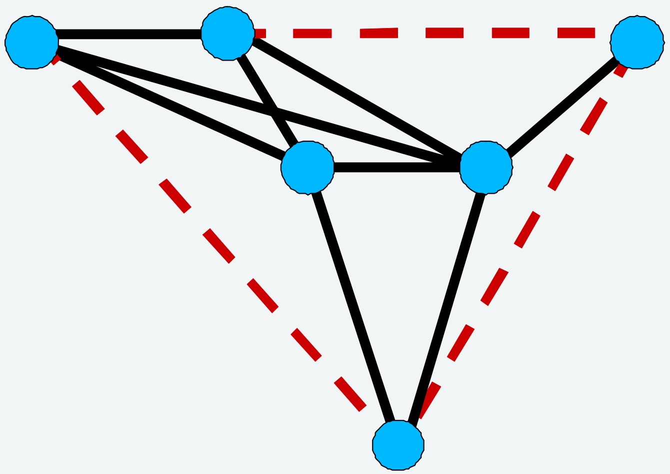
Jérôme Kunegis & Andreas Lommatzsch
DAI-Labor, Technische Universität Berlin, Germany
26th Int. Conf. on Machine Learning, Montréal 2009

CC IRML
Information Retrieval
& Machine Learning

Outline

- The Problem
  - Link prediction
  - Known solutions
- Learning
  - Spectral transformations
  - Finding the best spectral transformation
- Variants
  - Weighted and signed graphs
  - Bipartite graphs and the SVD
  - Graph Laplacian and normalization
- Some Applications

The Problem: Link Prediction



Known links **(A)**

Links to be predicted **(B)**

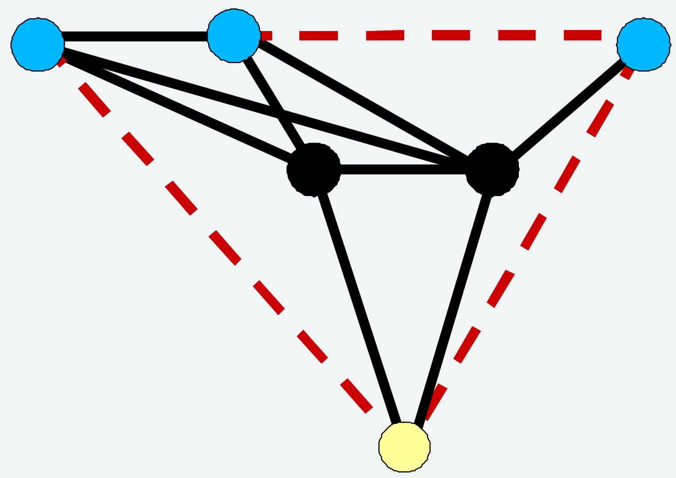
- Motivation: Recommend connections in a social network
- Predict links in an undirected, unweighted network

- Using the adjacency matrices **A** and **B,**
- Find a function $F(\mathbf{A})$ giving prediction values corresponding to **B**

$$F(\mathbf{A}) = \mathbf{B}$$

Path Counting

- Follow paths



- Number of paths of length k given by $\mathbf{A}^k$
- Nodes connected by many paths
- Nodes connected by short paths
- Weight powers of $\mathbf{A}$: $\alpha\mathbf{A}^2 + \beta\mathbf{A}^3 + \gamma\mathbf{A}^4 \ldots$ with $\alpha > \beta > \gamma \ldots [> 0]$
- Examples:

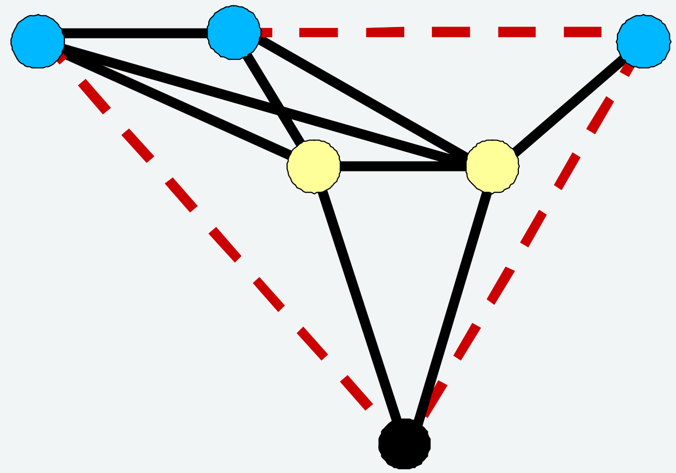
Exponential graph kernel: $e^{\alpha\mathbf{A}} = \Sigma_i \alpha^i/i!\ \mathbf{A}^i$

Von Neumann kernel: $(\mathbf{I} - \alpha\mathbf{A})^{-1} = \Sigma_i \alpha^i\ \mathbf{A}^i$

(with $0 < \alpha < 1$)

Laplacian Link Prediction Functions

- Graph Laplacian **L** = **D** − **A**



“Resistance Distance” (a.k.a. commute time)	$\mathbf{L}^{+}$
Regularized Laplacian	$(\mathbf{I} + \alpha\mathbf{L})^{-1}$
Heat diffusion kernel	$e^{-\alpha\mathbf{L}}$

Computation of Link Prediction Functions

Adjacency matrix — eigenvalue decomposition: $\mathbf{A} = \mathbf{U\Lambda U}^T$

Matrix polynomial	$\sum_i \alpha_i \mathbf{A}^i$	$= \mathbf{U} \left(\sum_i \alpha_i \mathbf{\Lambda}^i\right) \mathbf{U}^T$
Matrix exponential	$e^{\alpha \mathbf{A}}$	$= \mathbf{U}\, e^{\alpha \mathbf{\Lambda}}\, \mathbf{U}^T$
Von Neumann kernel	$(\mathbf{I} - \alpha \mathbf{A})^{-1}$	$= \mathbf{U}\, (\mathrm{I} - \alpha \mathbf{\Lambda})^{-1}\, \mathbf{U}^T$
Rank reduction	$\mathbf{A}_{(k)}$	$= \mathbf{U}\, \mathbf{\Lambda}_{(k)}\, \mathbf{U}^T$

Graph Laplacian — eigenvalue decomposition $\mathbf{L} = \mathbf{D} - \mathbf{A} = \mathbf{U\Lambda U}^T$

Resistance distance	$\mathbf{L}^+$	$= \mathbf{U}\, \mathbf{\Lambda}^+\, \mathbf{U}^T$
Regularized Laplacian	$(\mathbf{I} + \alpha \mathbf{L})^{-1}$	$= \mathbf{U}\, (\mathbf{I} + \alpha \mathbf{\Lambda})^{-1}\, \mathbf{U}^T$
Heat diffusion kernel	$e^{-\alpha \mathbf{L}}$	$= \mathbf{U}\, e^{-\alpha \mathbf{\Lambda}}\, \mathbf{U}^T$

Spectral transformation

Learning Spectral Transformations

- Link prediction functions are **spectral transformations** of **A** or **L**

$$F(\mathbf{A}) = \mathbf{U}F(\mathbf{\Lambda})\mathbf{U}^T$$
$$F(\mathbf{\Lambda})_{ii} = f(\mathbf{\Lambda}_{ii})$$

- A spectral transformation *F* corresponds to a function of reals *f*

Matrix polynomial	$F(\mathbf{A}) = \sum_i \alpha_i \mathbf{A}^i$	$f(x) = \sum_i \alpha_i x^i$	Real polynomial
Matrix exponential	$F(\mathbf{A}) = e^{\alpha \mathbf{A}}$	$f(x) = e^{\alpha x}$	Real exponential
Matrix inverse	$F(\mathbf{A}) = (\mathbf{I} \pm \alpha \mathbf{A})^{-1}$	$f(x) = 1 / (1 \pm \alpha x)$	Rational function
Pseudoinverse	$F(\mathbf{A}) = \mathbf{A}^+$	$f(x) = 1/x$ when $x > 0$, 0 otherwise	
Rank-*k* approximation	$F(\mathbf{A}) = \mathbf{A}_{(k)}$	$f(x) = x$ when $\lvert x \rvert \geq x_0$, 0 otherwise	

Finding the Best Spectral Transformation

- Find the best spectral transformation on test set **B**

$$\min_{F} \; \|F(A) - B\|_F$$

- Reduce minimization problem

$$= \min_{F} \; \|U F(\Lambda)\, U^T - B\|_F$$

$$= \min_{F} \; \|F(\Lambda) - U^T B U\|_F \qquad \text{norm is preserved by U}$$

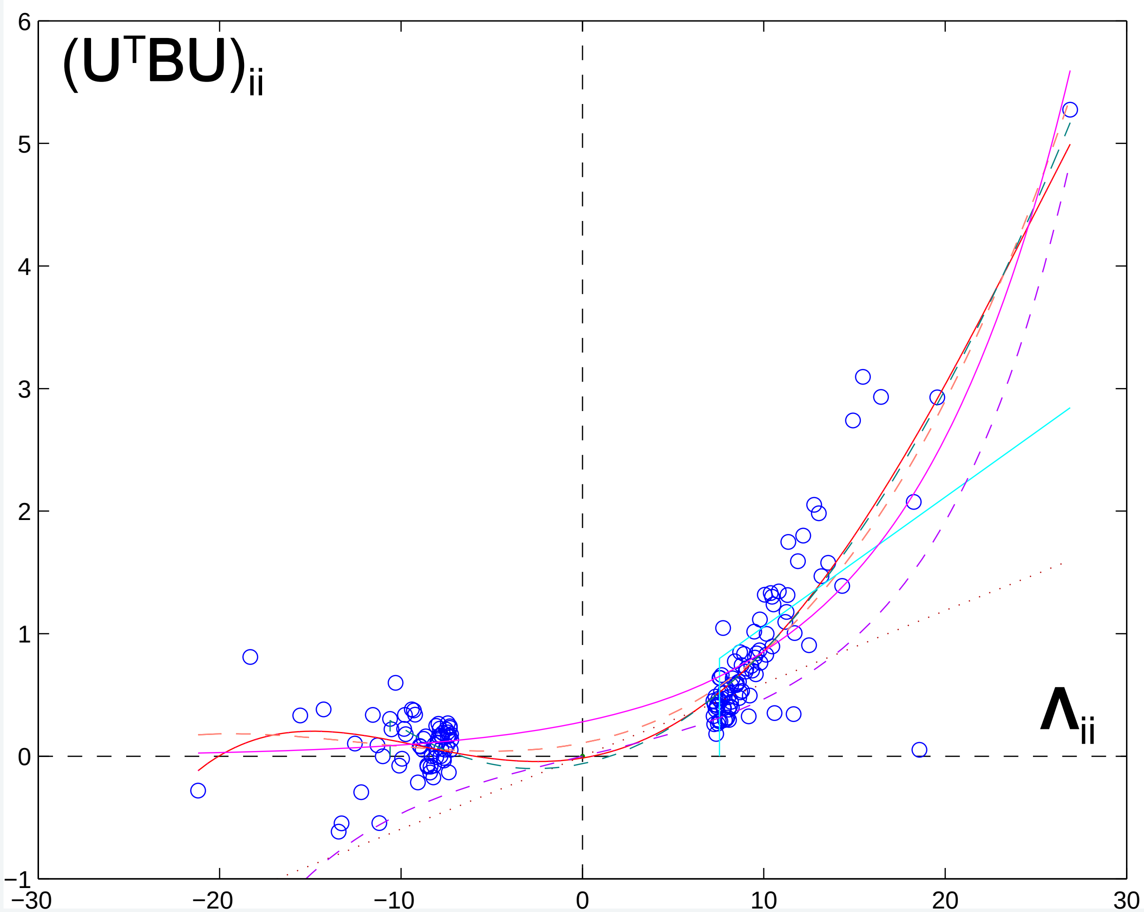
- Reduce to diagonal, because off-diagonal in $F(\Lambda)$ is constant zero

$$\min_{f} \; \sum_i \left(f(\Lambda_{ii}) - (U^T B U)_{ii}\right)^2$$

- The best spectral transformation is given by a one-dimensional least-squares problem

Example: DBLP Citation Network (undirected)

- DBLP citation network
- Symmetric adjacency matrices $\mathbf{A} = \mathbf{U\Lambda U}^T$, $\mathbf{B}$



$\alpha x^3 + \beta x^2 + \gamma x + \delta$

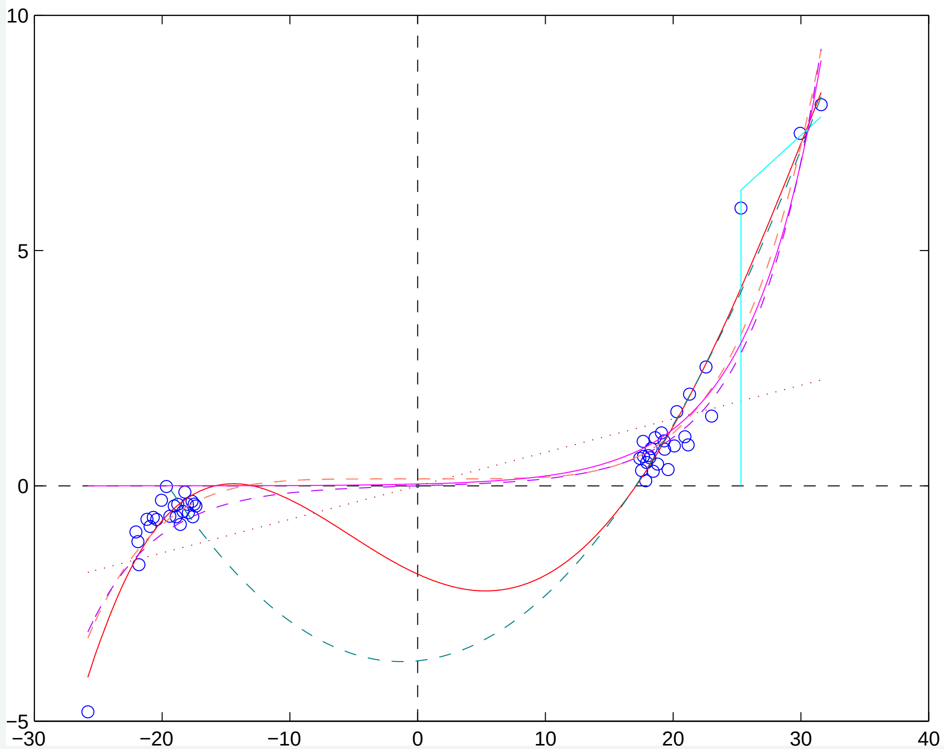
$\beta e^{\alpha x}$

x when $x < x_0$, else 0

Variants: Weighted and Signed Graphs

- Weighted undirected graphs: use **A** and **L** = **D** − **A** as is
- Signed graphs: use $\mathbf{D}_{ii} = \Sigma_j |A_{ij}|$ (signed graph Laplacian)
- Example: Slashdot Zoo

(social network with negative edges)

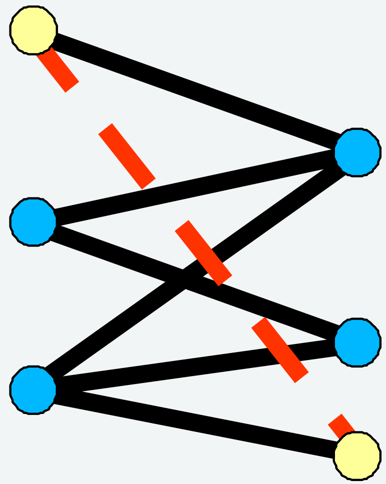


Bipartite Graphs

- Bipartite graphs: paths have **odd length**

- Compute sum of odd powers of **A**
- The resulting polynomial is **odd**

  $\alpha A^3 + \beta A^5 + \ldots$

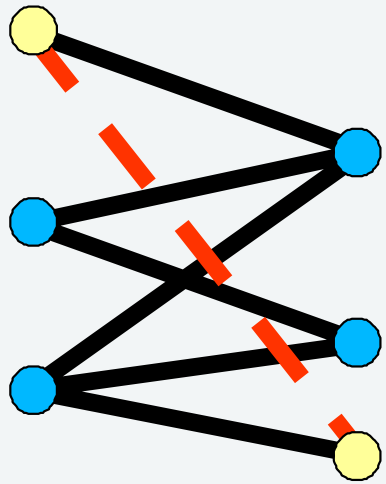


- For other link prediction functions, use the odd component

$e^{\alpha \mathbf{A}}$	$\sinh(\alpha \mathbf{A})$
$(\mathbf{I} - \alpha \mathbf{A})^{-1}$	$\alpha \mathbf{A} (\mathbf{I} - \alpha^2 \mathbf{A}^2)^{-1}$

Singular Value Decomposition

- Odd power of a bipartite graph's adjacency matrix



$\mathbf{A}^{2n+1}$ = [0 $\mathbf{R}$; $\mathbf{R}^T$ 0]$^{2n+1}$ = [0 $(\mathbf{R}\mathbf{R}^T)^n\mathbf{R}$; $\mathbf{R}^T(\mathbf{R}\mathbf{R}^T)^n$ 0]

- Using the singular value decomposition $R = U\Sigma V^T$

$(RR^T)^nR = (U\Sigma V^T V\Sigma U^T)^n U\Sigma V^T = (U\Sigma^2U^T)^n U\Sigma V^T = U\Sigma^{2n+1}V^T$

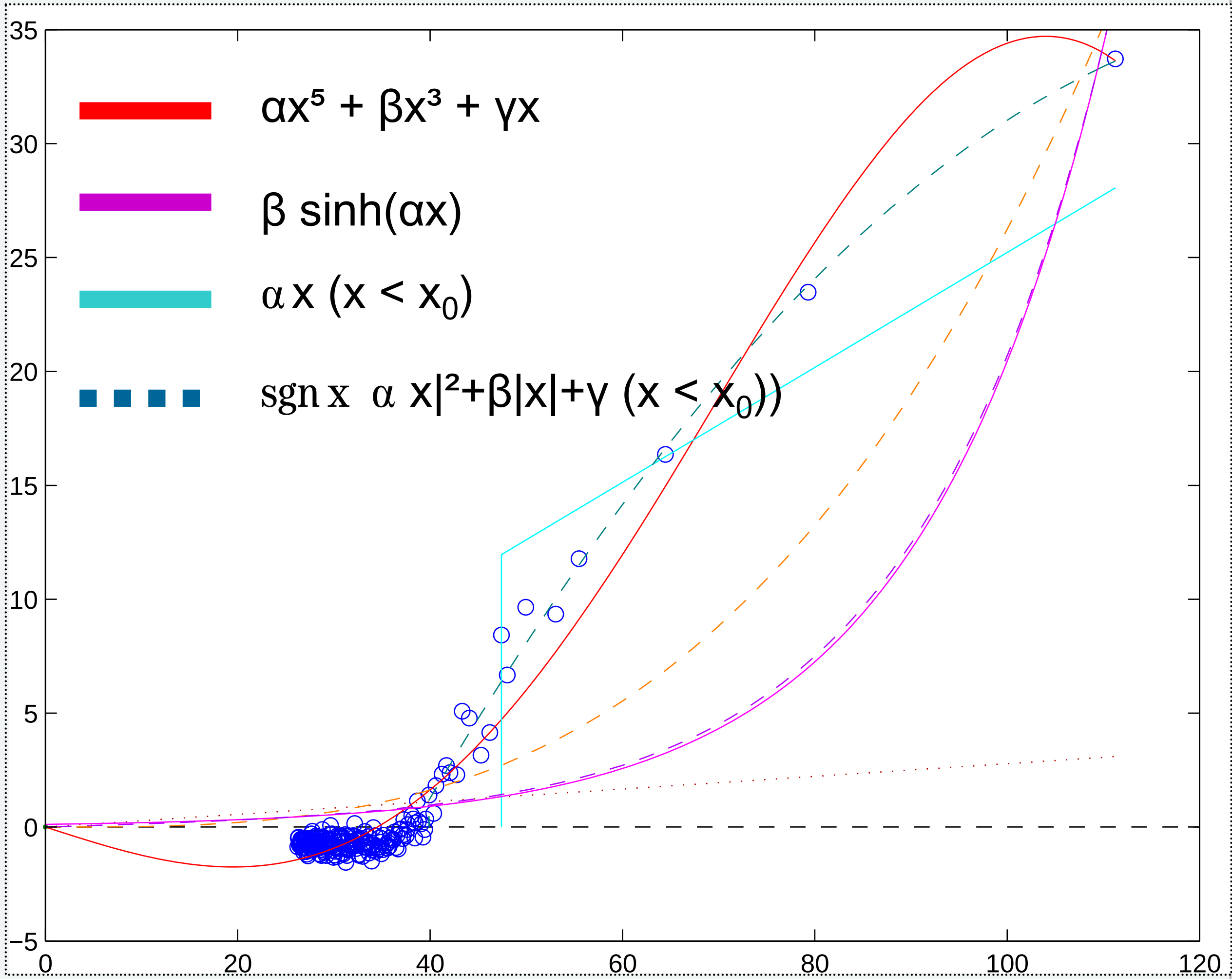
- Odd powers of A are given by odd spectral transformations of R

SVD Example: Bipartite Rating Graph

- MovieLens rating graph

Rating values

{−2, −1, 0, +1, +2}

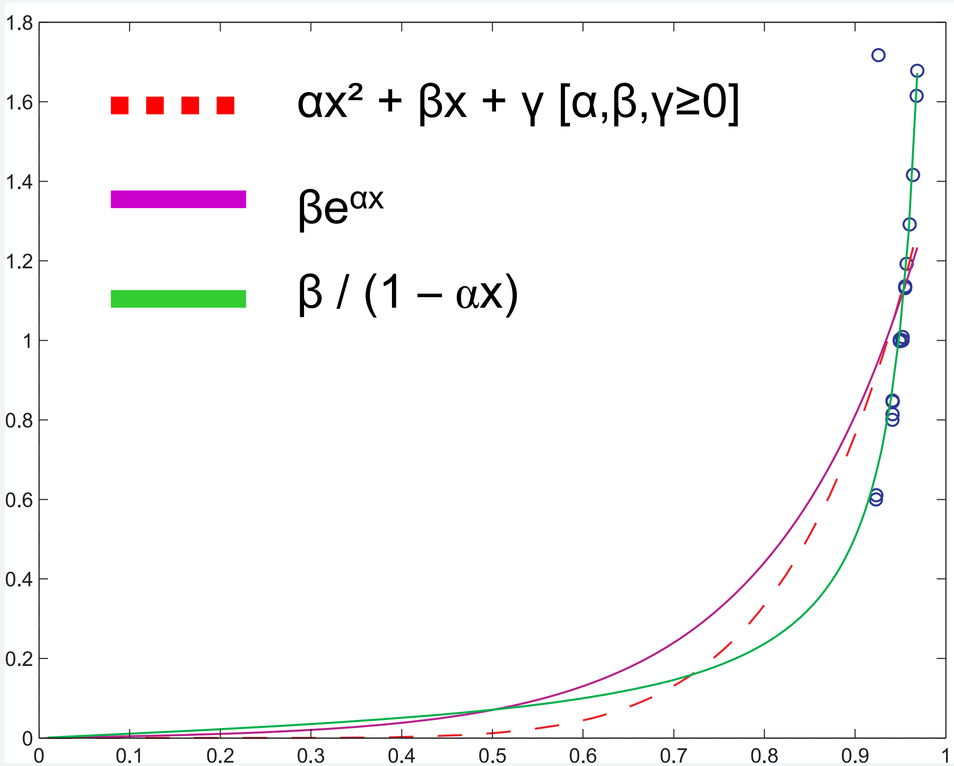


Base Matrices

- Base matrices **A** and **L**
- Normalized variants: $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$
- Example: $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$

Trust network Advogato

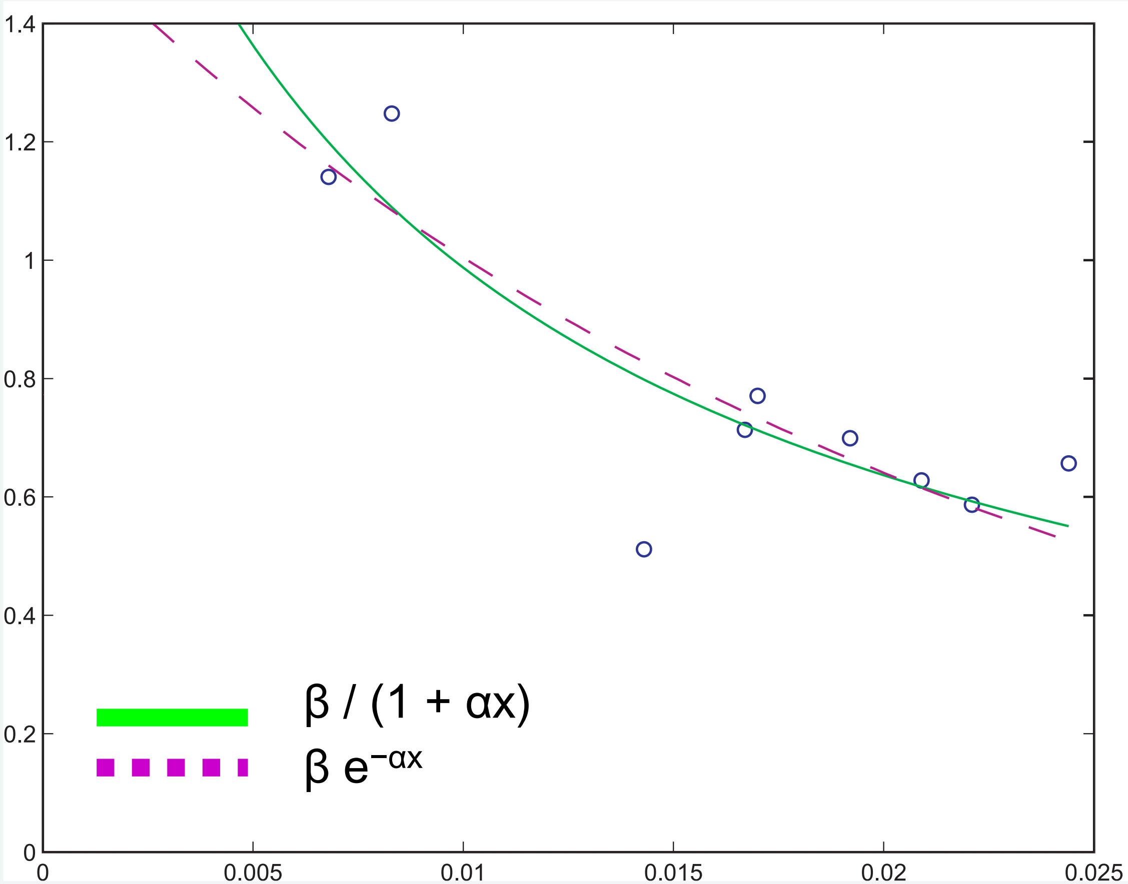
Note: ignore eigenvalue 1
(constant eigenvector)



Learning Laplacian Kernels

- Epinions (signed user-user network), using **L**

Note: $\boldsymbol{\Lambda}_{ii} > 0$ because the graph is signed and unbalanced

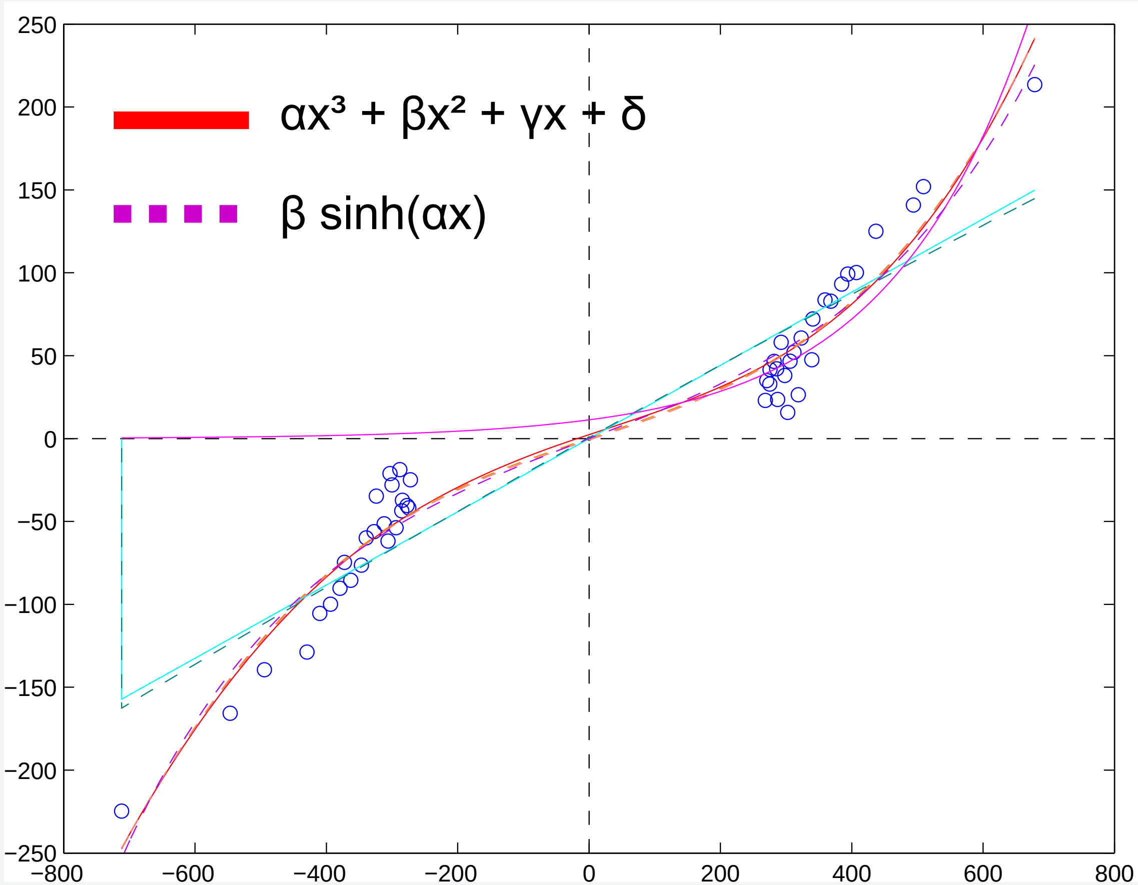


Almost Bipartite Networks

- Dating site LíbímSeTi.cz: (users rate users)

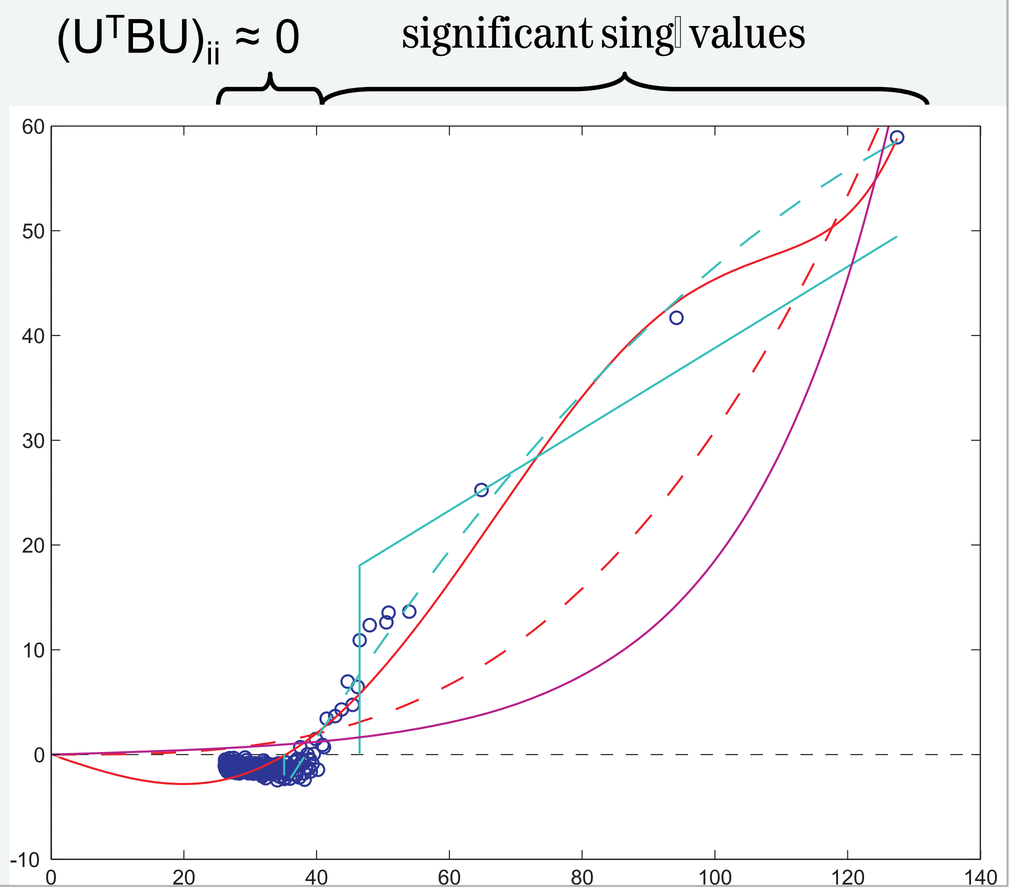
- Some networks are "almost" bipartite
- Plot has near central symmetry

- Bipartition: men/women



Learning the Reduced Rank *k*

- Some plots suggest a reduced rank *k*
- Example: MovieLens/SVD
- Learned: k = 14



Conclusion & Ongoing Research

Conclusions

- Many link prediction functions are spectral transformations
- Spectral transformations can be learned

Ongoing Research

- New link prediction function: sinh(A), odd von Neumann (pseudo-)kernel
- Signed graph Laplacian
- Other matrix decompositions
- Other norms
- More datasets

Thank You