Institute of Theoretical Physics
Chinese Academy of Sciences

Spectral detection of global structures in the noisy data

Pan Zhang
Institute of Theoretical Physics, Chinese Academy of Sciences

IoP 11.30 2016

Ref: PZ, *Advances In Neural Information Processing Systems 29*, 541 (2016)


RIP
34
13
17
MOLES

Machine Learning and Physics

Machine Learning and Physics

- **Machine learning —> physics**
  (Why? learn complex features by machines!)

Machine Learning and Physics

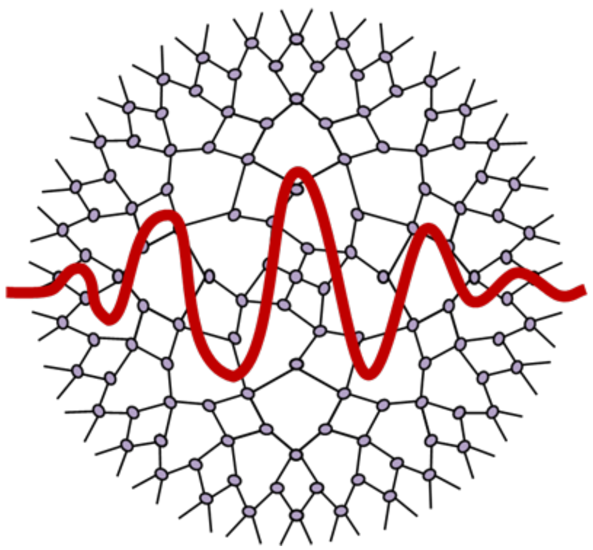
- **Machine learning —> physics**
  (Why? learn complex features by machines!)
  - Classification of matter using DNN

Machine Learning and Physics

- **Machine learning —> physics**
  (Why? learn complex features by machines!)
    - Classification of matter using DNN
    - Learning Hamiltonian using DNN

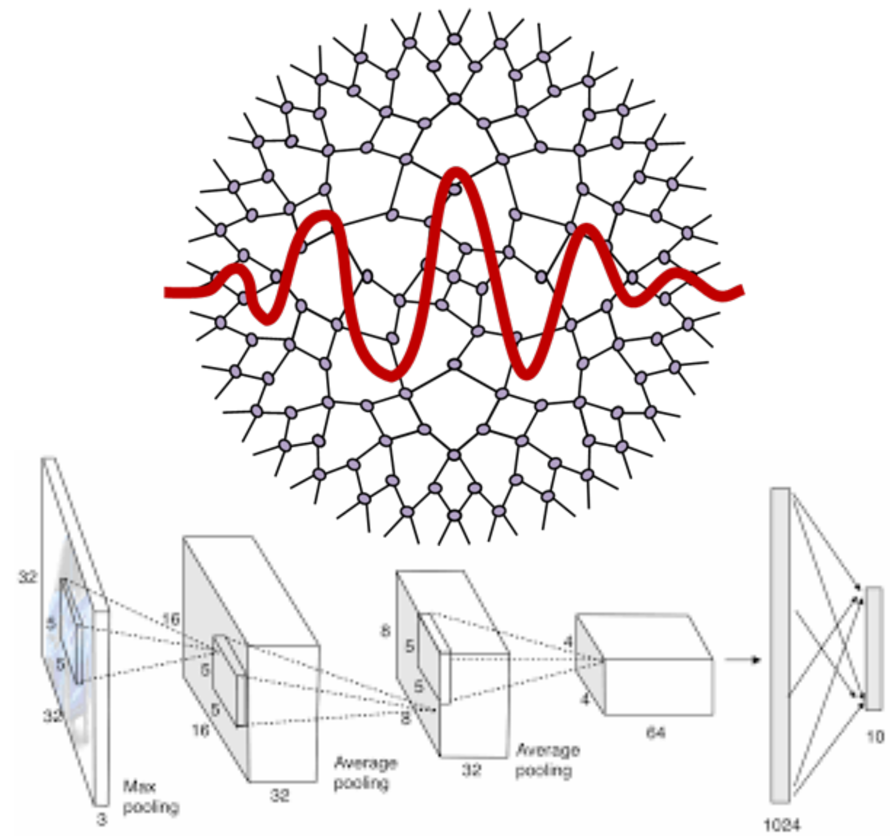
Machine Learning and Physics

- **Machine learning —> physics**
  (Why? learn complex features by machines!)
  - Classification of matter using DNN
  - Learning Hamiltonian using DNN
  - Tensor Networks



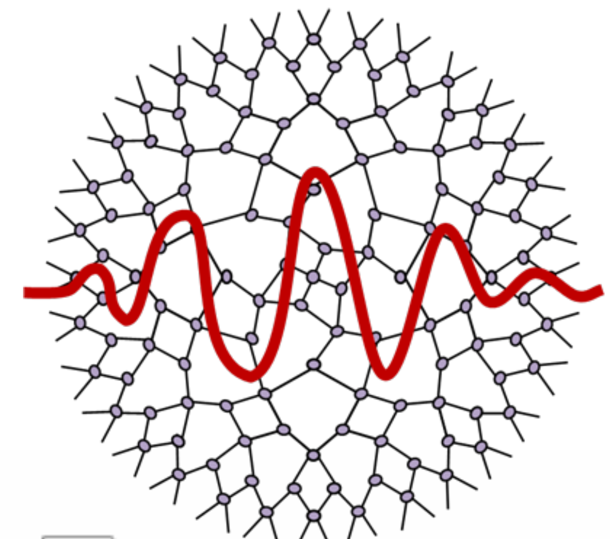
Machine Learning and Physics

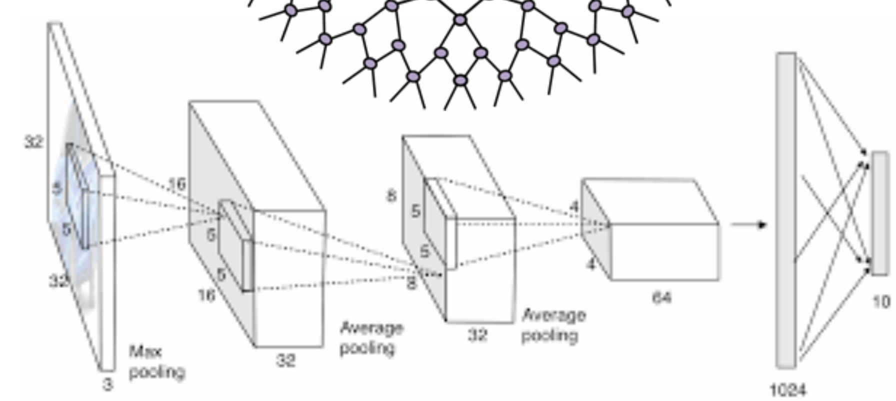
- **Machine learning —> physics**
  (Why? learn complex features by machines!)
  - Classification of matter using DNN
  - Learning Hamiltonian using DNN
  - Tensor Networks
  - Machine renormalization group?



Machine Learning and Physics

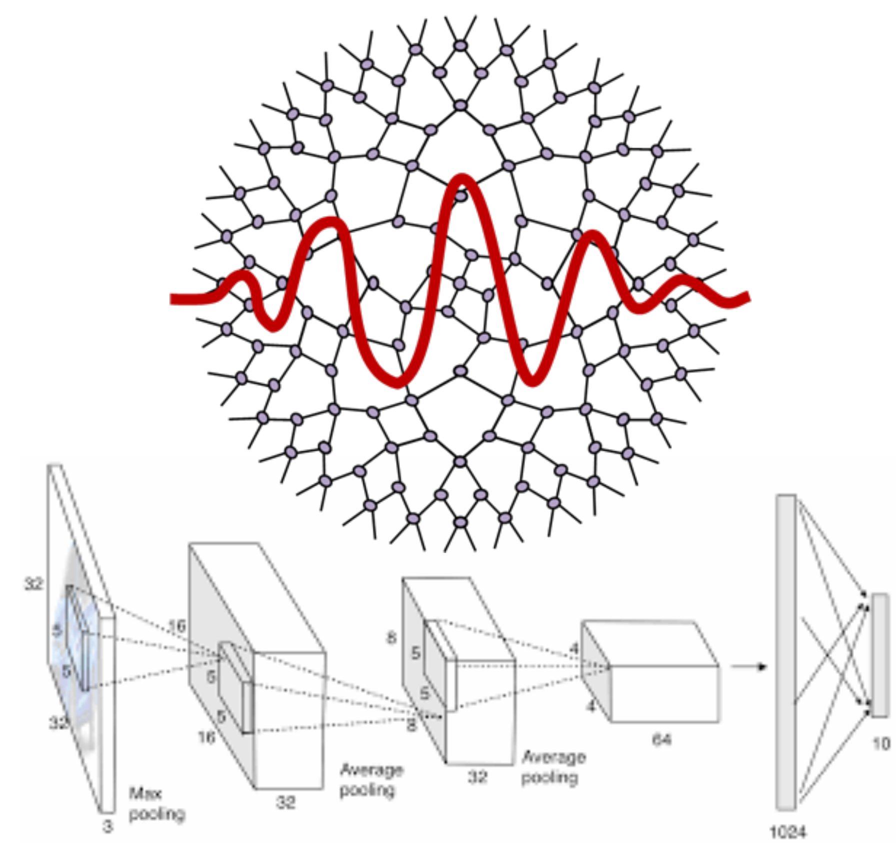
- **Machine learning —> physics**
  (Why? learn complex features by machines!)
  - Classification of matter using DNN
  - Learning Hamiltonian using DNN
  - Tensor Networks
  - Machine renormalization group?
- **Physics —> machine learning**





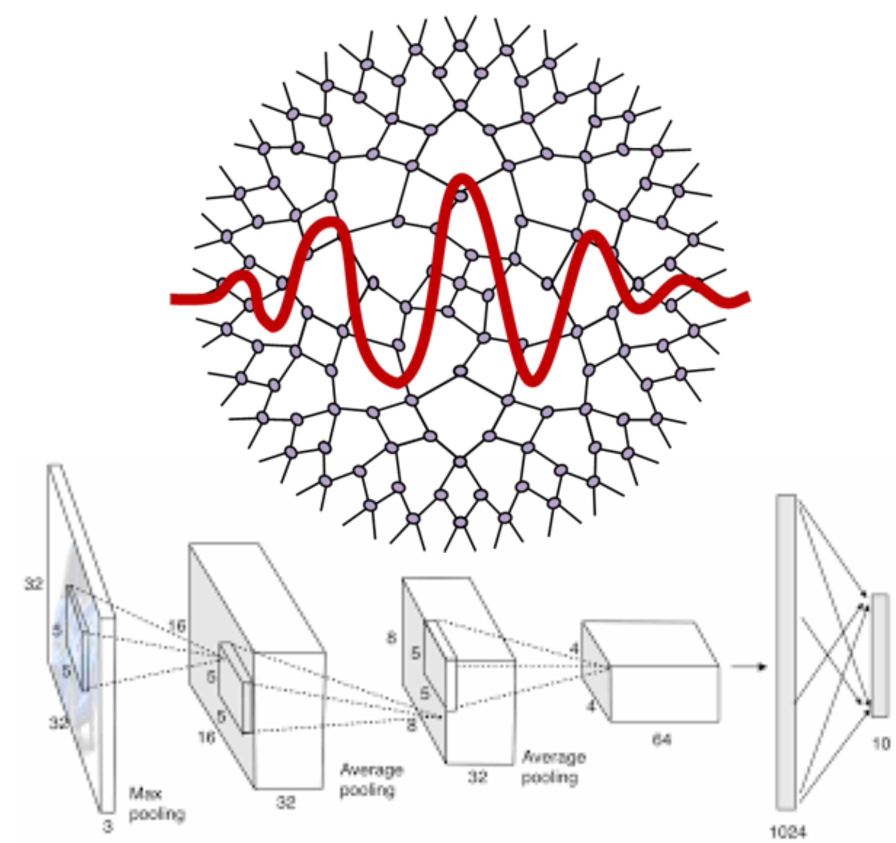
Machine Learning and Physics

- **Machine learning —> physics**
  (Why? learn complex features by machines!)
  - Classification of matter using DNN
  - Learning Hamiltonian using DNN
  - Tensor Networks
  - Machine renormalization group?
- **Physics —> machine learning**
  - Quantum information/computation



Machine Learning and Physics

- **Machine learning —> physics**
  (Why? learn complex features by machines!)
  - Classification of matter using DNN
  - Learning Hamiltonian using DNN
  - Tensor Networks
  - Machine renormalization group?



- **Physics —> machine learning**
  - Quantum information/computation
  - Spin Glass Theory, Various transitions, Approximate Bayesian inference

Machine Learning

- Supervised learning

- Unsupervised learning

Machine Learning

- Supervised learning
  - Classification
  - Regression
  - … …
- Unsupervised learning

Machine Learning

- Supervised learning
    - Classification
    - Regression
    - … …
- Unsupervised learning
    - Clustering
    - Dimensionality reduction
    - … …

Machine Learning

- Supervised learning
  - Classification
  - Regression
  - … …
- Unsupervised learning
  - Clustering
  - Dimensionality reduction
  - … …

- Semi-supervised Learning
- Active Learning
- Transfer Learning

Machine Learning Methods

- Supervised learning

- Unsupervised learning

Machine Learning Methods

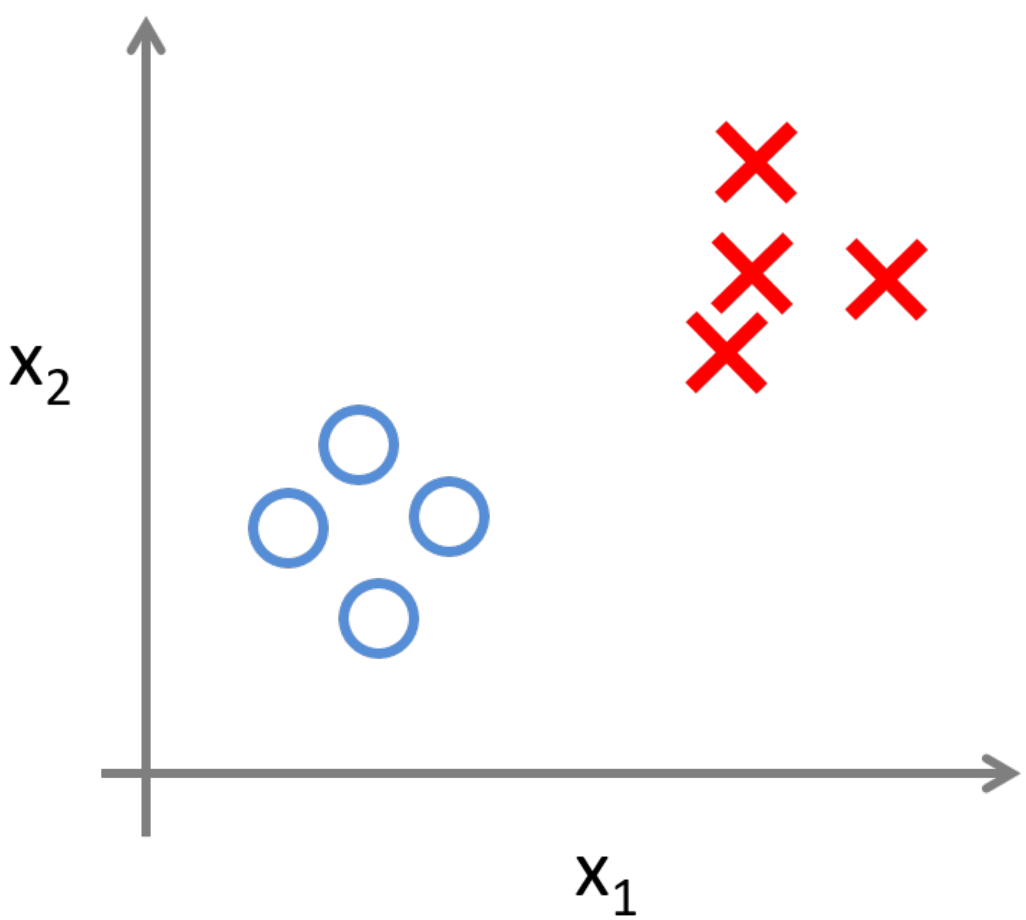
- Supervised learning
  - Deep neural networks
  - Support Vector Machines
  - Logistic Regression
  - … …
- Unsupervised learning

Machine Learning Methods

- Supervised learning
  - Deep neural networks
  - Support Vector Machines
  - Logistic Regression
  - … …
- Unsupervised learning
  - Principled Component Analysis
  - Singular Value Decompositions
  - Hidden Markov Models
  - Expectation-Maximization
  - Graphical Models
  - … …

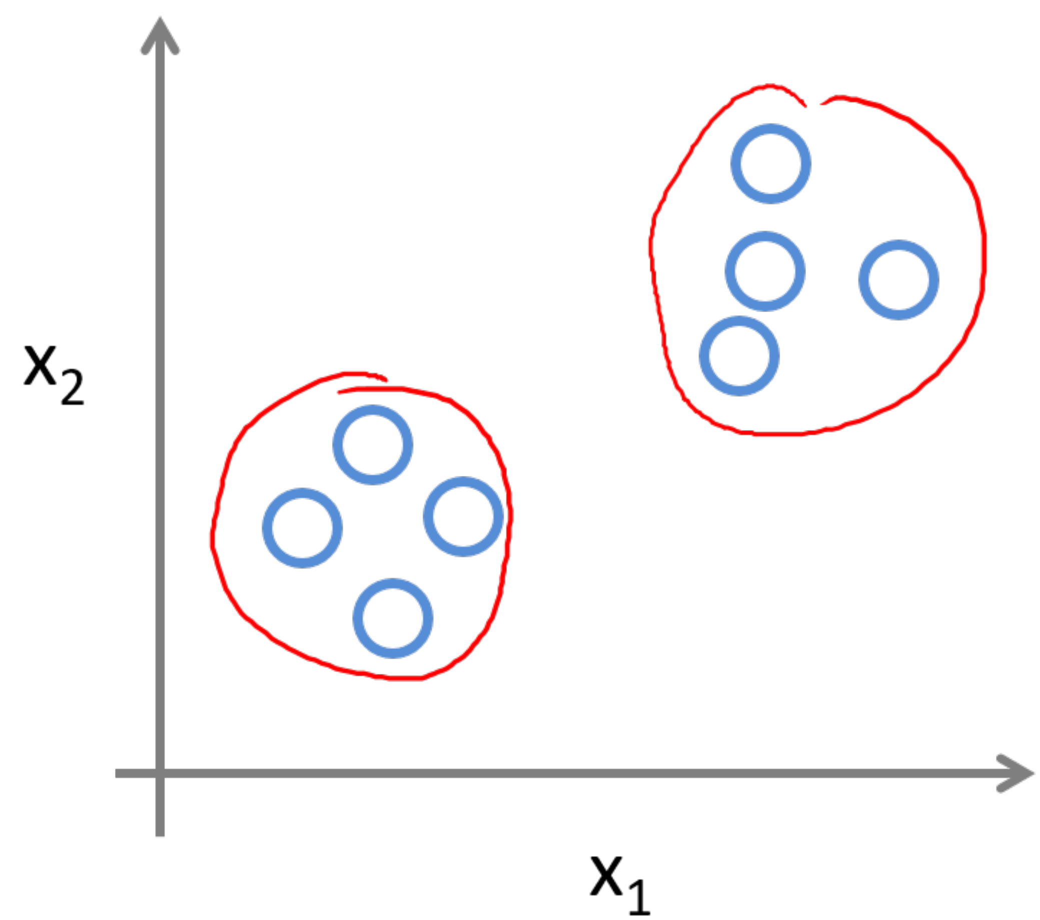
Supervised vs. Unsupervised learning

Supervised Learning



Predicting Labels

Unsupervised Learning



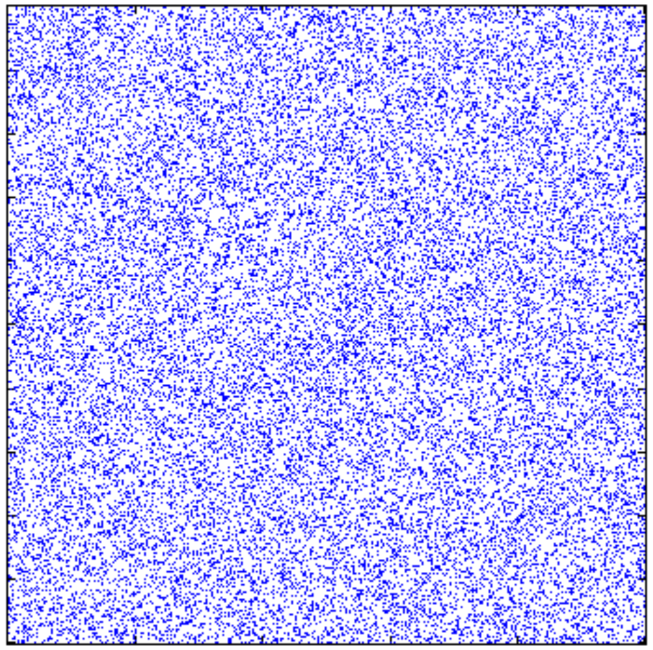
Finding structures in the data

Topic of today

Unsupervised learning: finding structures in the data matrix

Data matrices

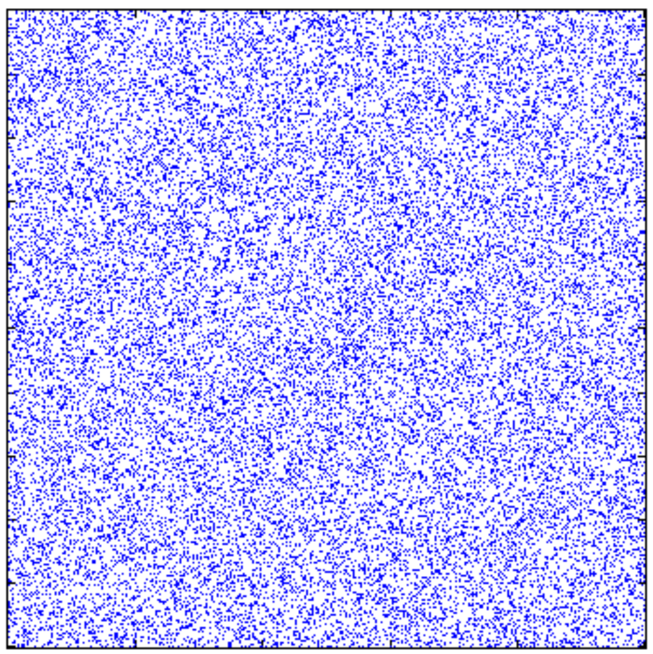
Data matrices

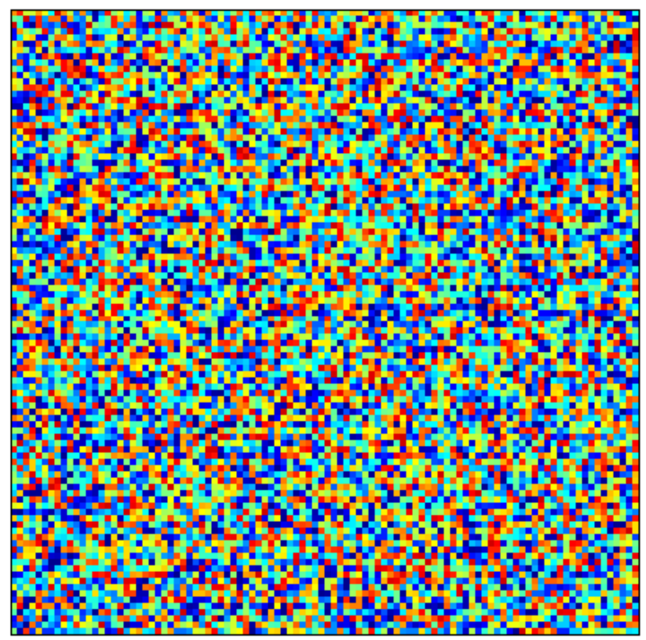


Network:
Adjacency matrix

$A \in \{0,1\}^{n \times n}$

Data matrices





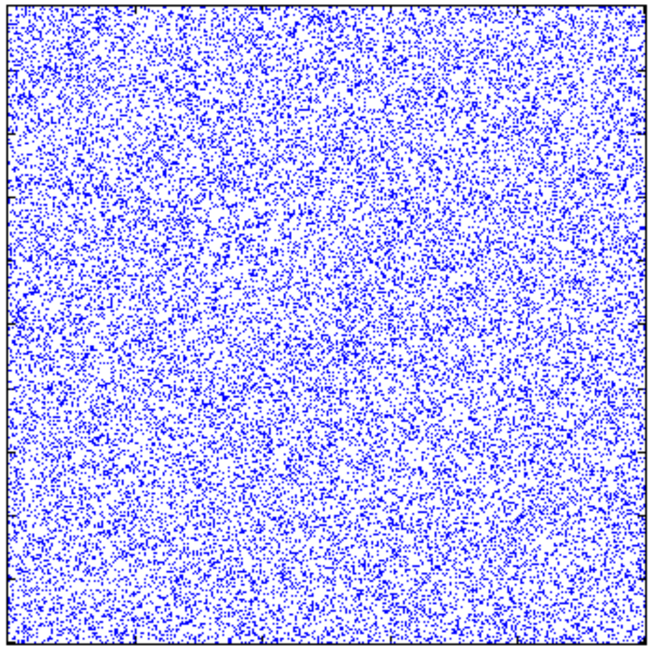
Network:
Adjacency matrix

$A \in \{0,1\}^{n \times n}$

Clustering:
Similarity matrix

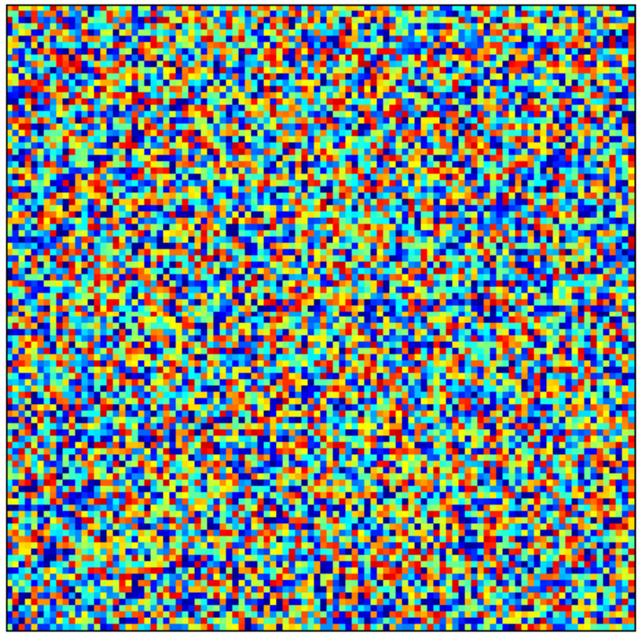
$A \in \mathbb{R}^{n \times n}$

Data matrices



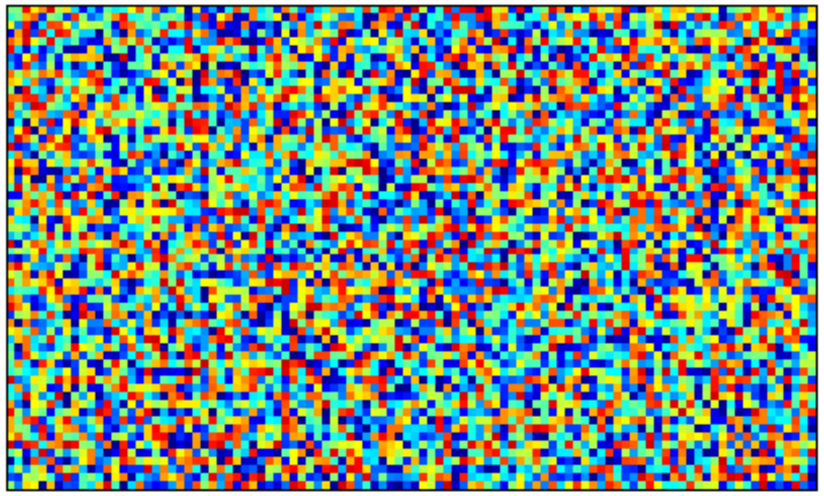
Network:
Adjacency matrix

$A \in \{0,1\}^{n \times n}$



Clustering:
Similarity matrix

$A \in \mathbb{R}^{n \times n}$

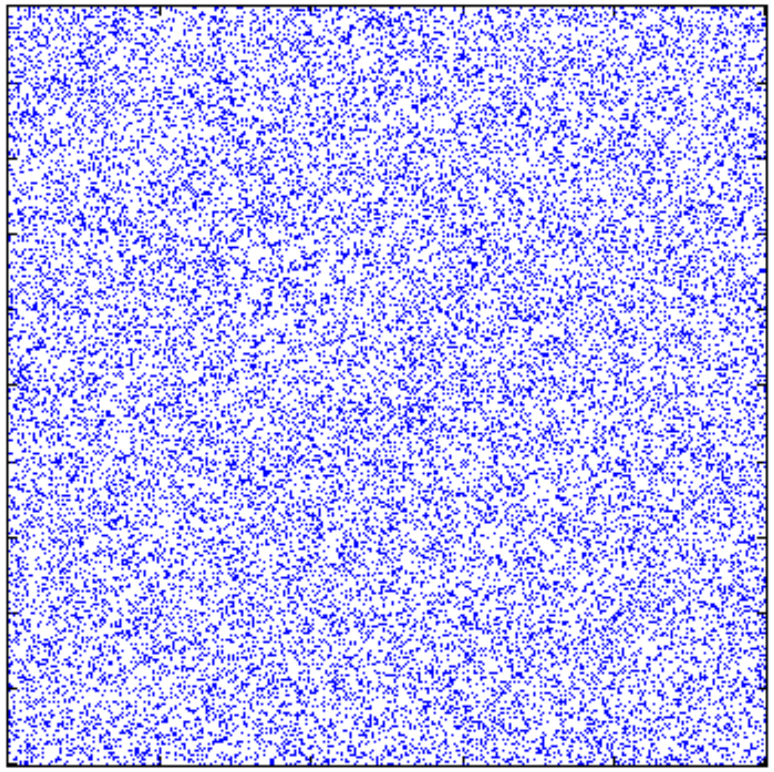


Recommendation:
Rating matrix

$A \in \mathbb{R}^{m \times n}$

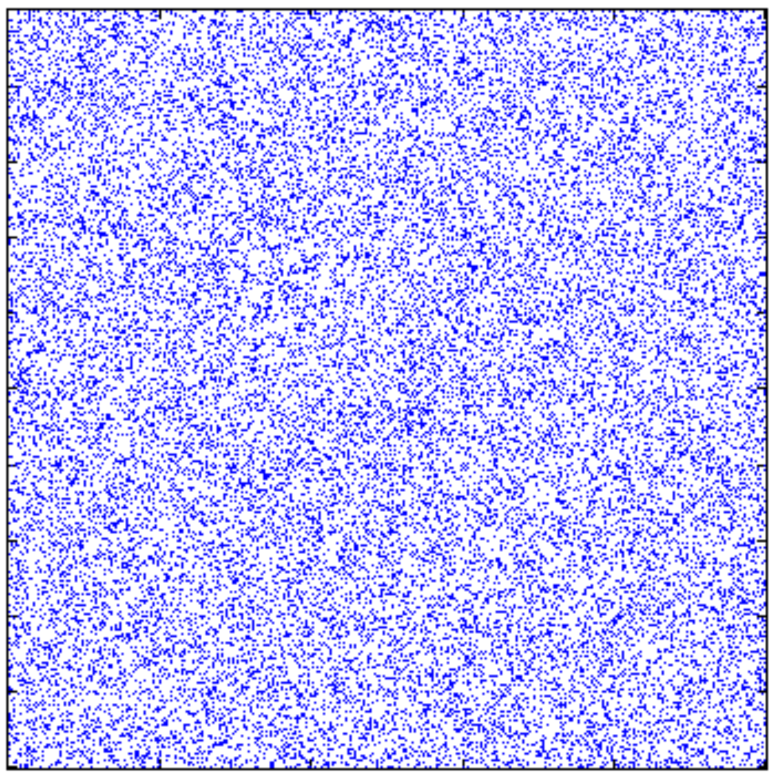
Community detection

Community detection



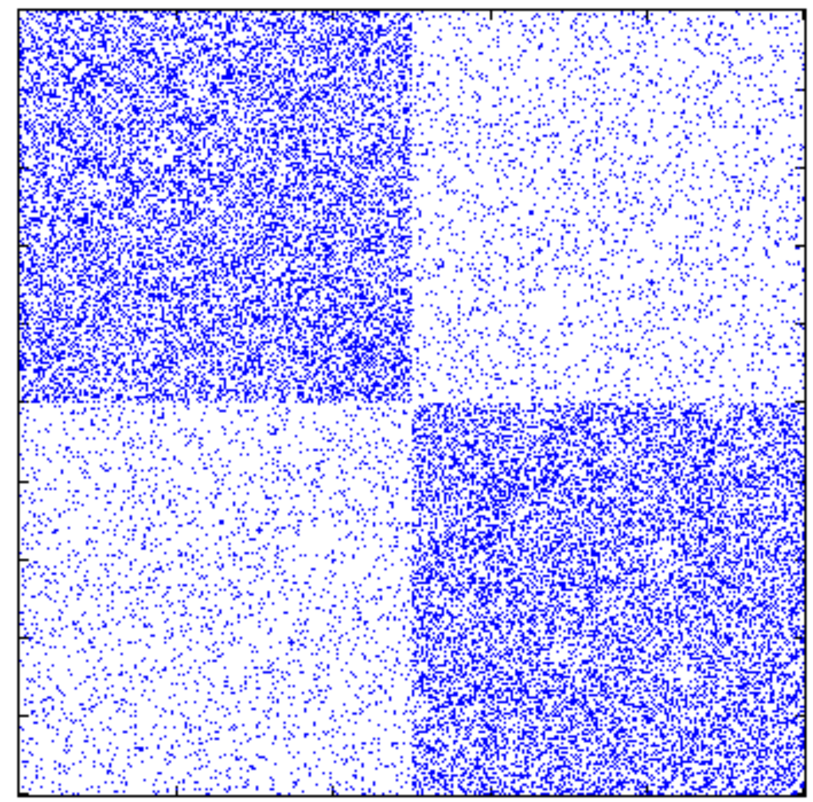
What you have:
DATA

Community detection



What you have:

DATA



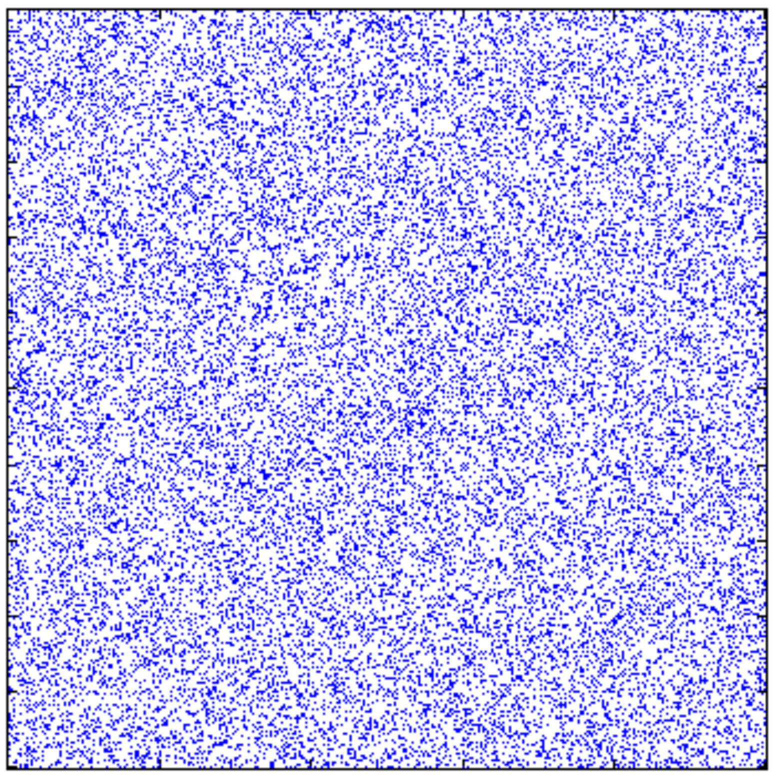
What you want:

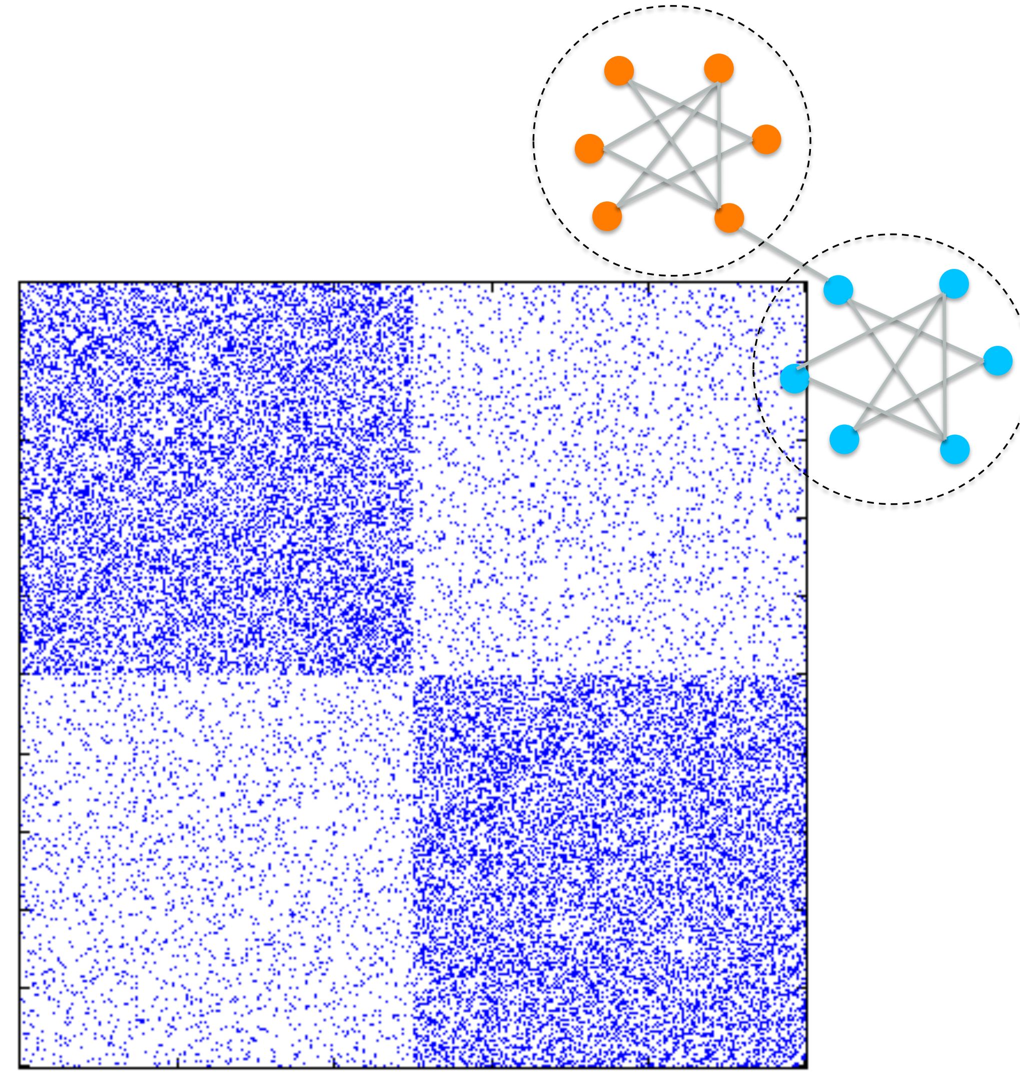
significant structure

which makes data different from noise

or different from ***random graphs***

Community detection



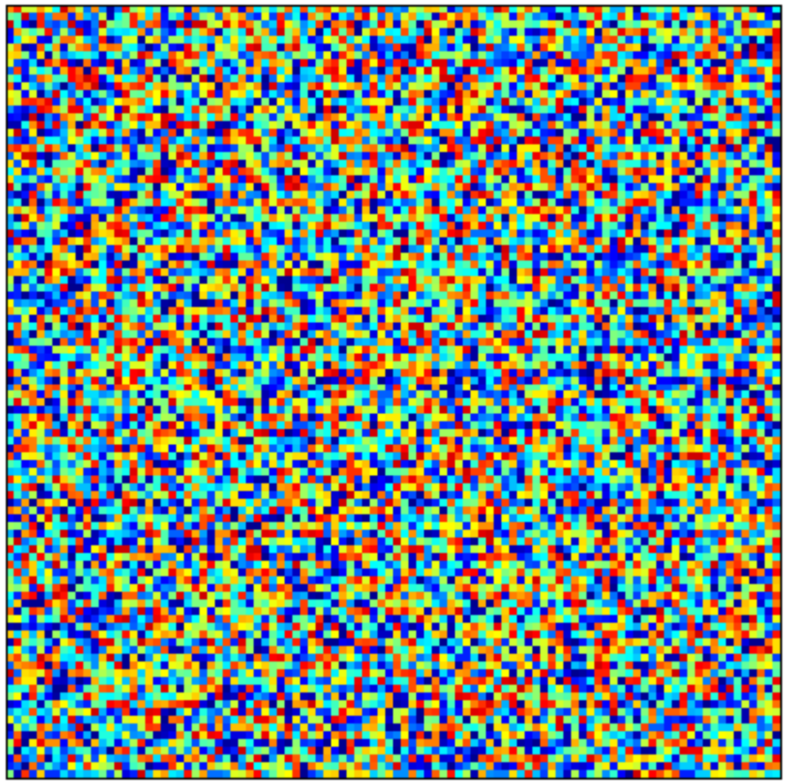


What you have:
DATA

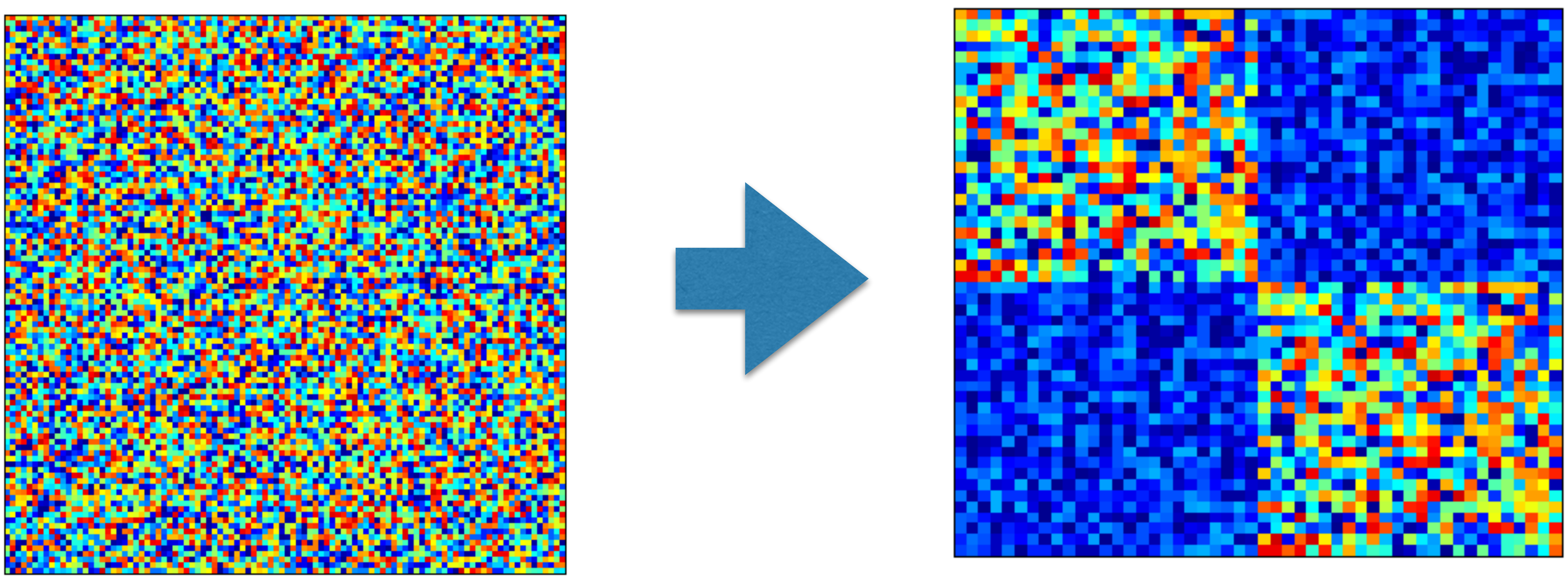
What you want:
significant structure
which makes data different from noise
or different from ***random graphs***

Clustering from similarities

Clustering from similarities

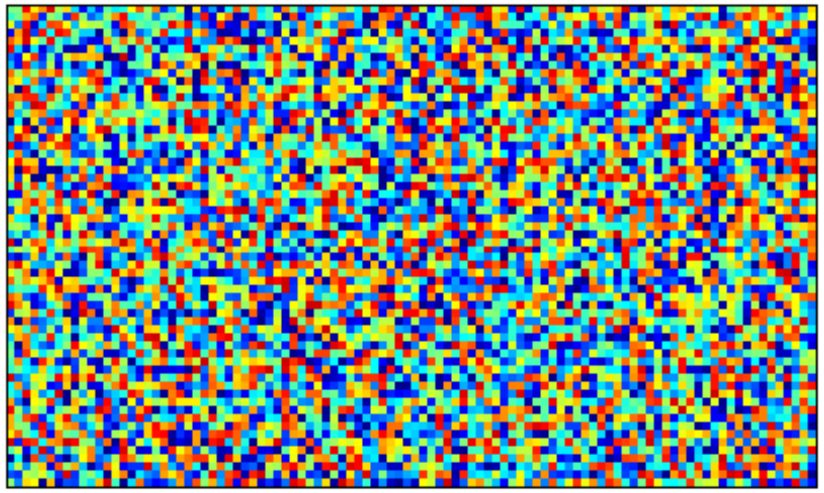


Clustering from similarities

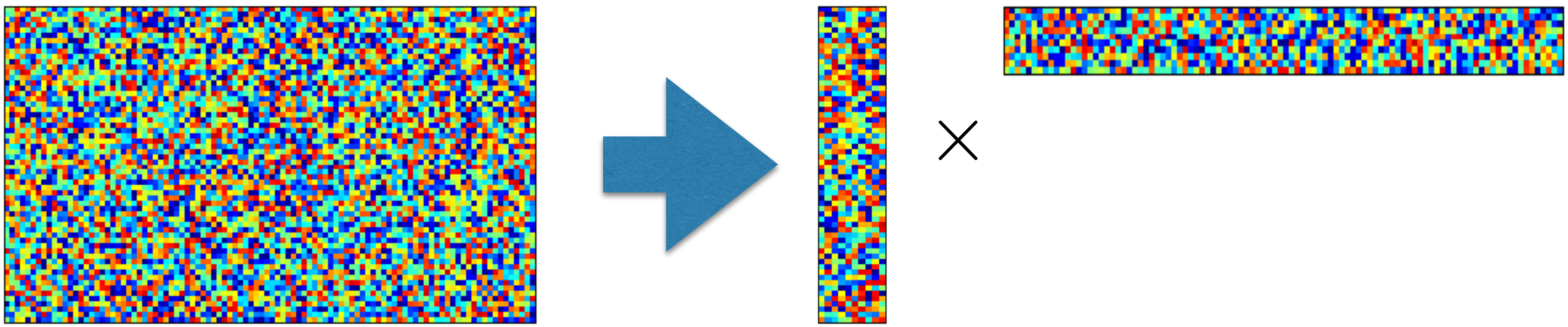


Low-rank matrix factorization

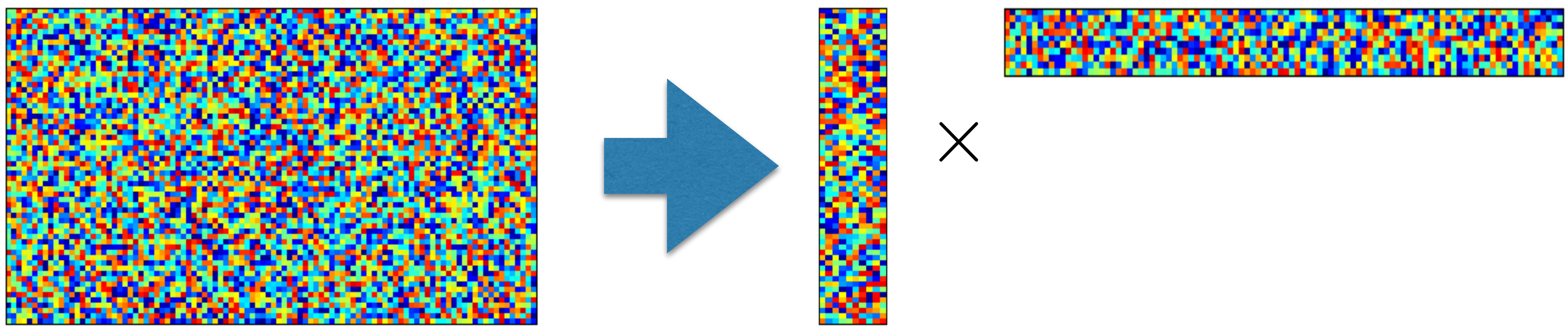
Low-rank matrix factorization



Low-rank matrix factorization



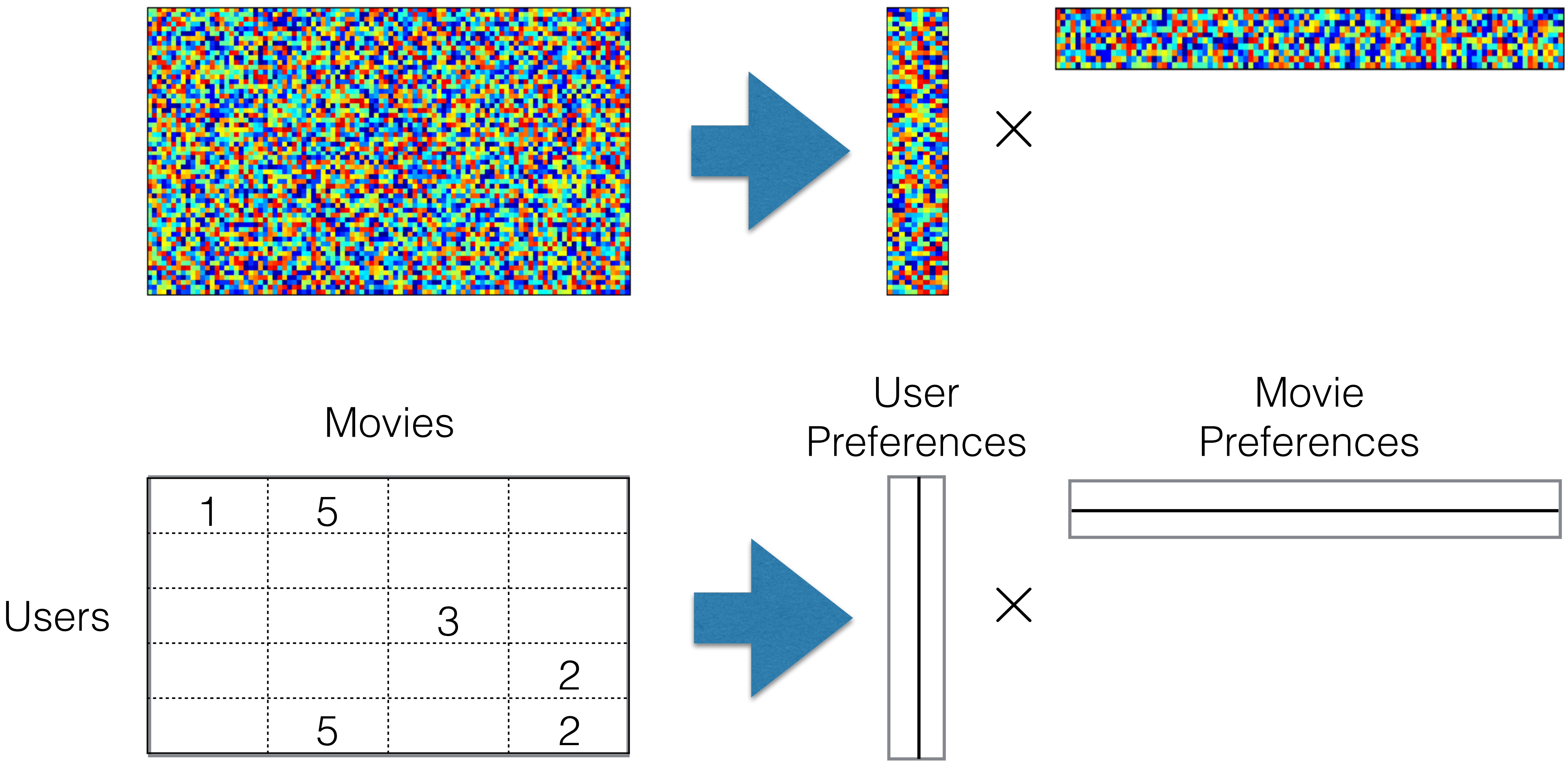
Low-rank matrix factorization



Movies

	1	5		
Users			3	
				2
		5		2

Low-rank matrix factorization

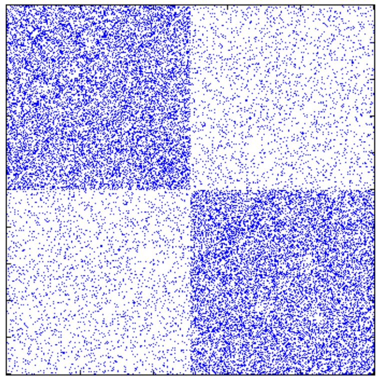


Spectral methods

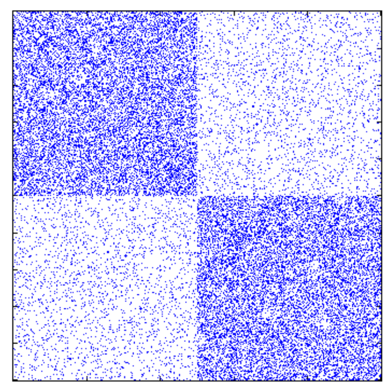
- Choose a matrix, such as
  - Data matrix A
  - Laplacians $L = D - A$
  - Normalized Laplacian $L_{\mathrm{sym}} = D^{-1/2} L D^{-1/2}$
  - Random walk matrix $P = D^{-1} A$
- Computer first several eigenvectors (or singular vectors) of the matrix.
- Construct clusters or low-rank approximations using the eigenvectors.

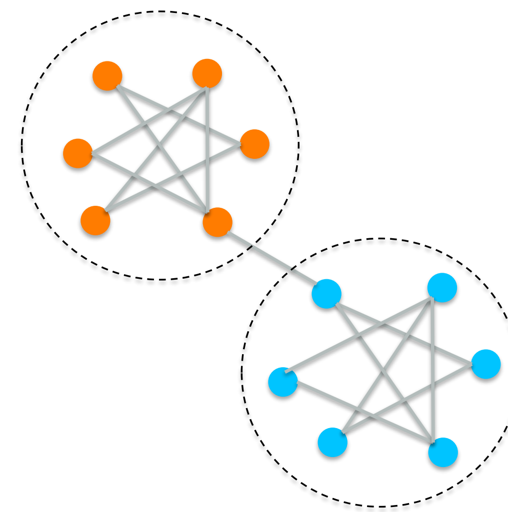
Why do they work?

Why do they work?

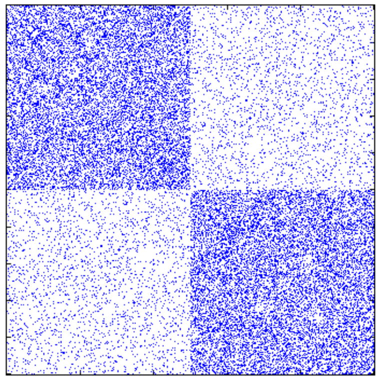


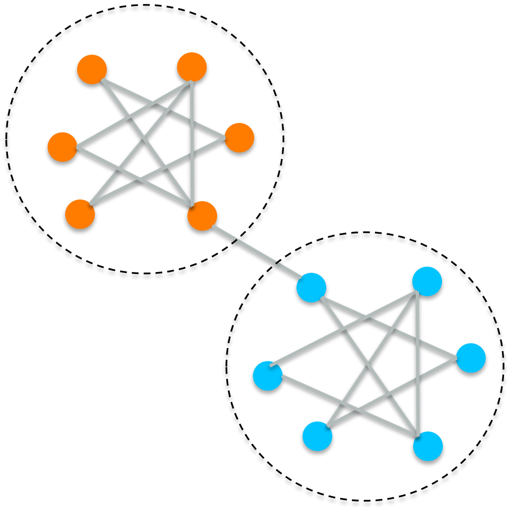
Why do they work?





Why do they work?

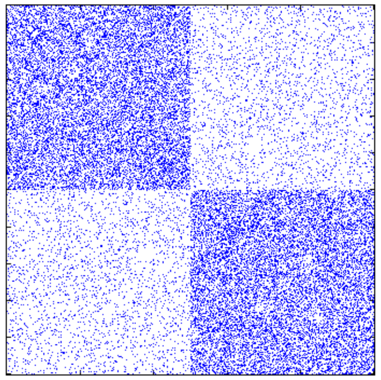


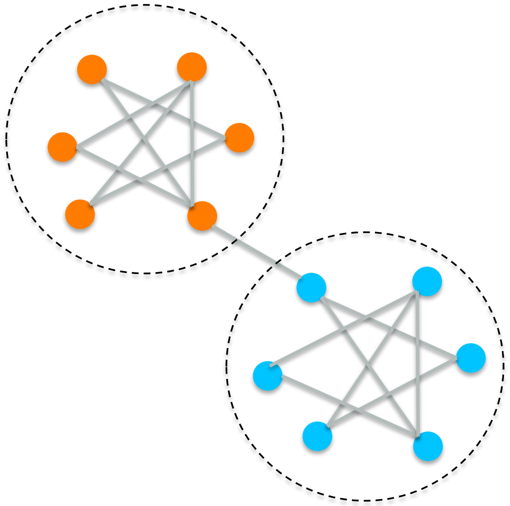


$$\hat{x} = \arg\max_x x^T A x$$
$$\text{S.T. } x \in \{-1, 1\}^n$$

Original Problem

Why do they work?





$$\hat{x} = \arg\max_x x^T A x$$
$$\text{S.T. } x \in \{-1, 1\}^n$$

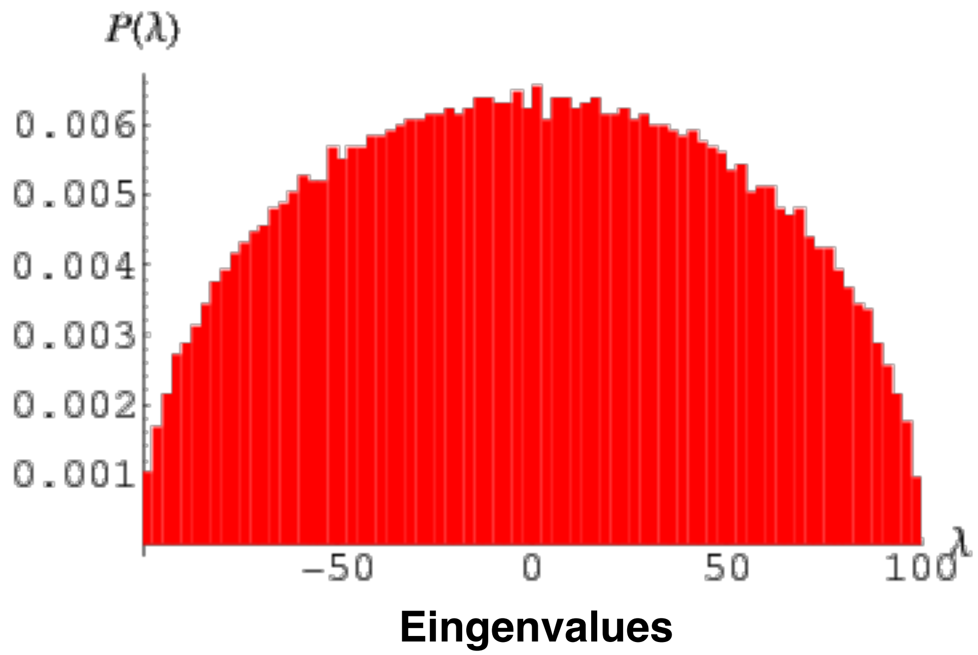
Original Problem

$$\hat{x} = \arg\max_x \frac{x^T A x}{x^T x}$$
$$\text{S.T. } x \in \mathbb{R}^n$$

Spectral relaxation

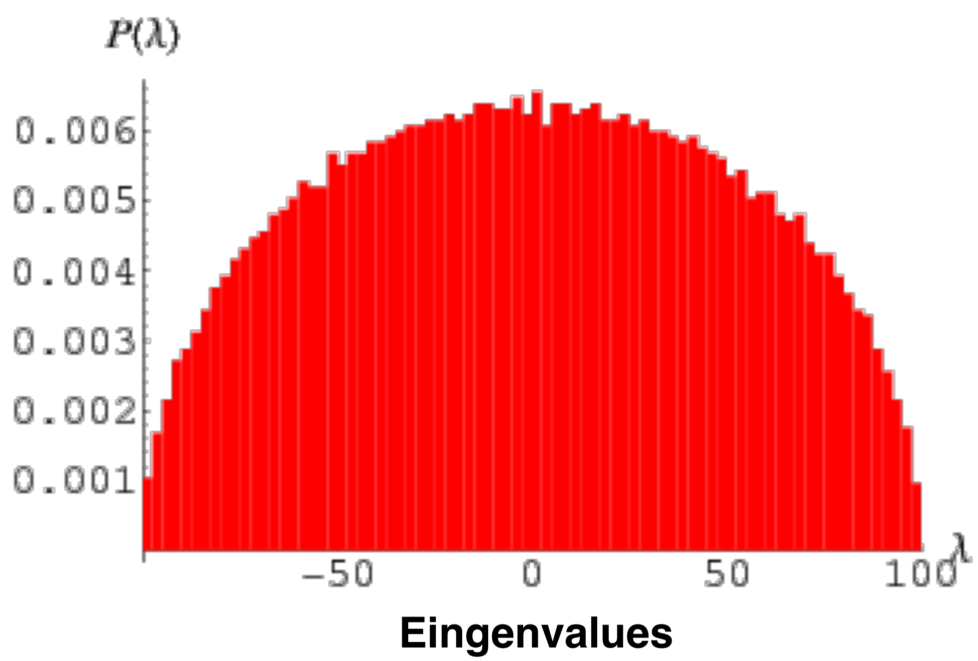
When do they work?

When do they work?

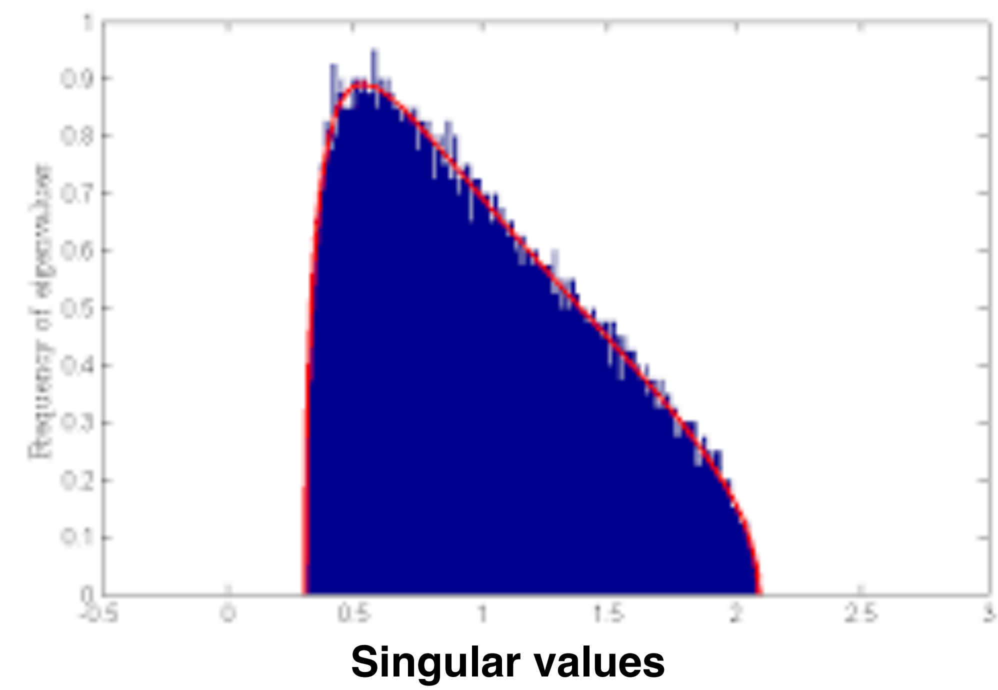


Wigner's semicircle law

When do they work?

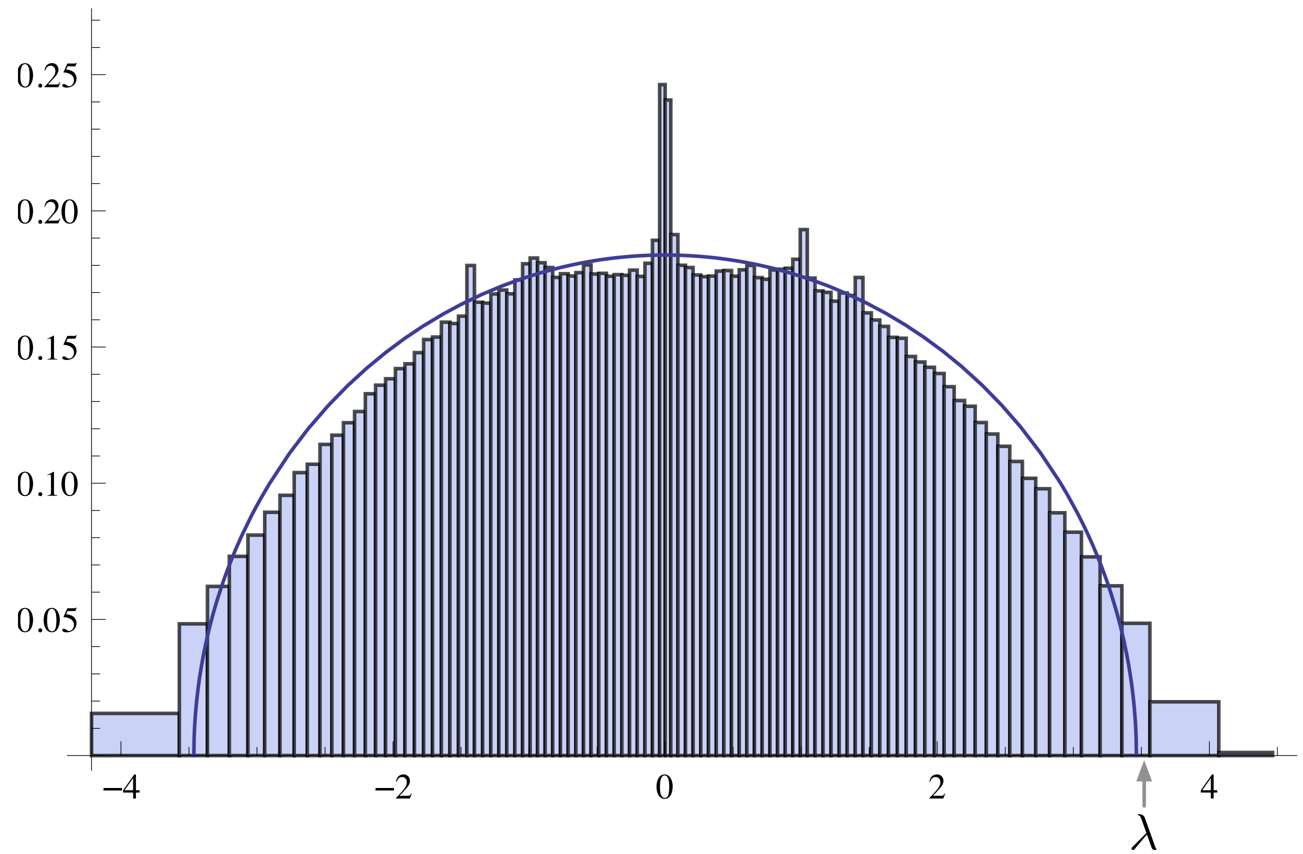


Wigner’s semicircle law



Marchenko Pastur law

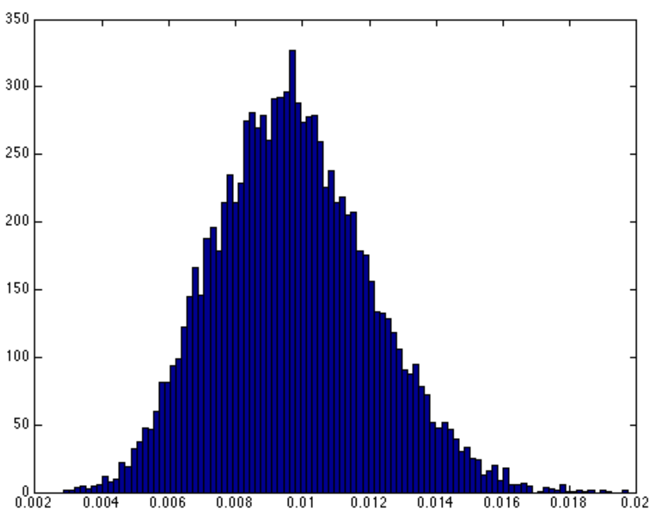
However they do not work well in large sparse matrices

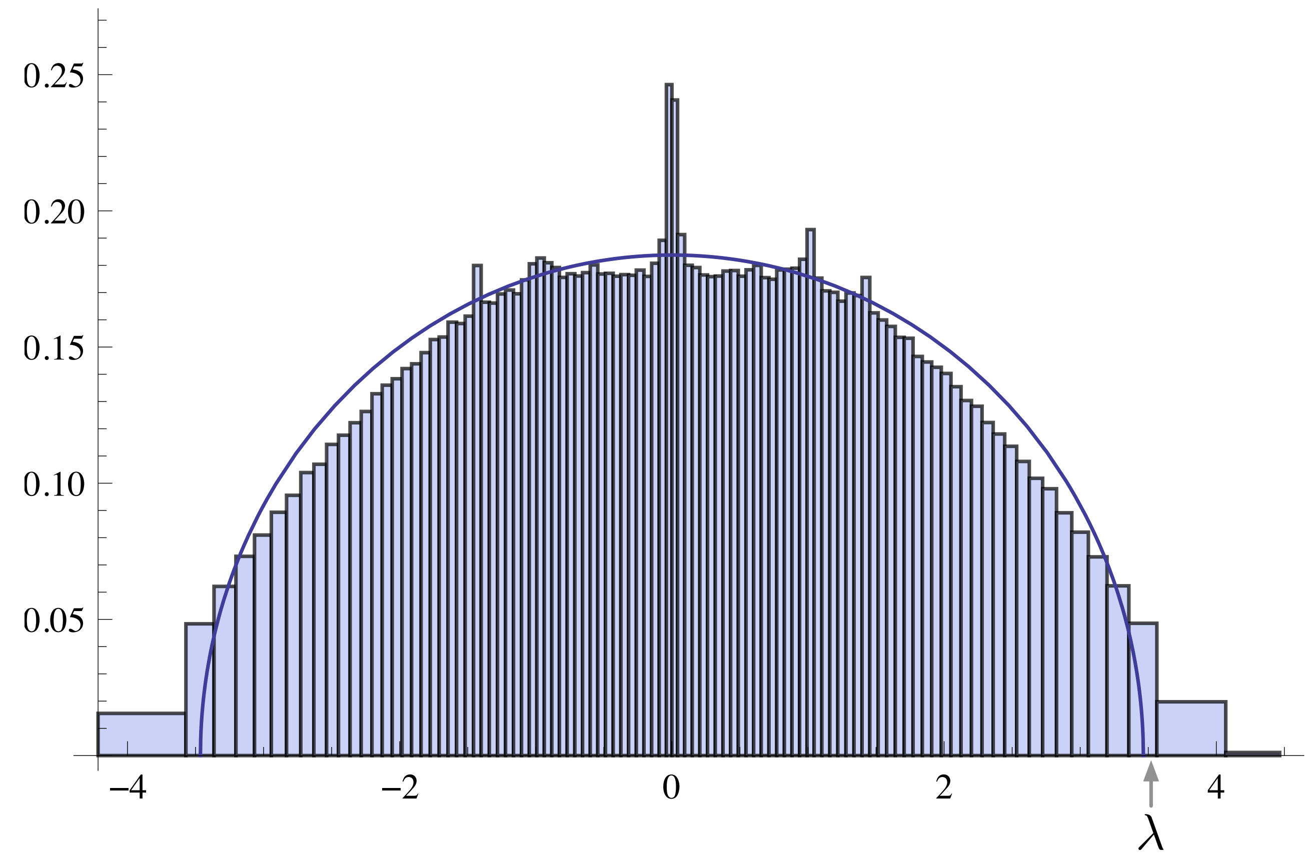


- Eigenvalues deviate from the semicircle rule
- Informative eigenvalues get lost in the bulk

Community detection: Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13'
Matrix Completion: Saade/Krzakala/Zdeborová NIPS 15'
Similarity Clustering: Saade/Krzakala/Zdeborová ISIT 16'

However they do not work well in large sparse matrices

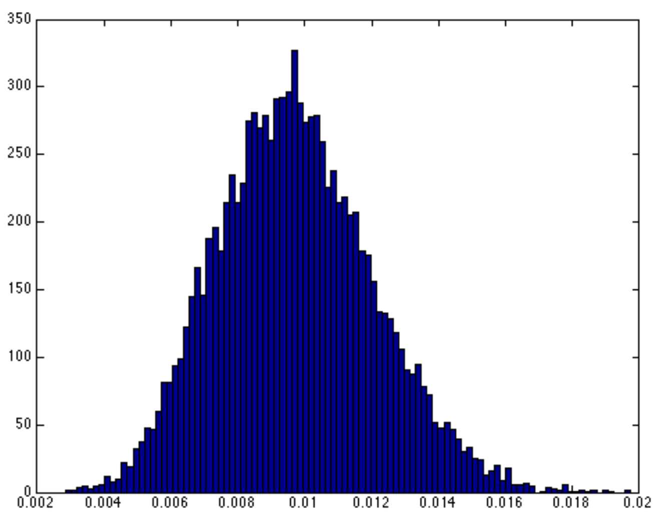


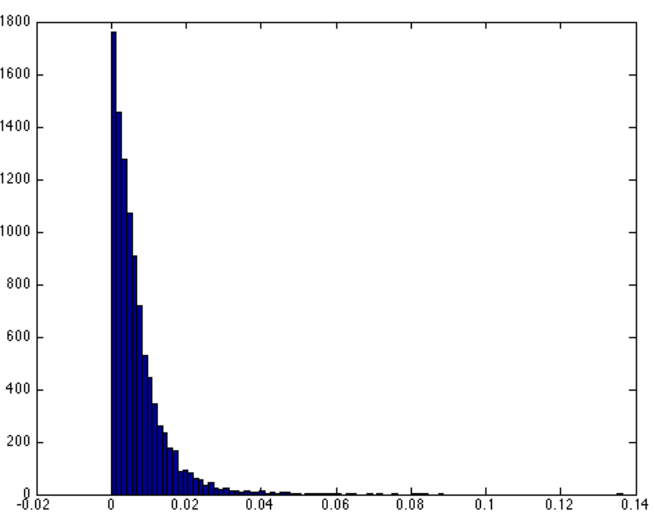


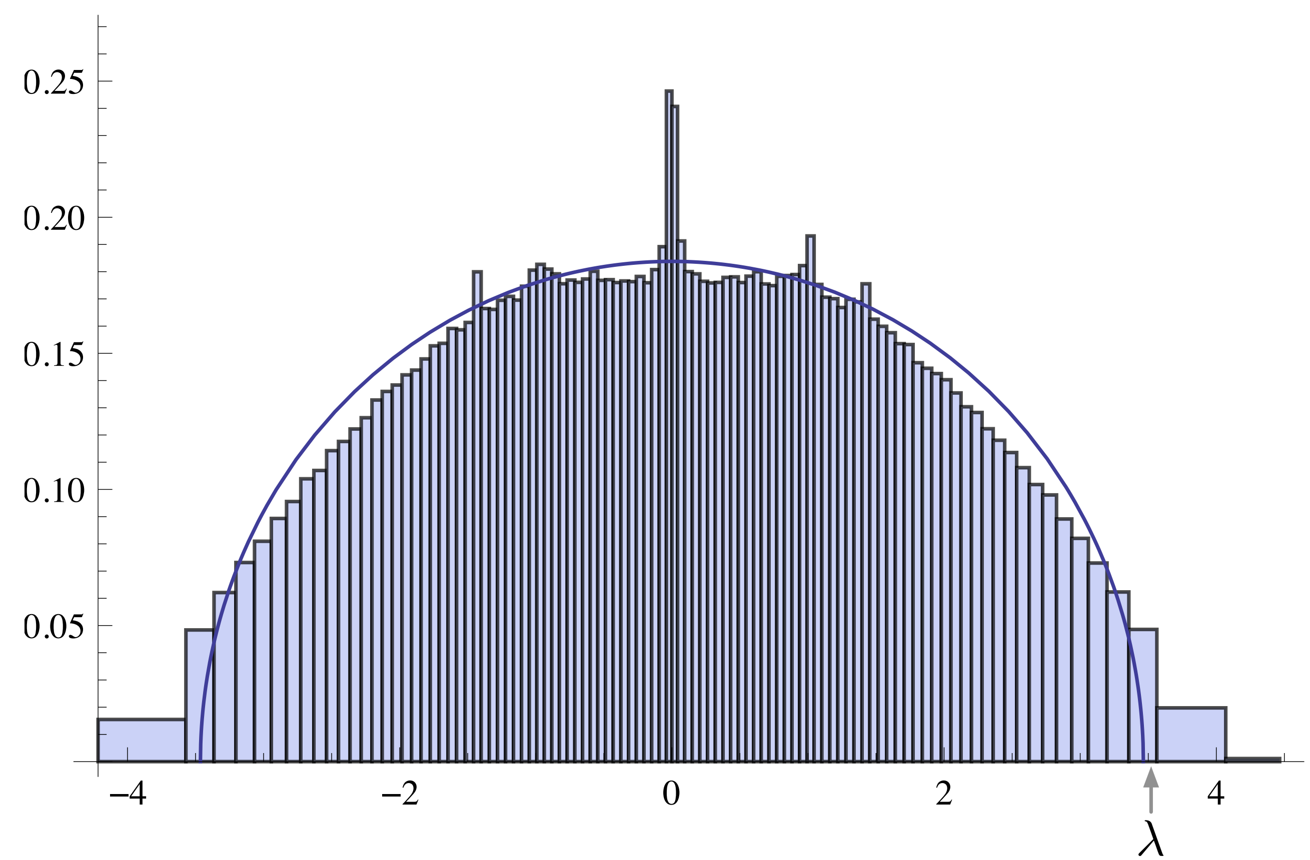
- Eigenvalues deviate from the semicircle rule
- Informative eigenvalues get lost in the bulk

Community detection: Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13'
Matrix Completion: Saade/Krzakala/Zdeborová NIPS 15'
Similarity Clustering: Saade/Krzakala/Zdeborová ISIT 16'

However they do not work well in large sparse matrices







- Eigenvalues deviate from the semicircle rule
- Informative eigenvalues get lost in the bulk

Community detection: Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13'
Matrix Completion: Saade/Krzakala/Zdeborová NIPS 15'
Similarity Clustering: Saade/Krzakala/Zdeborová ISIT 16'

Reason for the deviation: localization

- For adjacency matrix of Erdős–Rényi random graphs

$$d_{max} \approx \frac{\log n}{\log \log n}$$

$$\lambda_{\max}^2 \geq \frac{x^T A^T A x}{x^T x}$$

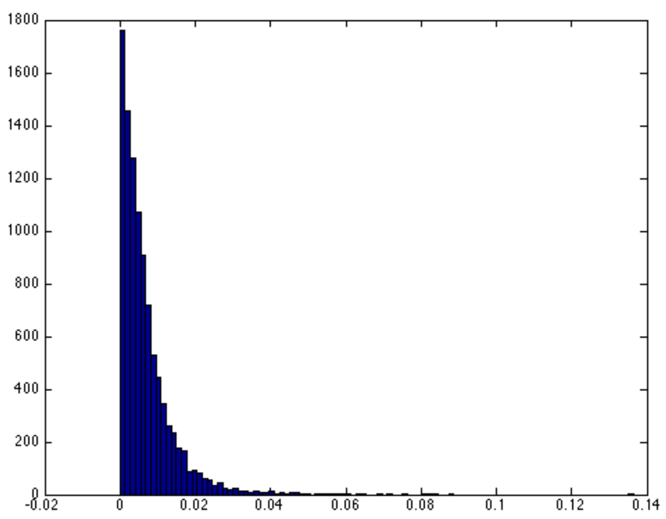
$$x = \{0, 0, 0...1...0, 0, 0\}$$

$$\lambda_{max} \geq \sqrt{d_{\max}} \approx \sqrt{\frac{\log n}{\log \log n}}$$

Localization on large-degree nodes

Reason for the deviation: localization

- For adjacency matrix of Erdős–Rényi random graphs

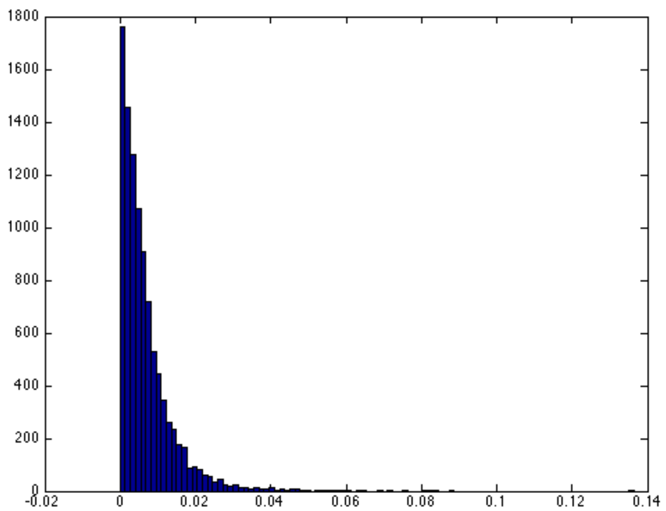


$$\left.\begin{aligned} d_{max} &\approx \frac{\log n}{\log\log n} \\ \lambda_{\max}^2 &\geq \frac{x^T A^T A x}{x^T x} \\ x &= \{0,0,0\ldots1\ldots0,0,0\} \end{aligned}\right\} \quad \lambda_{max} \geq \sqrt{d_{\max}} \approx \sqrt{\frac{\log n}{\log\log n}}$$

Localization on large-degree nodes

Reason for the deviation: localization

- For adjacency matrix of Erdős–Rényi random graphs



$$\left.\begin{aligned} d_{max} &\approx \frac{\log n}{\log\log n} \\ \lambda^2_{\max} &\geq \frac{x^T A^T A x}{x^T x} \\ x &= \{0,0,0...1...0,0,0\} \end{aligned}\right\} \quad \lambda_{max} \geq \sqrt{d_{\max}} \approx \sqrt{\frac{\log n}{\log\log n}}$$

Localization on large-degree nodes

- For matrices $P = D^{-1/2}A \qquad \tilde{A} = D^{-1/2}AD^{-1/2}$

$$L_{\text{sym}} = D^{-1/2}(D-A)D^{-1/2}$$

Localization on dangling sub-graphs

Localization in physics:

Localization in physics:

- Wavefunction amplitudes distribution: Inverse Participation Ratio (**IPR**).

$$\mathrm{IPR} = \sum_i |\psi_i|^4$$

Localization in physics:

- Wavefunction amplitudes distribution: Inverse Participation Ratio (**IPR**).

$$\mathrm{IPR} = \sum_i |\psi_i|^4$$

- Entangle entropy (between two subsystems)

$$S_2 = -\log \mathrm{tr}(\rho^2)$$

  - Low entanglement entropy <-> Localization.

  - Area Law, ground state.

Localization in physics:

- Wavefunction amplitudes distribution: Inverse Participation Ratio (**IPR**).

$$\mathrm{IPR} = \sum_i |\psi_i|^4$$

- Entangle entropy (between two subsystems)

$$S_2 = -\log \mathrm{tr}(\rho^2)$$

  - Low entanglement entropy <-> Localization.
  - Area Law, ground state.

- Energy-Level Statistic (Eigenvalue statistics)
  Poisson distribution vs. Wigner Dyson distribution

Approaches for localizations in sparse matrices

Approaches for localizations in sparse matrices

- Trimming: Remove rows and columns with large degree/ weights
  [Keshavan/Montanari/Oh 09']
  [Coja-Oghlan 10']

Approaches for localizations in sparse matrices

- Trimming: Remove rows and columns with large degree/weights
  [Keshavan/Montanari/Oh 09']
  [Coja-Oghlan 10']

- Teleportation: (rank-one regularizations)
  [Joseph/Yu 13']
  [Amini/Chen/Bickel/Levina 13']
  [Qin/Rohe 13']
  [Lei/Rinaldo 14']
  [Le/Levina/Vershynin 15']

Approaches for localizations in sparse matrices

- Trimming: Remove rows and columns with large degree/weights
  [Keshavan/Montanari/Oh 09']
  [Coja-Oghlan 10']

- Teleportation: (rank-one regularizations)
  [Joseph/Yu 13']
  [Amini/Chen/Bickel/Levina 13']
  [Qin/Rohe 13']
  [Lei/Rinaldo 14']
  [Le/Levina/Vershynin 15']

- Using non-backtracking matrix and Bethe Hessian
  [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang 2013]
  [Saade/Krzakala/Zdeborová 14']
  [Saade/Krzakala/Zdeborová 15']
  [Saade/Lelarge/Krzakala/Zdeborová 16']

Trimming

- Usually works in practice

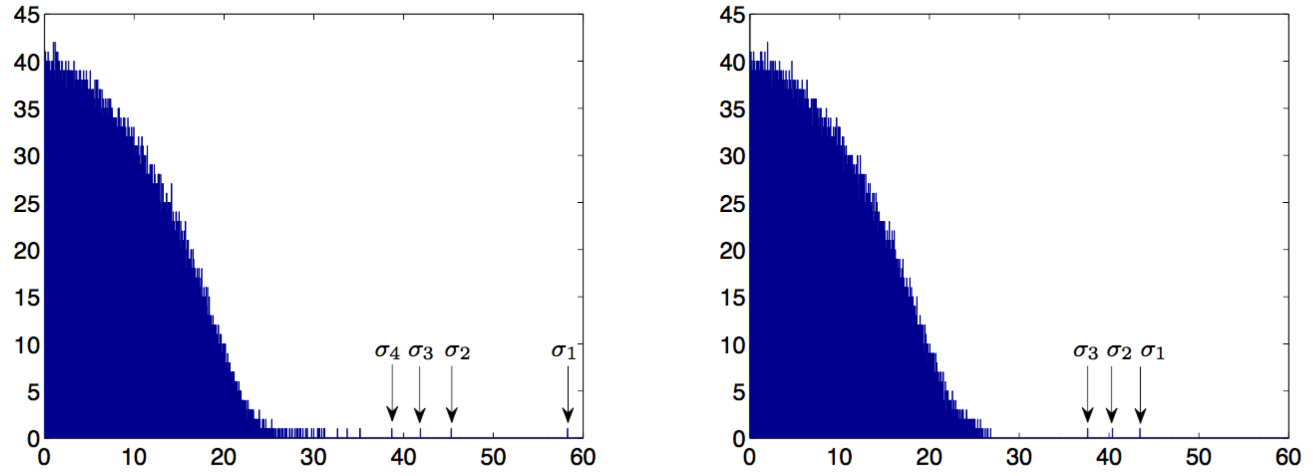


Figure 1: Histogram of the singular values of a partially revealed matrix M^E before trimming (left) and after trimming (right) for $10^4 \times 10^4$ random rank-3 matrix M with $\epsilon = 30$ and $\Sigma = \mathrm{diag}(1, 1.1, 1.2)$. After trimming the underlying rank-3 structure becomes clear. Here the number of revealed entries per row follows a heavy tail distribution with $\mathbb{P}\{N = k\} = \mathrm{const.}/k^3$.

Figure taken from Keshavan/Montanari/Oh 09’

Trimming

- Usually works in practice

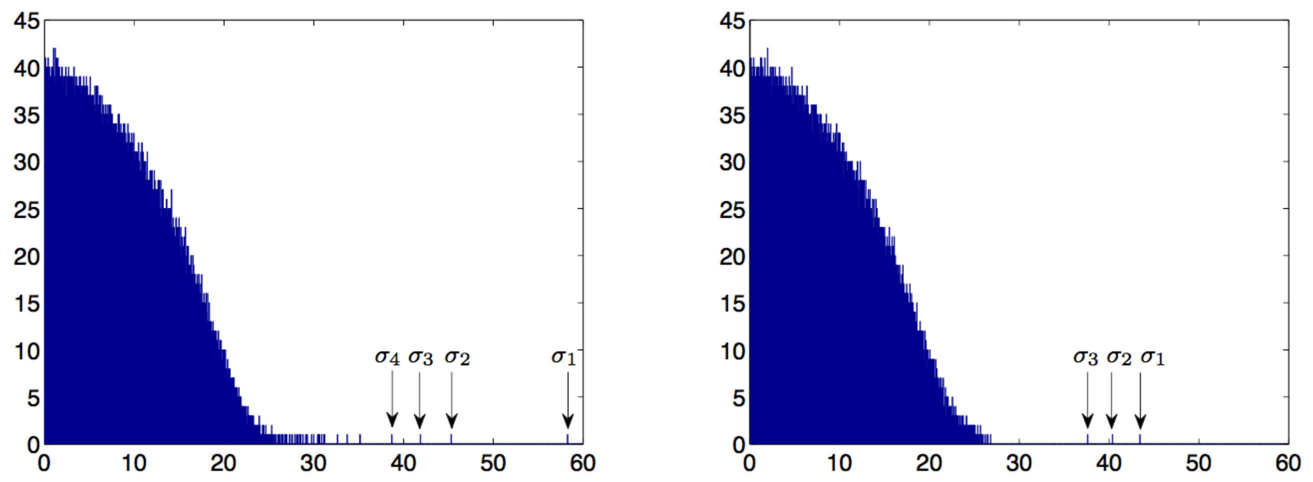


Figure 1: Histogram of the singular values of a partially revealed matrix M^E before trimming (left) and after trimming (right) for $10^4 \times 10^4$ random rank-3 matrix M with $\epsilon = 30$ and $\Sigma = \mathrm{diag}(1, 1.1, 1.2)$. After trimming the underlying rank-3 structure becomes clear. Here the number of revealed entries per row follows a heavy tail distribution with $\mathbb{P}\{N = k\} = \mathrm{const.}/k^3$.

Figure taken from Keshavan/Montanari/Oh 09'

- However in general suboptimal [Coja-Oghlan 10'], usually performs worse than other methods.

Teleportation

$$\hat{A} = D^{-1/2} A D^{-1/2} + z\mathbf{1}\mathbf{1}^T$$

Teleportation

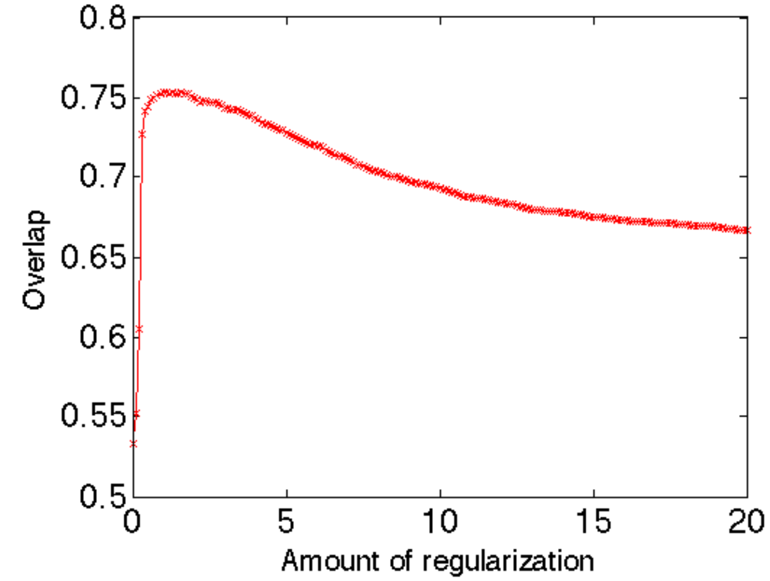
$$\hat{A} = D^{-1/2} A D^{-1/2} + z\mathbf{1}\mathbf{1}^T$$

- In practice proper regularization solves the dangling-sub-graph problem, as teleportation in the Google matrix $G = 0.85 * D^{-1}A + 0.15 * \mathbf{1}\mathbf{1}^T$

Teleportation

$$\hat{A} = D^{-1/2}AD^{-1/2} + z\mathbf{1}\mathbf{1}^T$$

- In practice proper regularization solves the dangling-sub-graph problem, as teleportation in the Google matrix $G = 0.85 * D^{-1}A + 0.15 * \mathbf{1}\mathbf{1}^T$

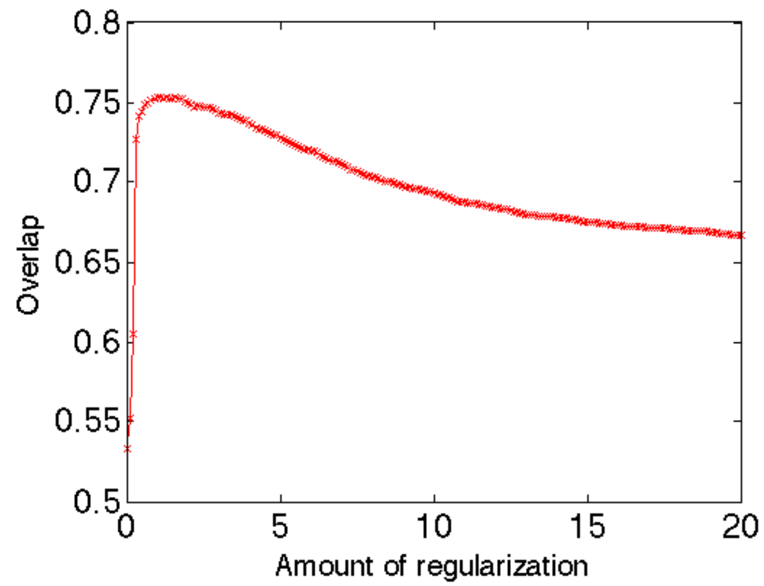


SBM network, c=3 n=10000, epsilon=0.2

Teleportation

$$\hat{A} = D^{-1/2} A D^{-1/2} + z\mathbf{1}\mathbf{1}^T$$

- In practice proper regularization solves the dangling-sub-graph problem, as teleportation in the Google matrix $G = 0.85 * D^{-1}A + 0.15 * \mathbf{1}\mathbf{1}^T$



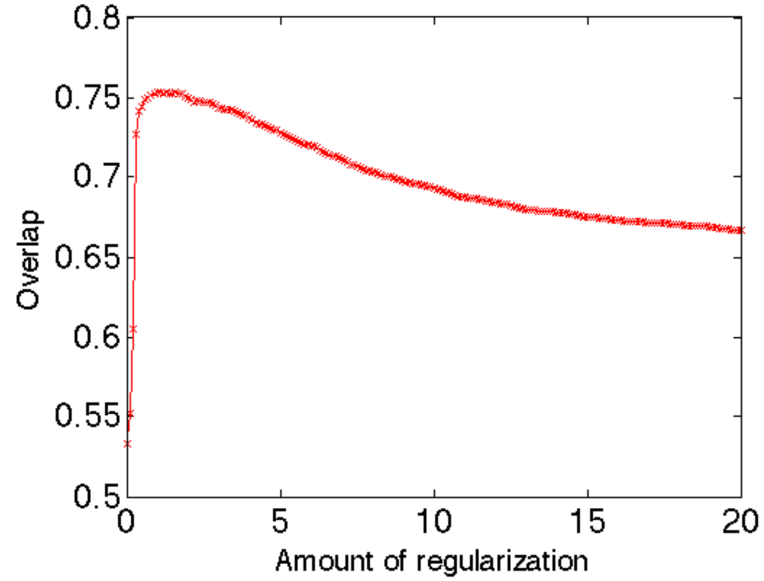
SBM network, c=3 n=10000, epsilon=0.2

- It is hard to optimize the regularization parameter, and it does not help with noise.

Teleportation

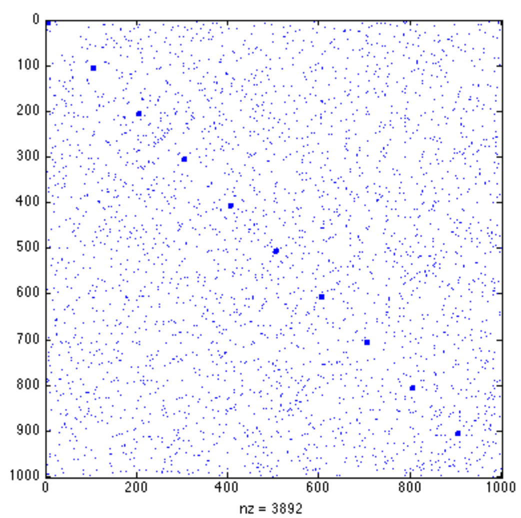
$$\hat{A} = D^{-1/2} A D^{-1/2} + z\mathbf{1}\mathbf{1}^T$$

- In practice proper regularization solves the dangling-sub-graph problem, as teleportation in the Google matrix $G = 0.85 * D^{-1}A + 0.15 * \mathbf{1}\mathbf{1}^T$



SBM network, c=3 n=10000, epsilon=0.2

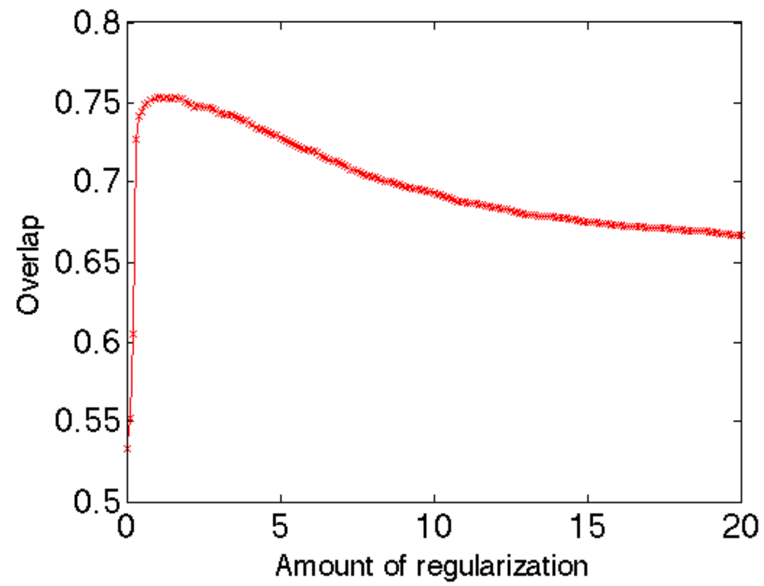
- It is hard to optimize the regularization parameter, and it does not help with noise.



Teleportation

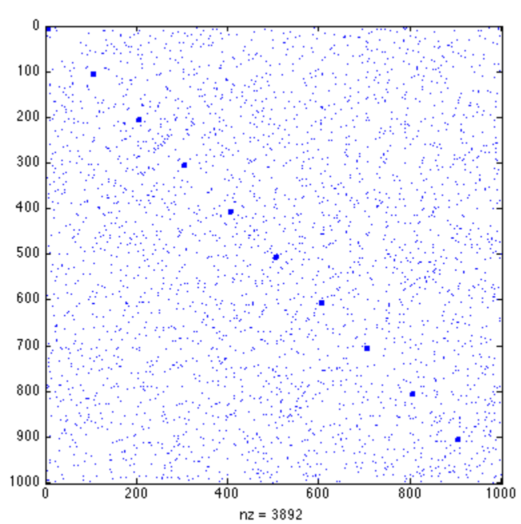
$$\hat{A} = D^{-1/2} A D^{-1/2} + z\mathbf{1}\mathbf{1}^T$$

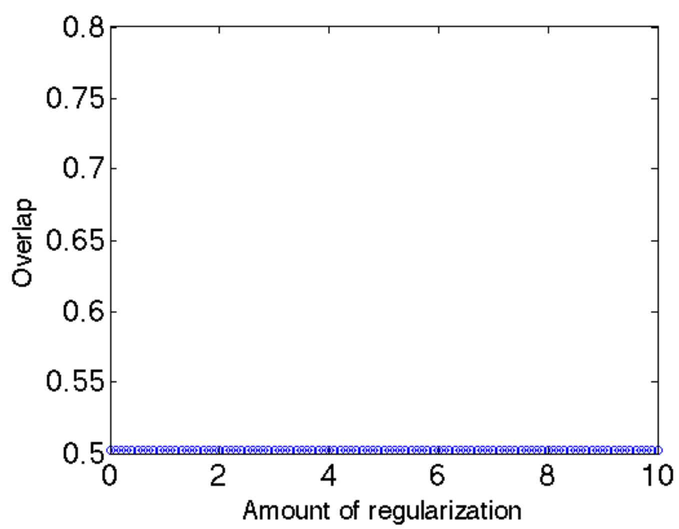
- In practice proper regularization solves the dangling-sub-graph problem, as teleportation in the Google matrix $G = 0.85 * D^{-1}A + 0.15 * \mathbf{1}\mathbf{1}^T$



SBM network, c=3 n=10000, epsilon=0.2

- It is hard to optimize the regularization parameter, and it does not help with noise.





SBM network, c=3 n=10000, epsilon=0.2 with 10 size-10 cliques

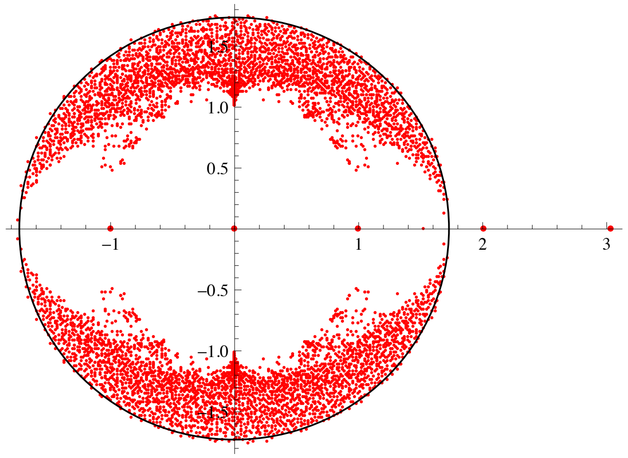
Non-backtracking matrix and Bethe-Hessian

Non-backtracking matrix and Bethe-Hessian

- They work all the way down to the detectability transition in sparse synthetic matrices.

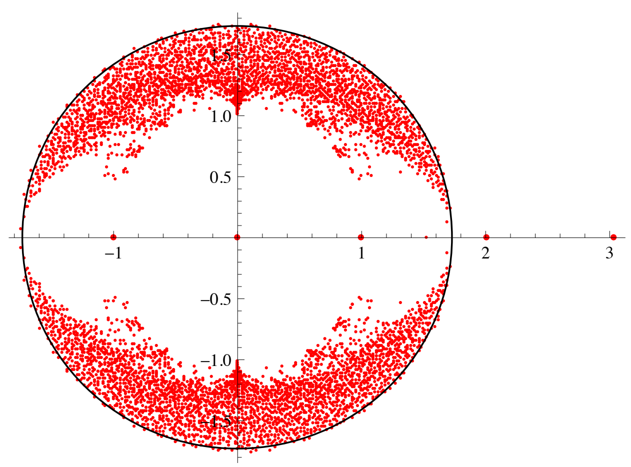
Non-backtracking matrix and Bethe-Hessian

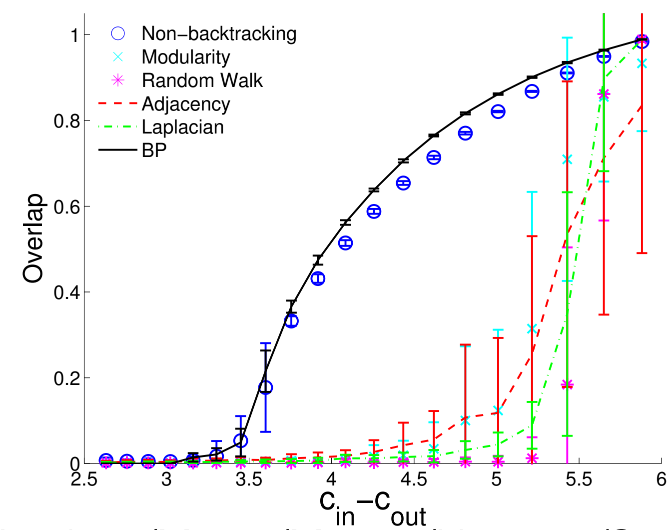
- They work all the way down to the detectability transition in sparse synthetic matrices.



Non-backtracking matrix and Bethe-Hessian

- They work all the way down to the detectability transition in sparse synthetic matrices.

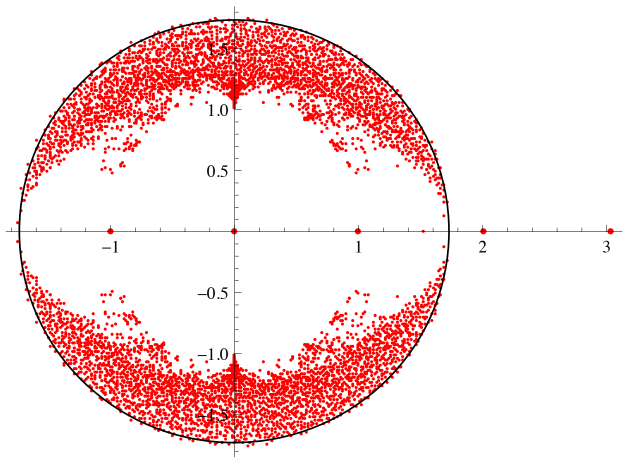


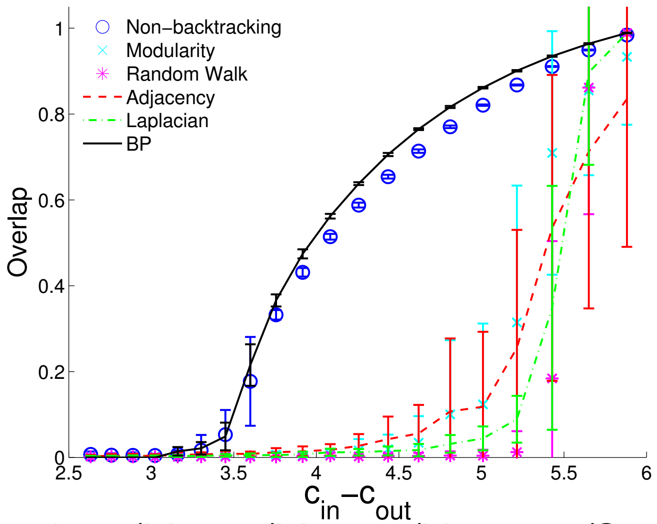


- [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']

Non-backtracking matrix and Bethe-Hessian

- They work all the way down to the detectability transition in sparse synthetic matrices.



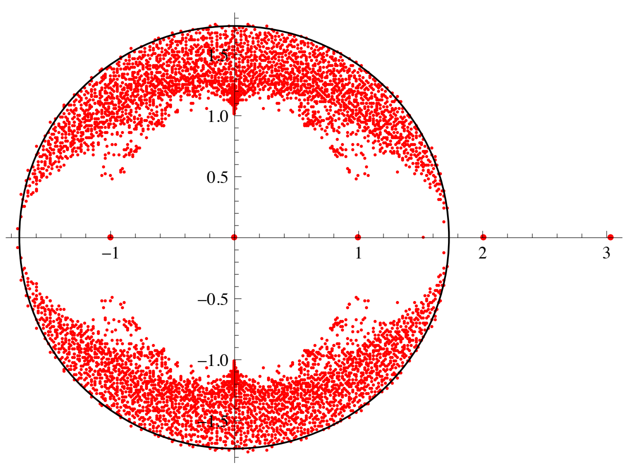


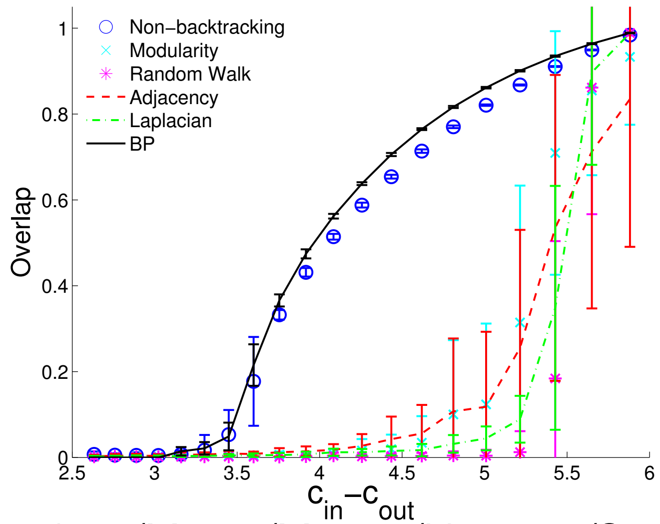
- [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']

- However, they are suboptimal when the graph has short-loops, or when system has noise.

Non-backtracking matrix and Bethe-Hessian

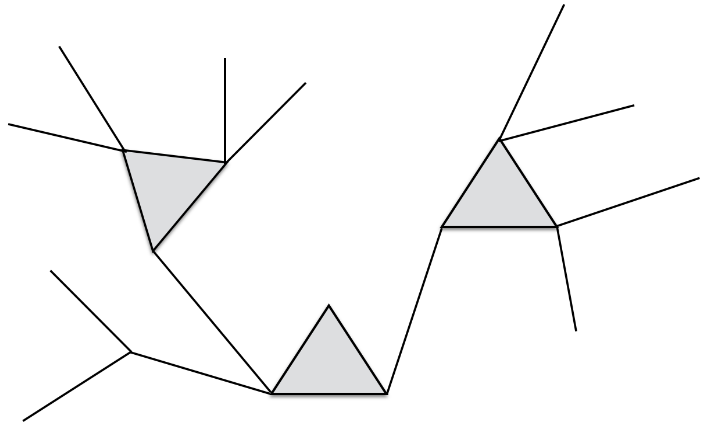
- They work all the way down to the detectability transition in sparse synthetic matrices.





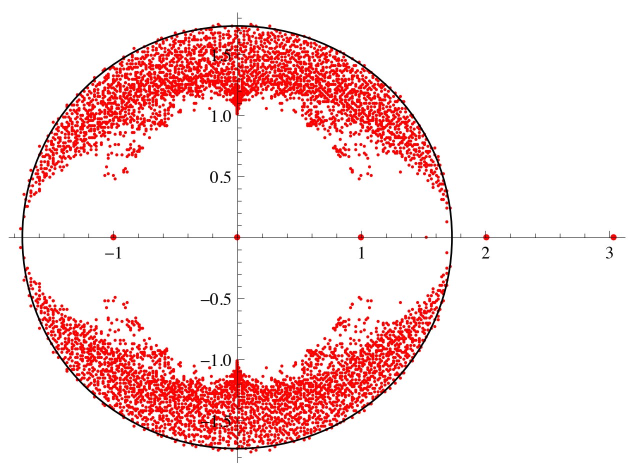
- [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']

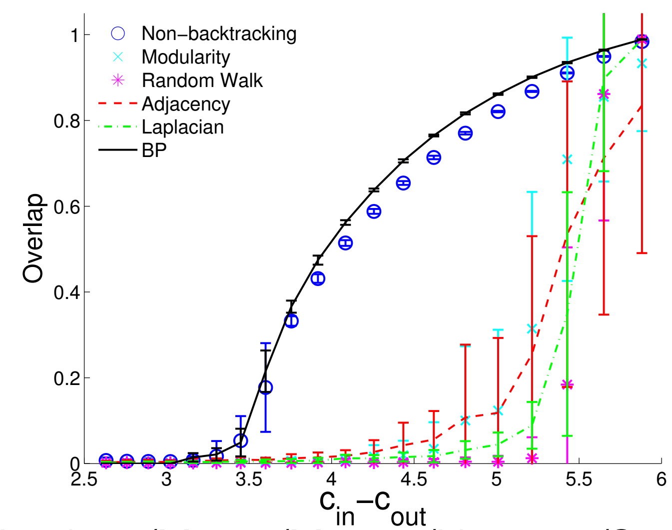
- However, they are suboptimal when the graph has short-loops, or when system has noise.



Non-backtracking matrix and Bethe-Hessian

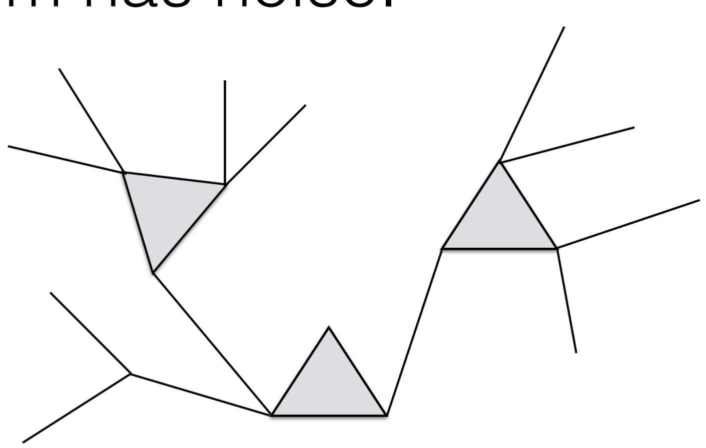
- They work all the way down to the detectability transition in sparse synthetic matrices.





- [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']

- However, they are suboptimal when the graph has short-loops, or when system has noise.



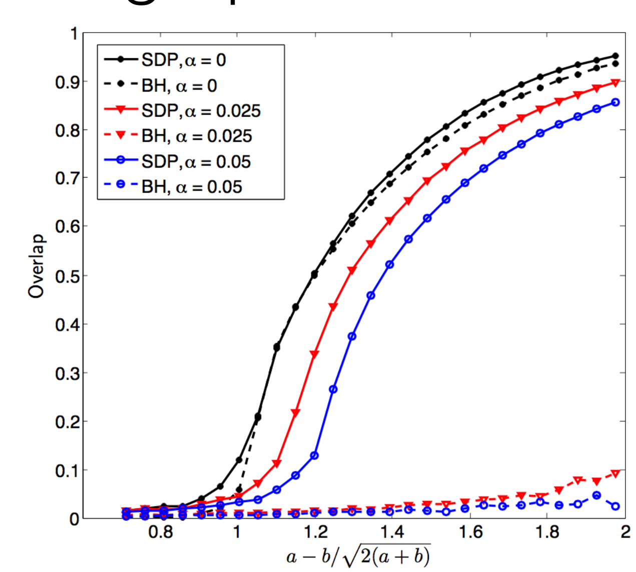


Figure taken from Javanmard/Montanari/Ricci-Tersenghi, PNAS16'

Non-backtracking matrix and regularization

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j})$$

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j})$$

$$Bv = \lambda v$$

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j})$$

$$Bv = \lambda v$$

$$v_{\mathrm{i}n} = \sum_{j\in i} v_{j\to i}$$
$$v_{\mathrm{o}ut} = \sum_{j\in i} v_{i\to j}$$

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j})$$

$$Bv = \lambda v$$

$$v_{\rm in} = \sum_{j\in i} v_{j\to i}$$
$$v_{out} = \sum_{j\in i} v_{i\to j}$$

$$\lambda \begin{pmatrix} (v_{\rm in}) \\ (v_{\rm out}) \end{pmatrix} = \begin{pmatrix} A & -I \\ D-I & 0 \end{pmatrix} \begin{pmatrix} (v_{\rm in}) \\ (v_{\rm out}) \end{pmatrix}$$

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j})$$

$$Bv = \lambda v$$

$$v_{\text{in}} = \sum_{j\in i} v_{j\to i}$$
$$v_{out} = \sum_{j\in i} v_{i\to j}$$

$$\lambda \begin{pmatrix} (v_{\text{in}}) \\ (v_{\text{out}}) \end{pmatrix} = \begin{pmatrix} A & -I \\ D-I & 0 \end{pmatrix} \begin{pmatrix} (v_{\text{in}}) \\ (v_{\text{out}}) \end{pmatrix}$$

$$\det\left[\lambda A - \lambda^2 I - (D-I)\right] = 0$$

Ihara's relation in theory of graph zeta functions

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j}) \qquad Bv = \lambda v$$

$$v_{in} = \textstyle\sum_{j\in i} v_{j\to i}$$
$$v_{out} = \textstyle\sum_{j\in i} v_{i\to j}$$

$$\lambda \begin{pmatrix} (v_{\rm in}) \\ (v_{\rm out}) \end{pmatrix} = \begin{pmatrix} A & -I \\ D-I & 0 \end{pmatrix} \begin{pmatrix} (v_{\rm in}) \\ (v_{\rm out}) \end{pmatrix}$$

$$\det\left[\lambda A - \lambda^2 I - (D-I)\right] = 0$$

Ihara's relation in theory of graph zeta functions

$$\lambda v_{\rm in} = \left[A - \lambda^{-1}(D-I)\right] v_{\rm in}$$

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j})$$

$$Bv = \lambda v$$

$$v_{in} = \sum_{j\in i} v_{j\to i}$$
$$v_{out} = \sum_{j\in i} v_{i\to j}$$

$$\lambda \begin{pmatrix} (v_{\text{in}}) \\ (v_{\text{out}}) \end{pmatrix} = \begin{pmatrix} A & -I \\ D-I & 0 \end{pmatrix} \begin{pmatrix} (v_{\text{in}}) \\ (v_{\text{out}}) \end{pmatrix}$$

$$\det\left[\lambda A - \lambda^2 I - (D-I)\right] = 0$$

Ihara's relation in theory of graph zeta functions

$$\lambda v_{\text{in}} = \left[A - \lambda^{-1}(D-I)\right] v_{\text{in}}$$

$$L_B = A - zD$$

Essentially the Bethe Hessian or z-Laplacian

Non-backtracking matrix and regularization

$$B_{i\to j,k\to l} = \delta_{i,l}(1-\delta_{k,j})$$

$$Bv = \lambda v$$

$$v_{\rm in} = \textstyle\sum_{j\in i} v_{j\to i}$$
$$v_{\rm out} = \textstyle\sum_{j\in i} v_{i\to j}$$

$$\lambda \begin{pmatrix} (v_{\rm in}) \\ (v_{\rm out}) \end{pmatrix} = \begin{pmatrix} A & -I \\ D-I & 0 \end{pmatrix} \begin{pmatrix} (v_{\rm in}) \\ (v_{\rm out}) \end{pmatrix}$$

$$\det\left[\lambda A - \lambda^2 I - (D-I)\right] = 0$$

Ihara's relation in theory of graph zeta functions

$$\lambda v_{\rm in} = \left[A - \lambda^{-1}(D-I)\right] v_{\rm in}$$

$$L_B = A - zD$$

Essentially the Bethe Hessian or z-Laplacian

[Ihara 1966']

[Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']
[Saade/Krzakala/Zdeborová NIPS 14']
[Banks/Moore/Newman/Zhang 14']

Approaches	Point-of-view of regularization	Function of the regularization
Trimming	$L_{\mathrm{Tri}} = A + R_{\mathrm{Tri}}$	Regularization extracts rows and columns of high degree
Teleportation	$L_{\mathrm{r1}} = D^{-1/2}AD^{-1/2} + z\mathbf{1}\mathbf{1}^T$	Regularization punishes low degree localizations
Non-backtracking and Bethe-Hessian	$L_B = A - zD$	Regularization punishes high degree localizations

Why they fail in other (noisy) case?

Approaches	Point-of-view of regularization	Function of the regularization
Trimming	$L_{\mathrm{Tri}} = A + R_{\mathrm{Tri}}$	Regularization extracts rows and columns of high degree
Teleportation	$L_{\mathrm{r1}} = D^{-1/2}AD^{-1/2} + z\mathbf{1}\mathbf{1}^T$	Regularization punishes low degree localizations
Non-backtracking and Bethe-Hessian	$L_B = A - zD$	Regularization punishes high degree localizations

Why they fail in other (noisy) case?

Incorrect guess of source of localization!

Approaches	Point-of-view of regularization	Function of the regularization
Trimming	$L_{\mathrm{Tri}} = A + R_{\mathrm{Tri}}$	Regularization extracts rows and columns of high degree
Teleportation	$L_{\mathrm{r1}} = D^{-1/2}AD^{-1/2} + z\mathbf{1}\mathbf{1}^T$	Regularization punishes low degree localizations
Non-backtracking and Bethe-Hessian	$L_B = A - zD$	Regularization punishes high degree localizations

Why they fail in other (noisy) case?

Incorrect guess of source of localization!

Approaches	Point-of-view of regularization	Function of the regularization
Trimming	$L_{\mathrm{Tri}} = A + R_{\mathrm{Tri}}$	Regularization extracts rows and columns of high degree
Teleportation	$L_{\mathrm{r1}} = D^{-1/2} A D^{-1/2} + z\mathbf{1}\mathbf{1}^T$	Regularization punishes low degree localizations
Non-backtracking and Bethe-Hessian	$L_B = A - zD$	Regularization punishes high degree localizations

- In the sparse matrices, we know that sparsity results to high-degree or low-degree localizations.

Why they fail in other (noisy) case?

Incorrect guess of source of localization!

Approaches	Point-of-view of regularization	Function of the regularization
Trimming	$L_{\mathrm{Tri}} = A + R_{\mathrm{Tri}}$	Regularization extracts rows and columns of high degree
Teleportation	$L_{\mathrm{r1}} = D^{-1/2} A D^{-1/2} + z\mathbf{1}\mathbf{1}^T$	Regularization punishes low degree localizations
Non-backtracking and Bethe-Hessian	$L_B = A - zD$	Regularization punishes high degree localizations

- In the sparse matrices, we know that sparsity results to high-degree or low-degree localizations.
- However in noisy cases, the source of localization is neither high-degree nor low-degrees!

Why they fail in other (noisy) case?
Incorrect guess of source of localization!

Approaches	Point-of-view of regularization	Function of the regularization
Trimming	$L_{\mathrm{Tri}} = A + R_{\mathrm{Tri}}$	Regularization extracts rows and columns of high degree
Teleportation	$L_{\mathrm{r1}} = D^{-1/2}AD^{-1/2} + z\mathbf{1}\mathbf{1}^T$	Regularization punishes low degree localizations
Non-backtracking and Bethe-Hessian	$L_B = A - zD$	Regularization punishes high degree localizations

- In the sparse matrices, we know that sparsity results to high-degree or low-degree localizations.
- However in noisy cases, the source of localization is neither high-degree nor low-degrees!
- What should we do without knowing the source of localization?

My proposal: Learning a regularization

My proposal: Learning a regularization

- Usually we do not know the source for the localization.

My proposal: Learning a regularization

- Usually we do not know the source for the localization.

- So we should not use regularizations that target the “guessed” source of the localization.

My proposal: Learning a regularization

- Usually we do not know the source for the localization.

- So we should not use regularizations that target the "guessed" source of the localization.

- Instead, let's learn a regularization from the existing localizations, i.e. localized eigenvectors.

Algorithm:
the X-Laplacian

Algorithm: the X-Laplacian

$$L_X = A + X$$

Algorithm: the X-Laplacian

$$L_X = A + X$$

Inverse participation ratio of eigenvector u: $I(u) = \sum_{i=1}^{n} u_i^4$

Algorithm:
the X-Laplacian

$$L_X = A + X$$

Inverse participation ratio of eigenvector u: $I(u) = \sum_{i=1}^{n} u_i^4$

Algorithm 1: Regularization Learning

Algorithm: the X-Laplacian

$$L_X = A + X$$

Inverse participation ratio of eigenvector u: $I(u) = \sum_{i=1}^{n} u_i^4$

Algorithm 1: Regularization Learning

Input: Real symmetric matrix A, number of eigenvectors q, learning rate $\eta = O(1)$, threshold Δ.
Output: X-Laplacian, L_X, whose leading eigenvectors reveal the global structures in A.

Algorithm: the X-Laplacian

$$L_X = A + X$$

Inverse participation ratio of eigenvector u: $I(u) = \sum_{i=1}^{n} u_i^4$

Algorithm 1: Regularization Learning

Input: Real symmetric matrix A, number of eigenvectors q, learning rate $\eta = O(1)$, threshold Δ.
Output: X-Laplacian, L_X, whose leading eigenvectors reveal the global structures in A.

1. Set X to be all-zero matrix.

Algorithm: the X-Laplacian

$$L_X = A + X$$

Inverse participation ratio of eigenvector u: $I(u) = \sum_{i=1}^{n} u_i^4$

Algorithm 1: Regularization Learning

Input: Real symmetric matrix A, number of eigenvectors q, learning rate $\eta = O(1)$, threshold Δ.
Output: X-Laplacian, L_X, whose leading eigenvectors reveal the global structures in A.

1. Set X to be all-zero matrix.
2. Find set of eigenvectors $U = \{u_1, u_2, ..., u_q\}$ associated with the first q largest eigenvalues (in algebra) of L_X.

Algorithm: the X-Laplacian

$$L_X = A + X$$

Inverse participation ratio of eigenvector u: $I(u) = \sum_{i=1}^{n} u_i^4$

Algorithm 1: Regularization Learning

Input: Real symmetric matrix A, number of eigenvectors q, learning rate $\eta = O(1)$, threshold Δ.
Output: X-Laplacian, L_X, whose leading eigenvectors reveal the global structures in A.

1. Set X to be all-zero matrix.
2. Find set of eigenvectors $U = \{u_1, u_2, ..., u_q\}$ associated with the first q largest eigenvalues (in algebra) of L_X.
3. Identify the eigenvector v that has the largest inverse participation ratio among the q eigenvectors in U. I.e. find $v = \text{argmax}_{u \in U} I(u)$.

Algorithm: the X-Laplacian

$$L_X = A + X$$

Inverse participation ratio of eigenvector u: $I(u) = \sum_{i=1}^{n} u_i^4$

Algorithm 1: Regularization Learning

Input: Real symmetric matrix A, number of eigenvectors q, learning rate $\eta = O(1)$, threshold Δ.
Output: X-Laplacian, L_X, whose leading eigenvectors reveal the global structures in A.

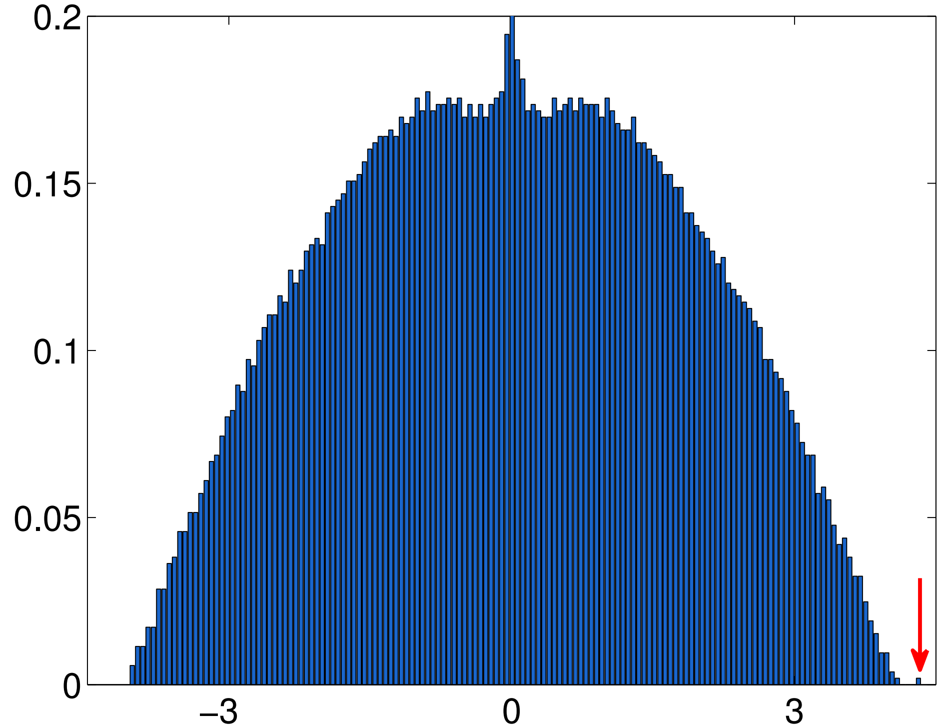
1. Set X to be all-zero matrix.
2. Find set of eigenvectors $U = \{u_1, u_2, ..., u_q\}$ associated with the first q largest eigenvalues (in algebra) of L_X.
3. Identify the eigenvector v that has the largest inverse participation ratio among the q eigenvectors in U. I.e. find $v = \mathrm{argmax}_{u \in U} I(u)$.
4. if $I(v) < \Delta$, return $L_X = A + X$; Otherwise, $\forall i, X_{ii} \leftarrow X_{ii} - \eta v_i^2$, then go to step 2.

Algorithm: Whack-a-mole

Algorithm: Whack-a-mole

Localized eigenvalues are moles

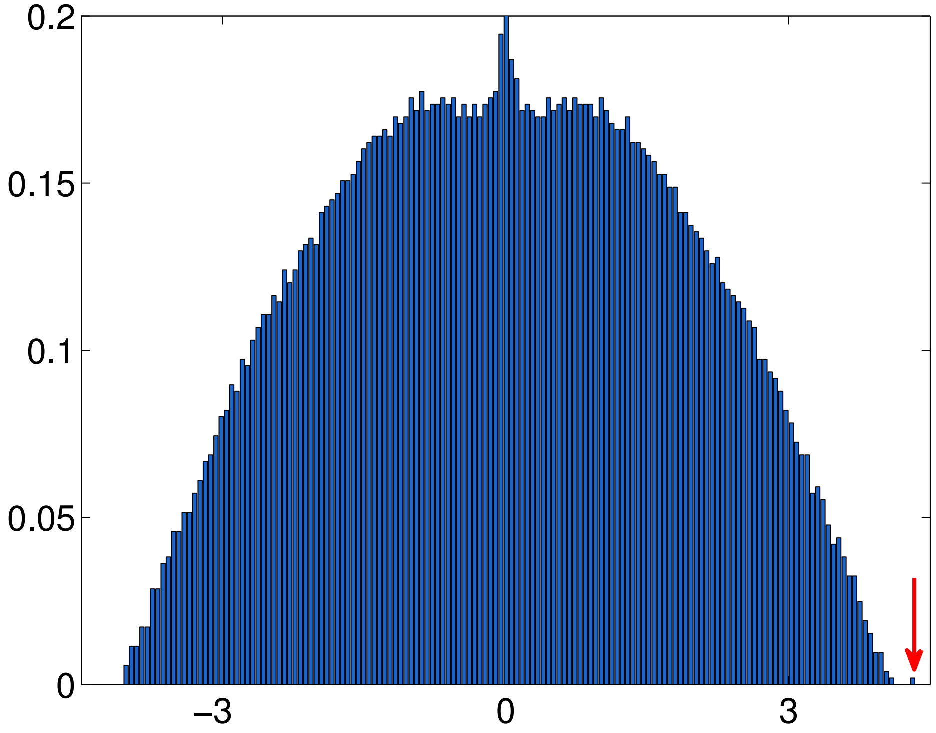
Localized eigenvalues are moles



Spectral density of A
Before learning

A is generated by the Stochastic Block Model with n=10000 nodes, average degree c=3, q=2 groups, and c_out/c_in=0.125.

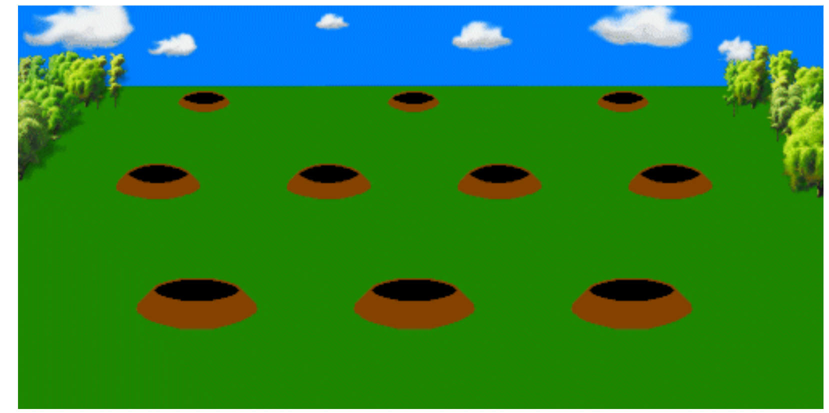
Localized eigenvalues are moles

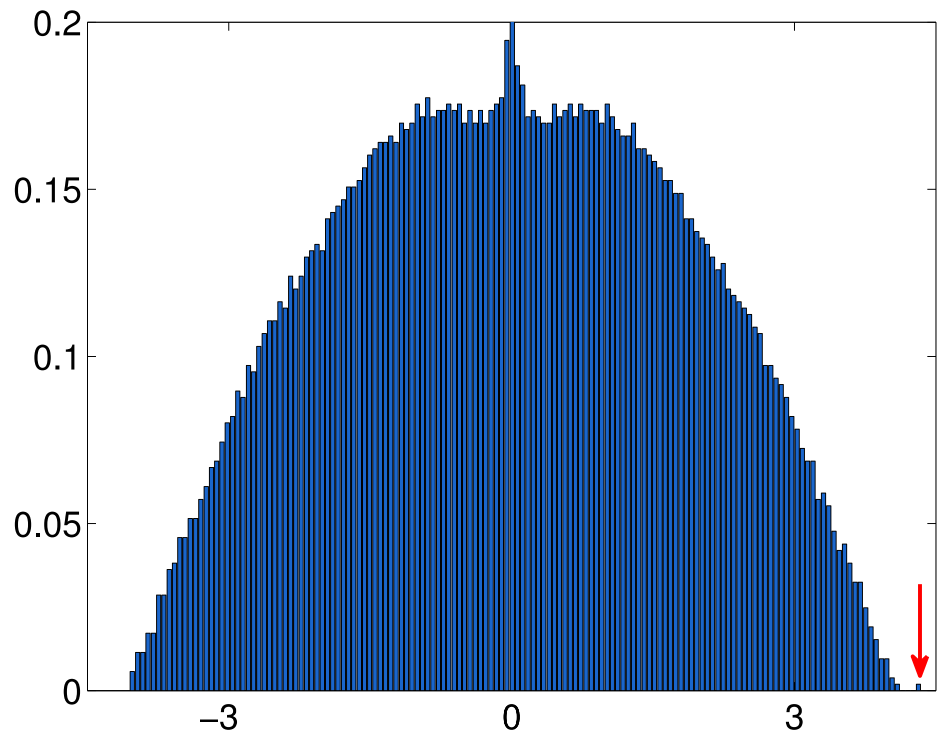


Spectral density of A
Before learning

A is generated by the Stochastic Block Model with n=10000 nodes, average degree c=3, q=2 groups, and c_out/c_in=0.125.

Localized eigenvalues are moles

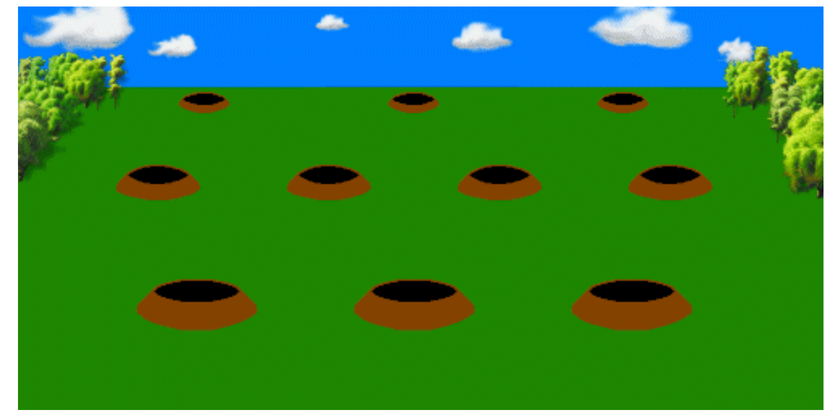


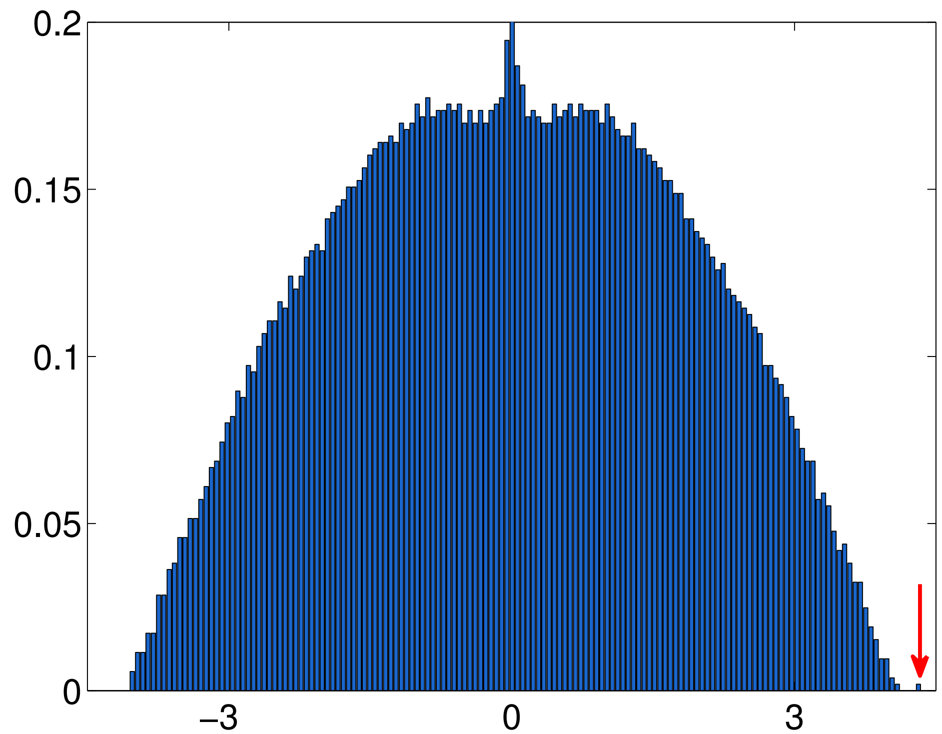


Spectral density of A
Before learning

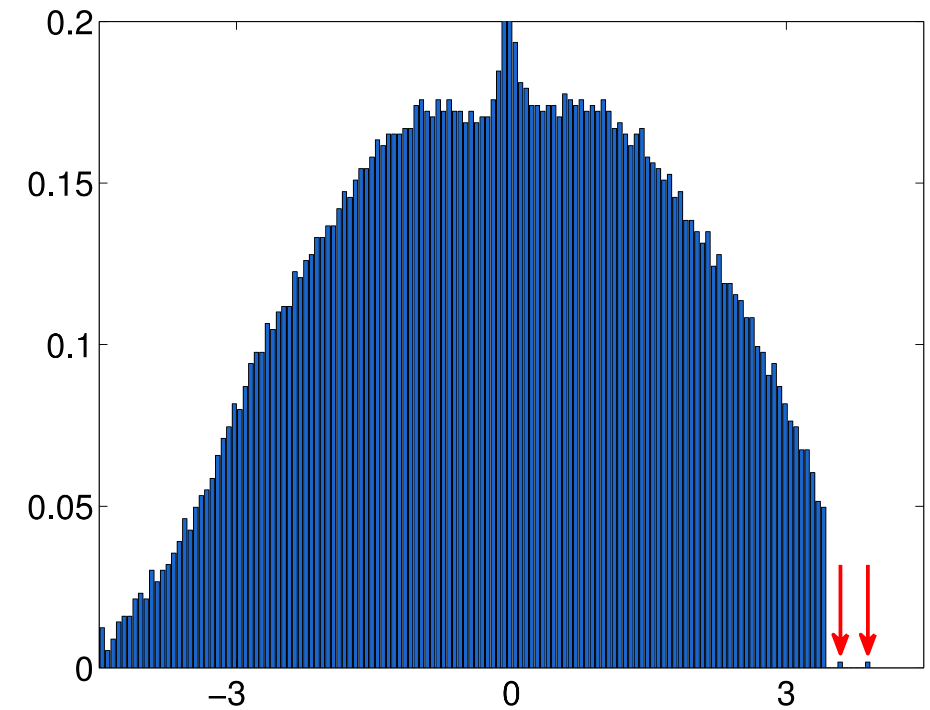
A is generated by the Stochastic Block Model with n=10000 nodes, average degree c=3, q=2 groups, and c_out/c_in=0.125.

Localized eigenvalues are moles





Spectral density of A
Before learning



Spectral density of A+X
After learning

A is generated by the Stochastic Block Model with n=10000 nodes,
average degree c=3, q=2 groups, and c_out/c_in=0.125.

Matrix perturbation analysis

Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$

Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$ $\eta = O(1)$

Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$ $\eta = O(1)$ $v_i^2 = O(1/n)$

Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$ $\qquad \eta = O(1) \quad v_i^2 = O(1/n)$

Matrix perturbation:

$$L_X u_i = \lambda_i u_i$$

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$ $\eta = O(1)$ $v_i^2 = O(1/n)$

Matrix perturbation:

$$L_X u_i = \lambda_i u_i$$

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvalue: $\hat{\lambda}_i = u_i^T \hat{L} u_i$

$$\hat{L} = -\eta v_i^2$$

Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$ $\eta = O(1)$ $v_i^2 = O(1/n)$

Matrix perturbation:

$$L_X u_i = \lambda_i u_i$$

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvalue: $\hat{\lambda}_i = u_i^T \hat{L} u_i$

$$\hat{L} = -\eta v_i^2$$

Change of the selected eigenvalue: $\hat{\lambda}_v = -\eta \sum_{i=1}^{n} v_i^4 = -\eta I(v)$

Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$ $\quad \eta = O(1) \quad v_i^2 = O(1/n)$

Matrix perturbation:

$$L_X u_i = \lambda_i u_i$$

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvalue: $\hat{\lambda}_i = u_i^T \hat{L} u_i$

$$\hat{L} = -\eta v_i^2$$

Change of the selected eigenvalue: $\hat{\lambda}_v = -\eta \sum_{i=1}^{n} v_i^4 = -\eta I(v)$

After perturbation, the eigenvalue of selected eigenvector v decreases by amount proportional to its Inverse Participate Ratio

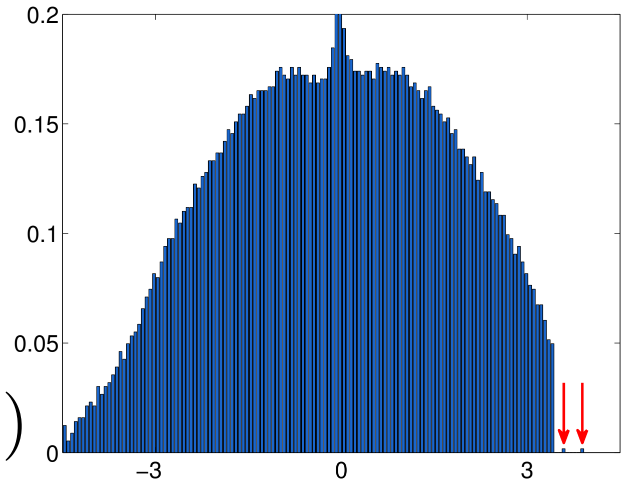
Matrix perturbation analysis

Learning: $X_{ii} \leftarrow X_{ii} - \eta v_i^2$ $\quad \eta = O(1) \quad v_i^2 = O(1/n)$

Matrix perturbation:

$$L_X u_i = \lambda_i u_i$$

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$



Change of an eigenvalue: $\hat{\lambda}_i = u_i^T \hat{L} u_i$

$$\hat{L} = -\eta v_i^2$$

Change of the selected eigenvalue: $\hat{\lambda}_v = -\eta \sum_{i=1}^{n} v_i^4 = -\eta I(v)$

After perturbation, the eigenvalue of selected eigenvector v decreases by amount proportional to its Inverse Participate Ratio

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvector: $$\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j$$

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvector: $$\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j \qquad \hat{L} = -\eta v_i^2$$

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvector:

$$\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j \qquad \hat{L} = -\eta v_i^2$$

$$= -\eta \sum_{j \neq i} \frac{\sum_k u_{jk} v_k^2 u_{ik}}{\lambda_i - \lambda_j} u_j$$

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvector:

$$\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j \qquad \hat{L} = -\eta v_i^2$$

$$= -\eta \sum_{j \neq i} \frac{\sum_k u_{jk} v_k^2 u_{ik}}{\lambda_i - \lambda_j} u_j$$

Change of the IPR of an eigenvector:

$$I(u_i + \hat{u}_i) - I(u_i) \approx -4\eta \sum_{l=1}^{n} \sum_{j \neq i} \frac{u_{jl}^2 v_l^2 u_{il}^4}{\lambda_i - \lambda_j}$$

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvector:

$$\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j \qquad \hat{L} = -\eta v_i^2$$

$$= -\eta \sum_{j \neq i} \frac{\sum_k u_{jk} v_k^2 u_{ik}}{\lambda_i - \lambda_j} u_j$$

Change of the IPR of an eigenvector: $I(u_i + \hat{u}_i) - I(u_i) \approx -4\eta \sum_{l=1}^{n} \sum_{j \neq i} \frac{u_{jl}^2 v_l^2 u_{il}^4}{\lambda_i - \lambda_j}$

After perturbation, the leading eigenvectors are delocalizing.

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

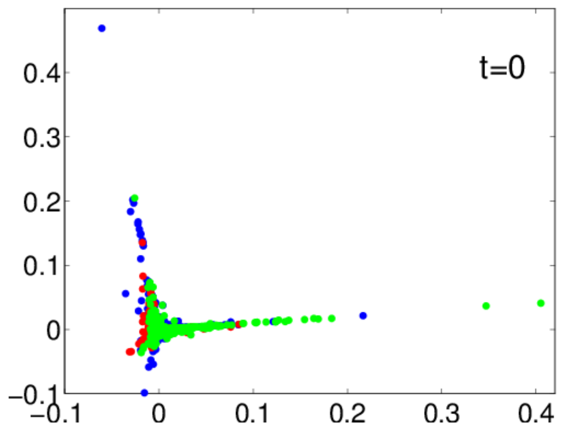
Change of an eigenvector:

$$\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j \qquad \hat{L} = -\eta v_i^2$$

$$= -\eta \sum_{j \neq i} \frac{\sum_k u_{jk} v_k^2 u_{ik}}{\lambda_i - \lambda_j} u_j$$

Change of the IPR of an eigenvector: $I(u_i + \hat{u}_i) - I(u_i) \approx -4\eta \sum_{l=1}^{n} \sum_{j \neq i} \frac{u_{jl}^2 v_l^2 u_{il}^4}{\lambda_i - \lambda_j}$

After perturbation, the leading eigenvectors are delocalizing.



The second eigenvector vs. the third eigenvector, during learning
Network is generated by SBM with n=42000 nodes, average degree c=3, q=3 groups, c_out/c_in=0.08.

Matrix perturbation analysis

$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

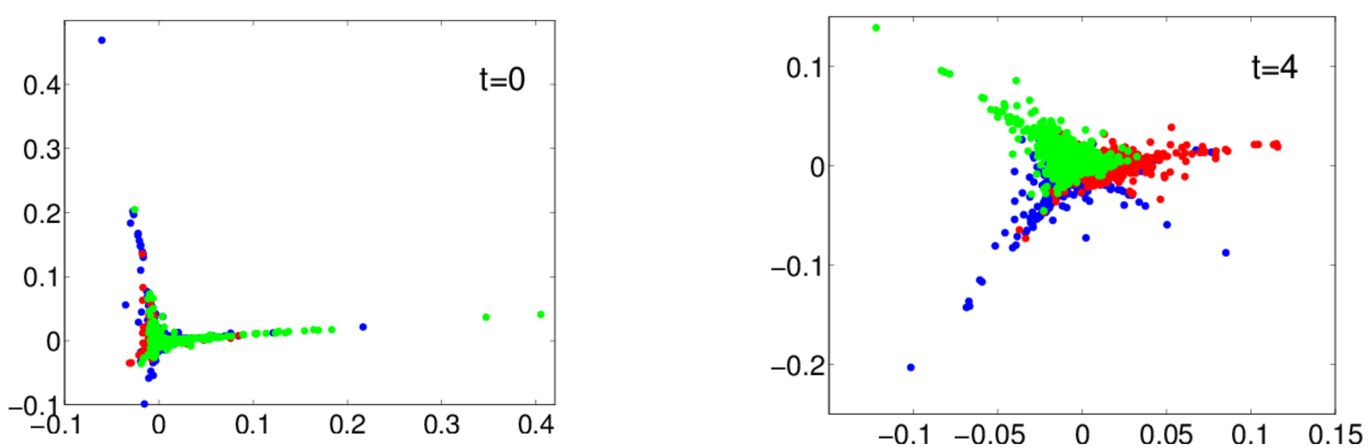
Change of an eigenvector:

$$\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j \qquad \hat{L} = -\eta v_i^2$$

$$= -\eta \sum_{j \neq i} \frac{\sum_k u_{jk} v_k^2 u_{ik}}{\lambda_i - \lambda_j} u_j$$

Change of the IPR of an eigenvector: $I(u_i + \hat{u}_i) - I(u_i) \approx -4\eta \sum_{l=1}^{n} \sum_{j \neq i} \frac{u_{jl}^2 v_l^2 u_{il}^4}{\lambda_i - \lambda_j}$

After perturbation, the leading eigenvectors are delocalizing.



The second eigenvector vs. the third eigenvector, during learning
Network is generated by SBM with n=42000 nodes, average degree c=3, q=3 groups, c_out/c_in=0.08.

Matrix perturbation analysis

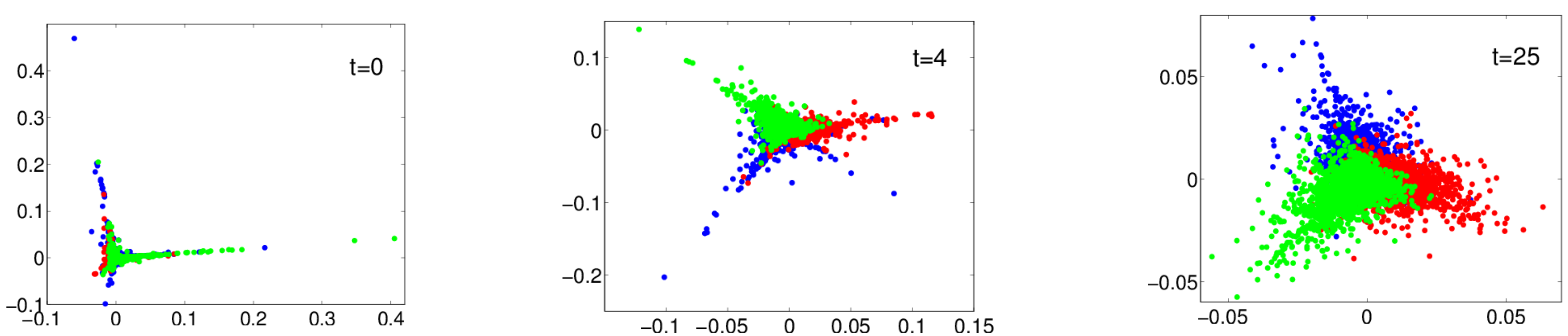
$$(L_X + \hat{L})(u_i + \hat{u}_i) = (\lambda_i + \hat{\lambda}_i)(u_i + \hat{u}_i)$$

Change of an eigenvector: $\hat{u}_i = \sum_{j \neq i} \frac{u_j^T \hat{L} u_i}{\lambda_i - \lambda_j} u_j \qquad \hat{L} = -\eta v_i^2$

$$= -\eta \sum_{j \neq i} \frac{\sum_k u_{jk} v_k^2 u_{ik}}{\lambda_i - \lambda_j} u_j$$

Change of the IPR of an eigenvector: $I(u_i + \hat{u}_i) - I(u_i) \approx -4\eta \sum_{l=1}^{n} \sum_{j \neq i} \frac{u_{jl}^2 v_l^2 u_{il}^4}{\lambda_i - \lambda_j}$

After perturbation, the leading eigenvectors are delocalizing.

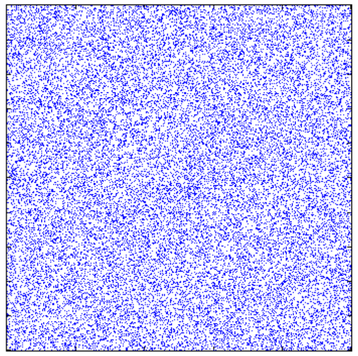


The second eigenvector vs. the third eigenvector, during learning
Network is generated by SBM with n=42000 nodes, average degree c=3, q=3 groups, c_out/c_in=0.08.

Experimental evaluations

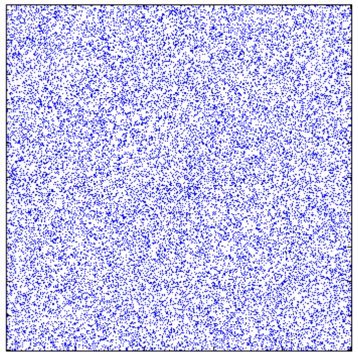
Experimental evaluations

- Community detection in sparse graphs with noise.
  [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']
  [Saade/Krzakala/Zdeborová NIPS 14']
  [Javanmard/Montanari/Ricci-Tersenghi PNAS16']

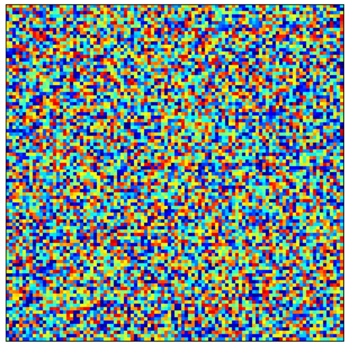


Experimental evaluations

- Community detection in sparse graphs with noise.
  [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']
  [Saade/Krzakala/Zdeborová NIPS 14']
  [Javanmard/Montanari/Ricci-Tersenghi PNAS16']

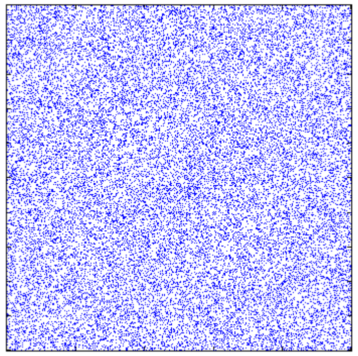


- Clustering from sparse pairwise similarities.
  [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']

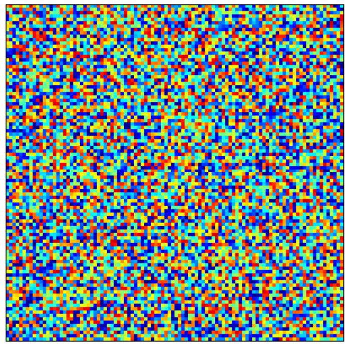


Experimental evaluations

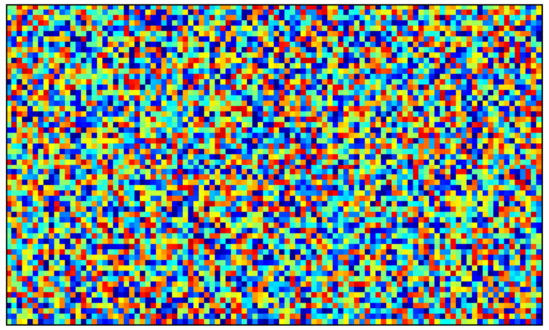
- Community detection in sparse graphs with noise.
  [Krzakala/Moore/Mossel/Neeman/Sly/Zdeborová/Zhang PNAS 13']
  [Saade/Krzakala/Zdeborová NIPS 14']
  [Javanmard/Montanari/Ricci-Tersenghi PNAS16']



- Clustering from sparse pairwise similarities.
  [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']

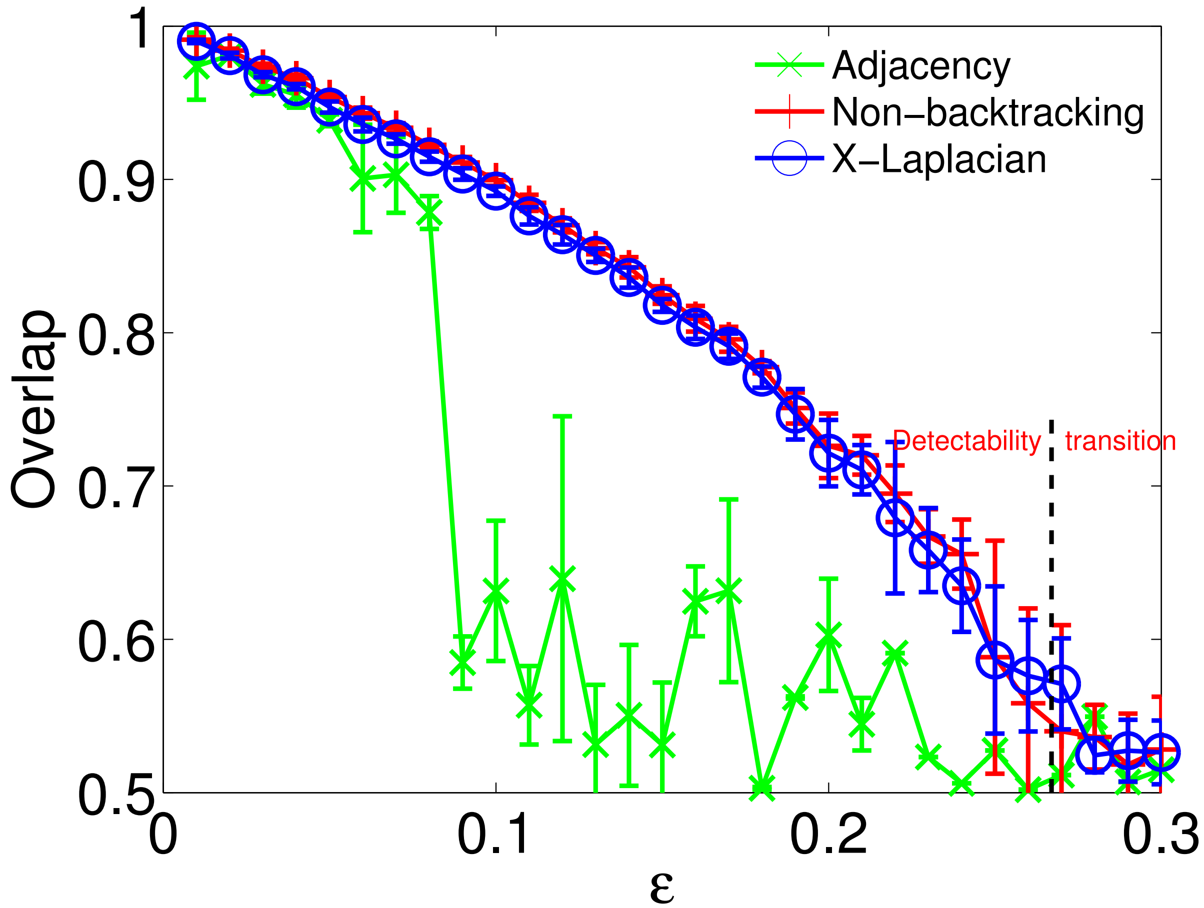


- Rank estimation and matrix completion.
  [Keshavan/Oh/Montanari 09']
  [Saade/Krzakala/Zdeborová NIPS 15']



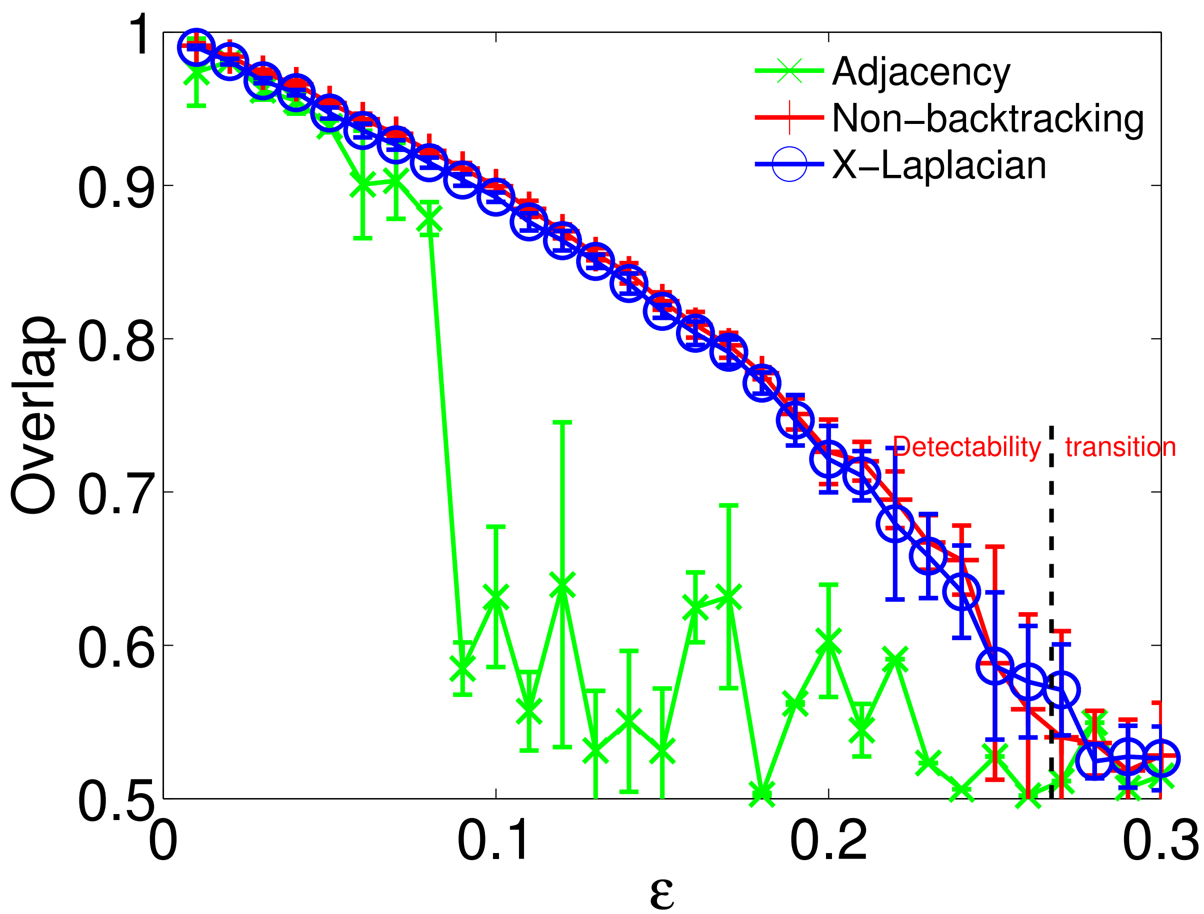
Community detection

Community detection

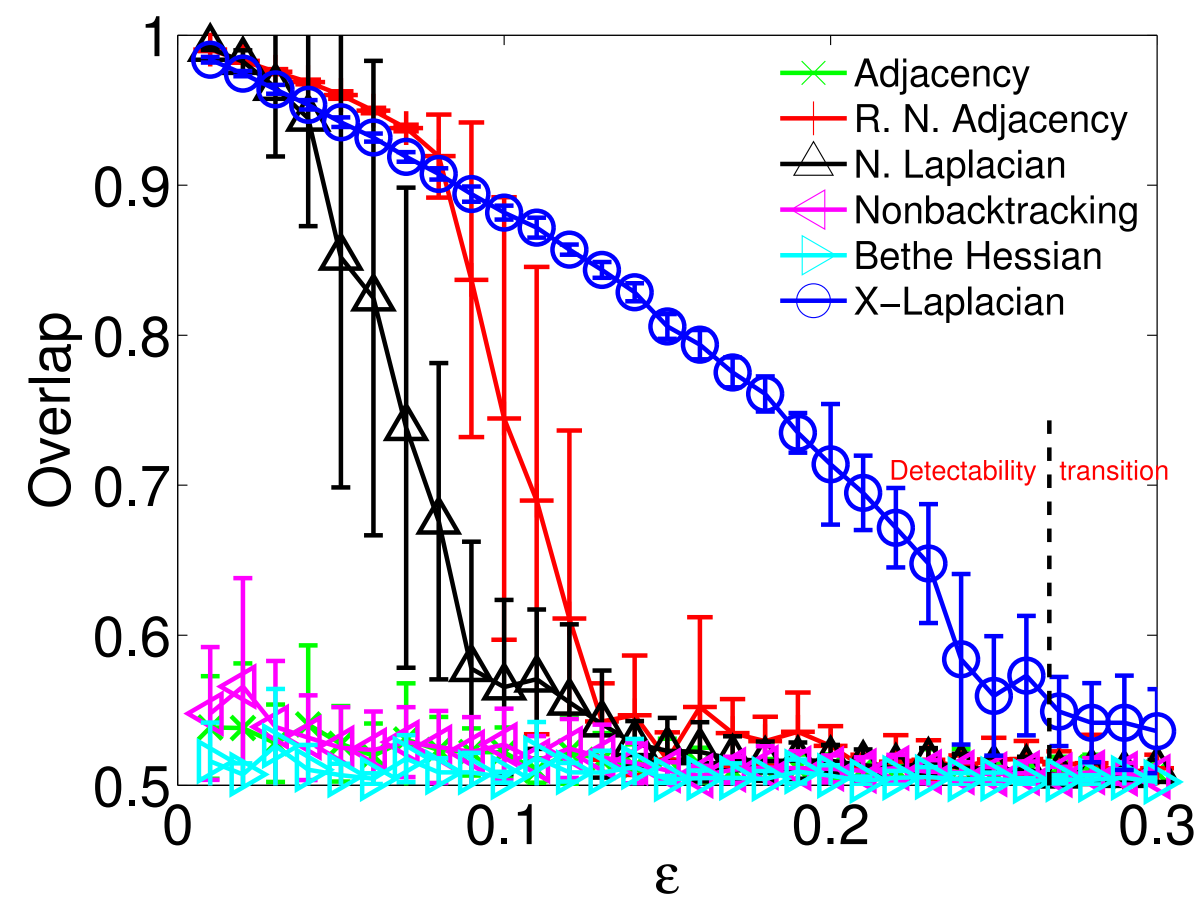


Stochastic Block Model
n=10000, q=2, c=3

Community detection



Stochastic Block Model
n=10000, q=2, c=3



Noisy Stochastic Block Model
n=10000, q=2, c=3

Community detection

Community detection

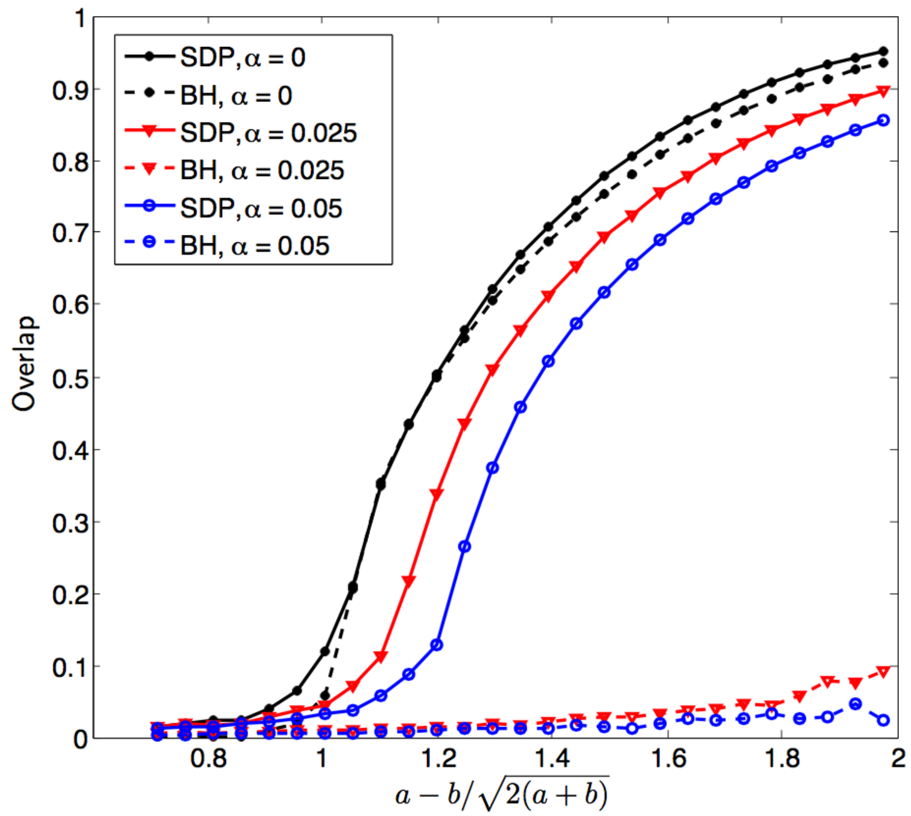


Figure taken from

[Javanmard/Montanari/Ricci-Tersenghi PNAS 16']

Community detection

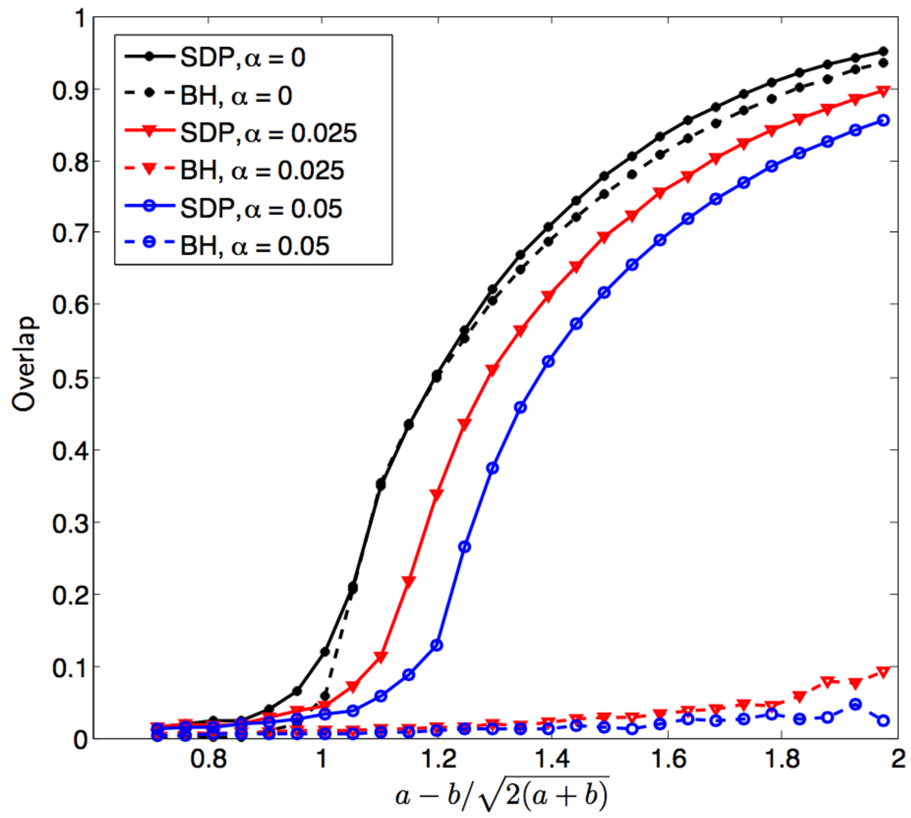
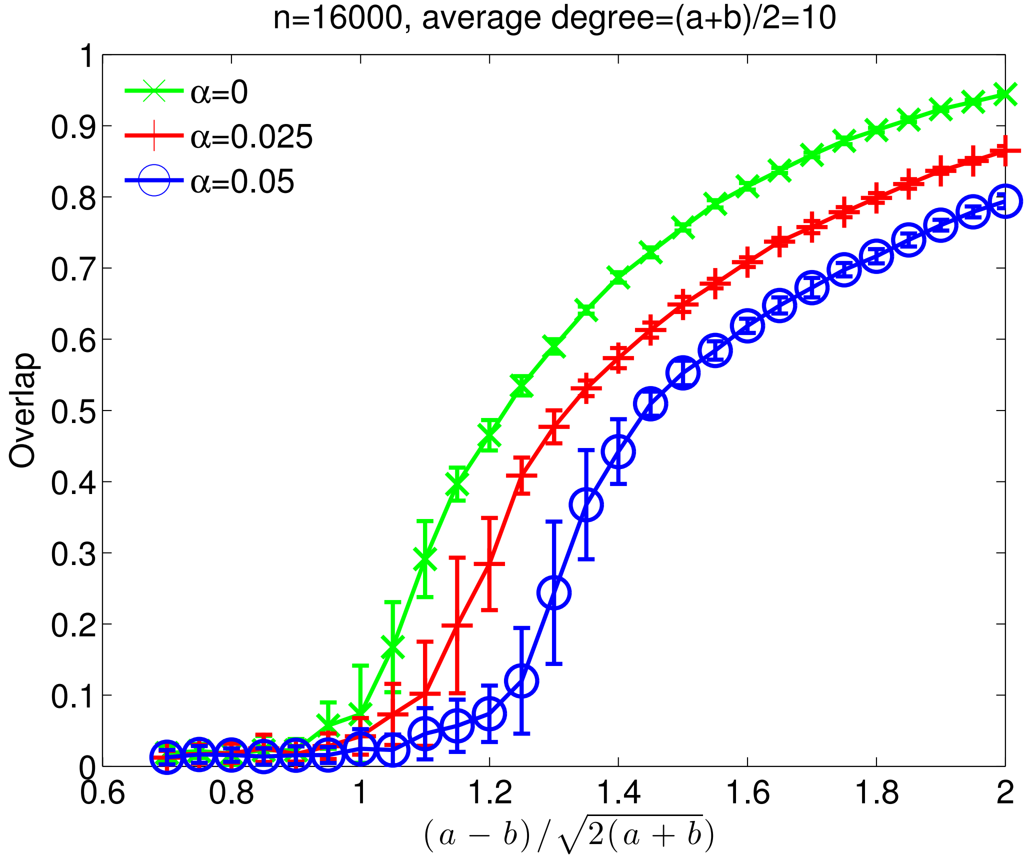


Figure taken from

[Javanmard/Montanari/Ricci-Tersenghi PNAS 16']



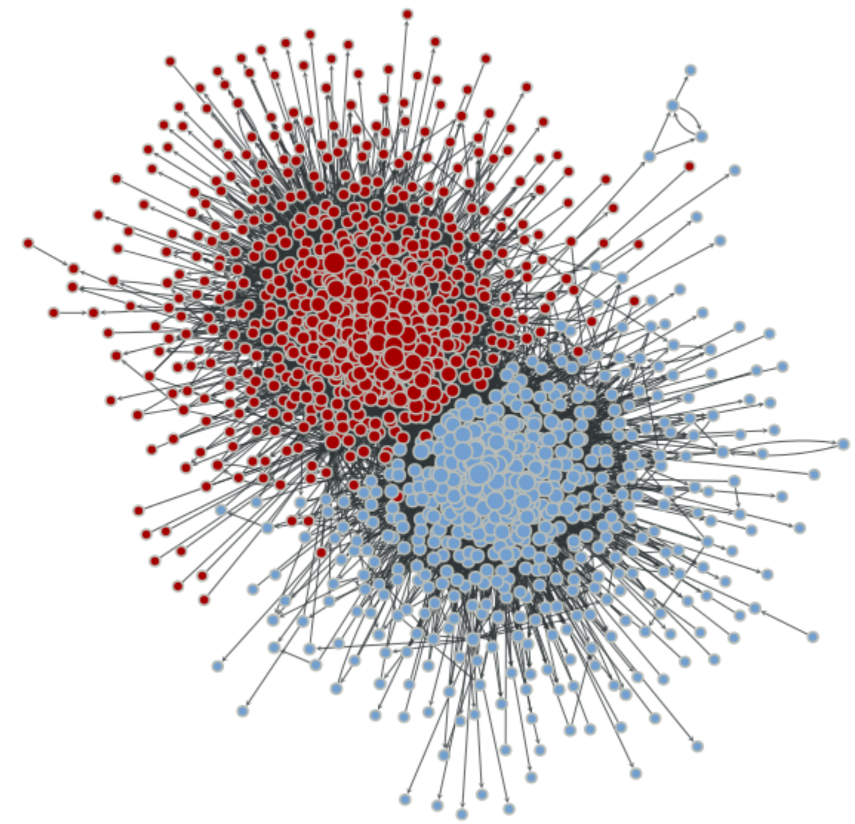
X-Laplacian

Community detection

Community detection

Network of political blogs

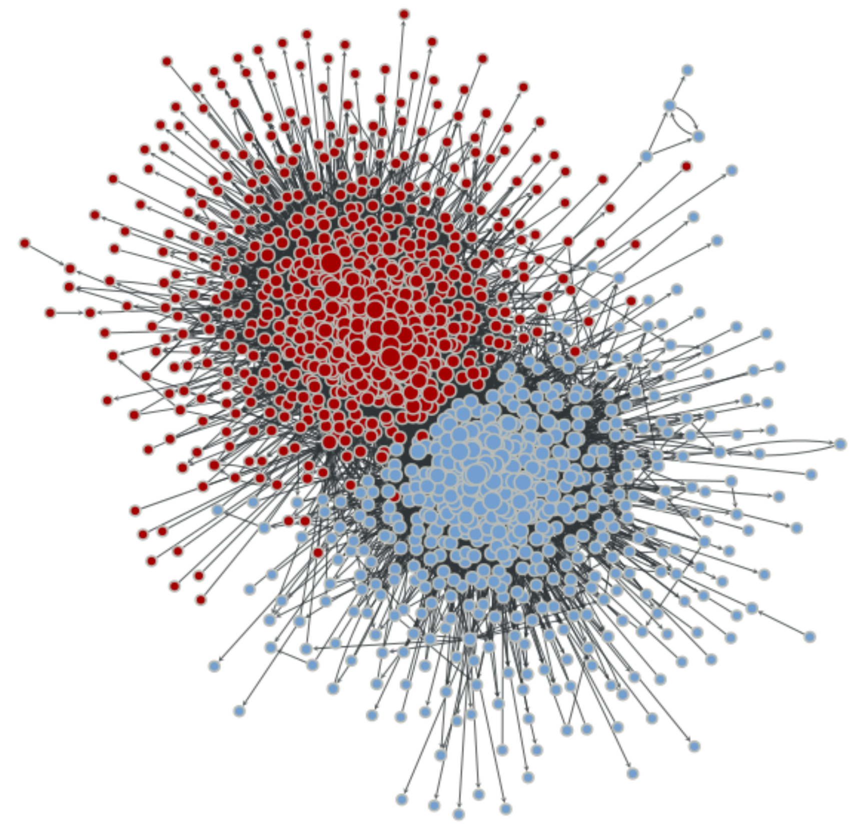
[Adamic/Glance 05']
1222 nodes, 16714 edges



Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges

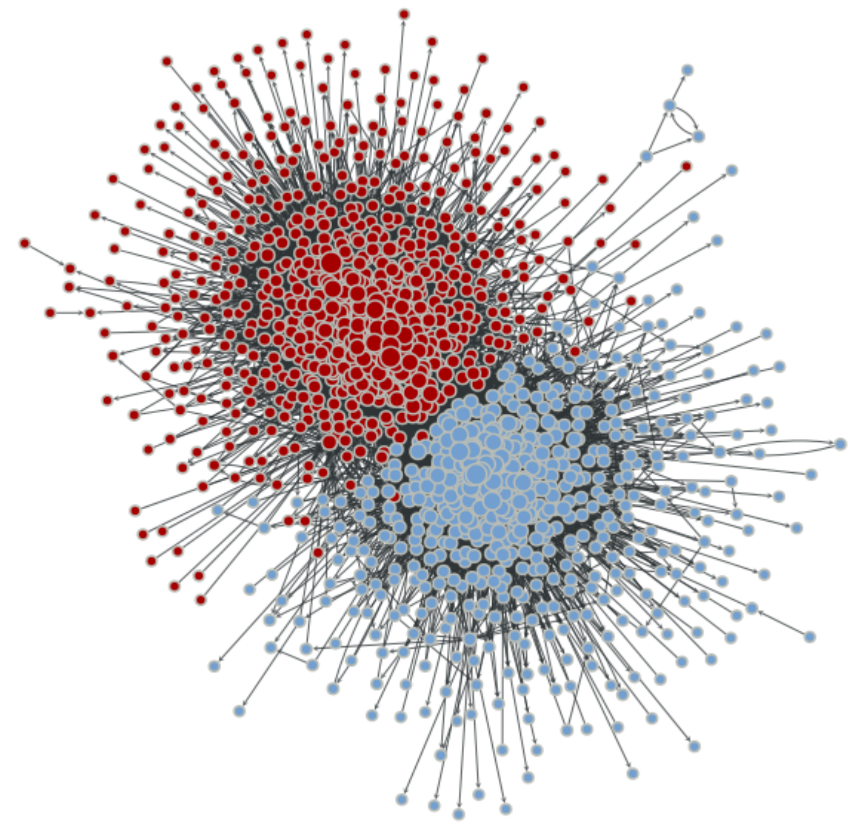


Method	
# Misclassified nodes	

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges

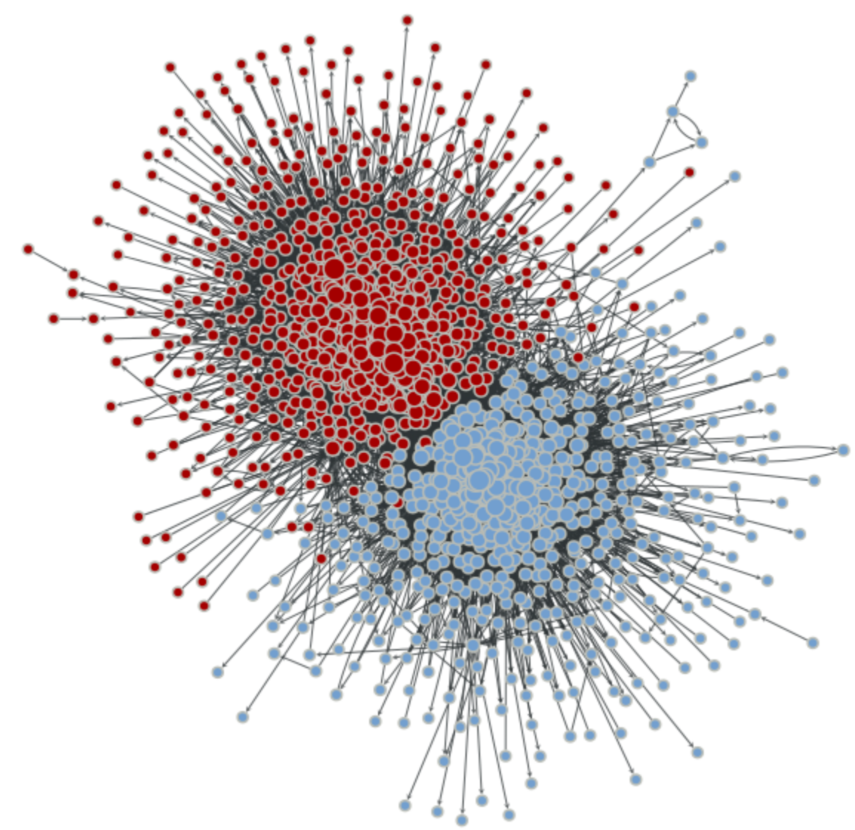


Method	Adjacency matrix
# Misclassified nodes	83

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges

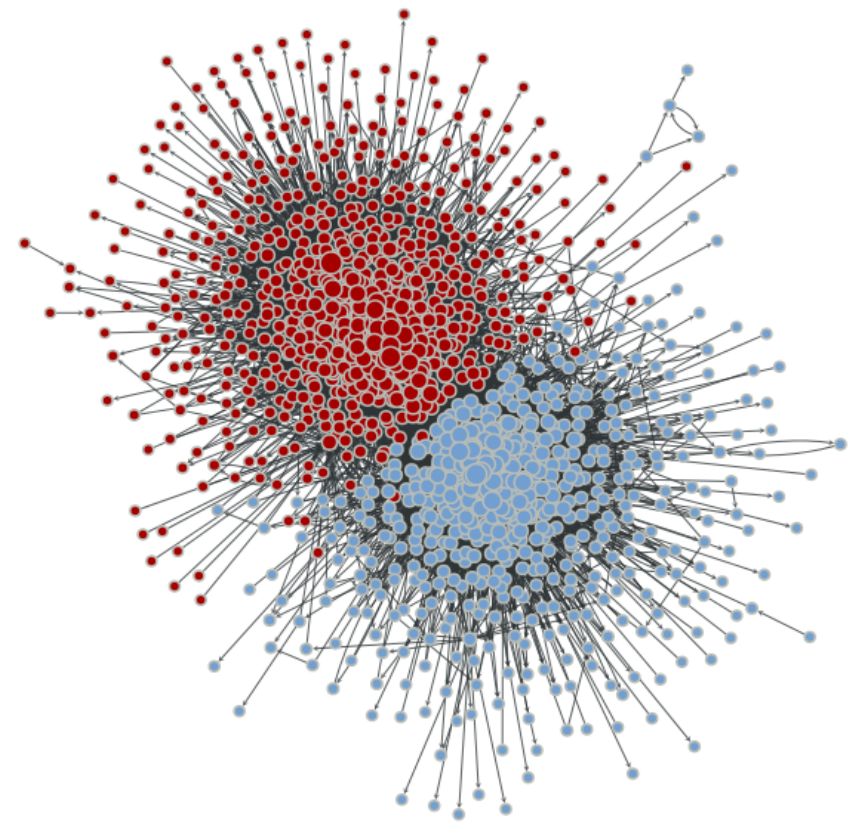


Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']
# Misclassified nodes	83	80

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges

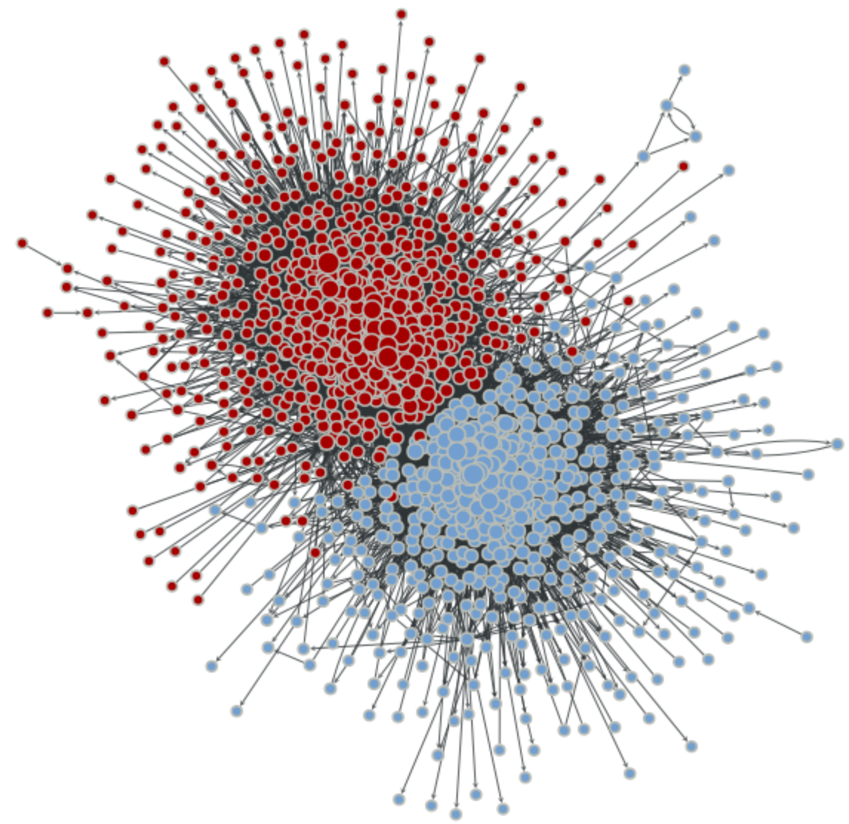


Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']
# Misclassified nodes	83	80	63

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges

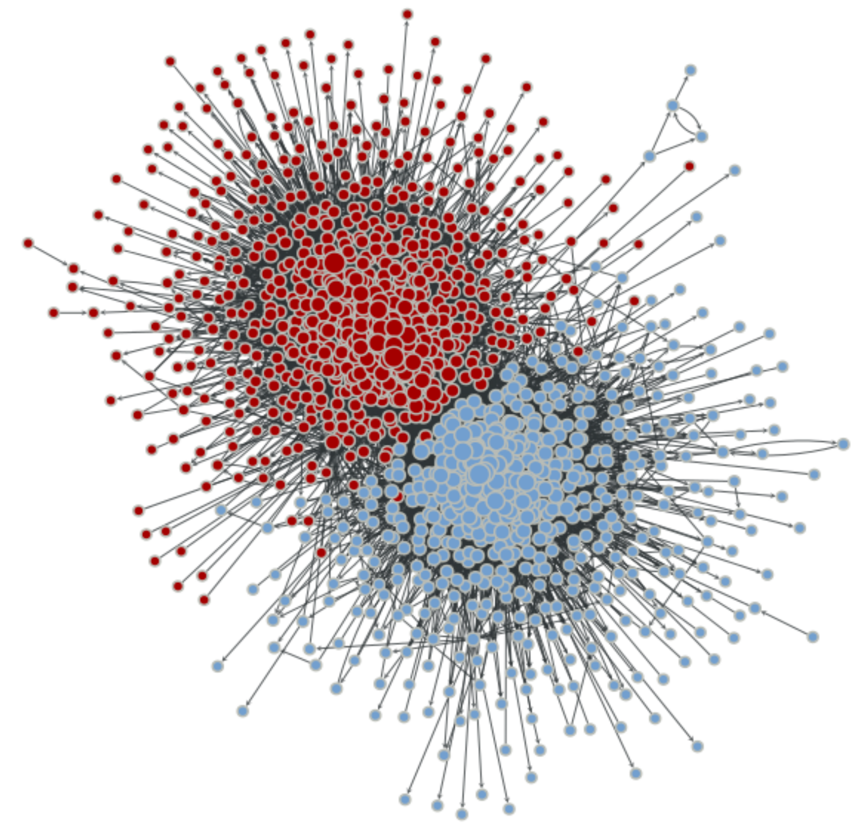


Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']	Semidefinite Progarmming [Cai/Li 13']
# Misclassified nodes	83	80	63	63

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges

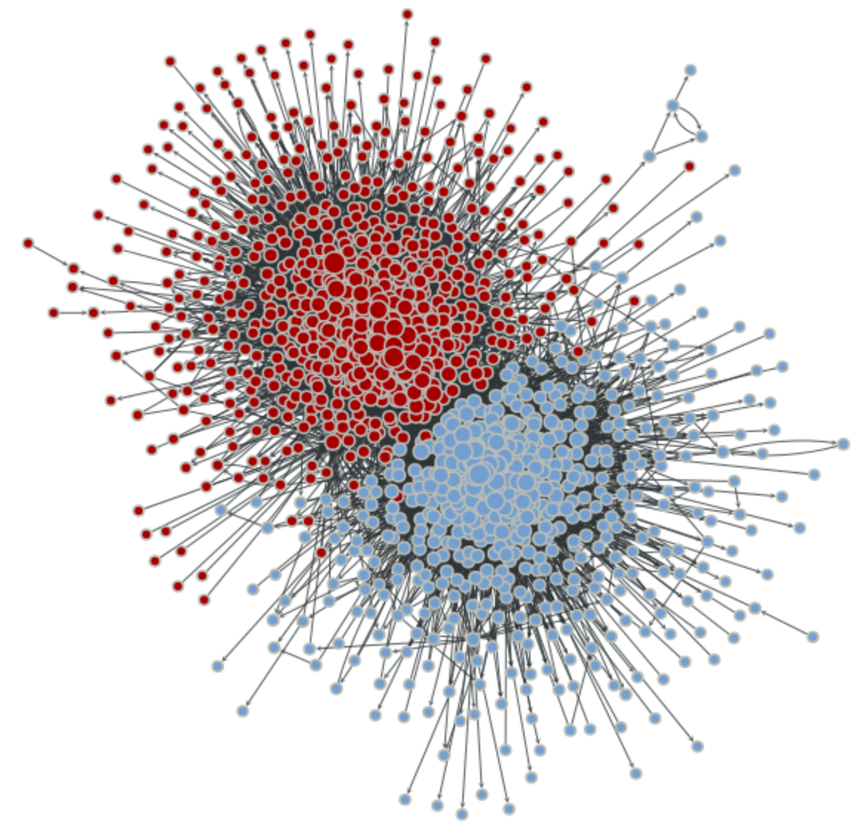


Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']	Semidefinite Progarmmin g [Cai/Li 13']	DCSBM BP
# Misclassified nodes	83	80	63	63	61

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges

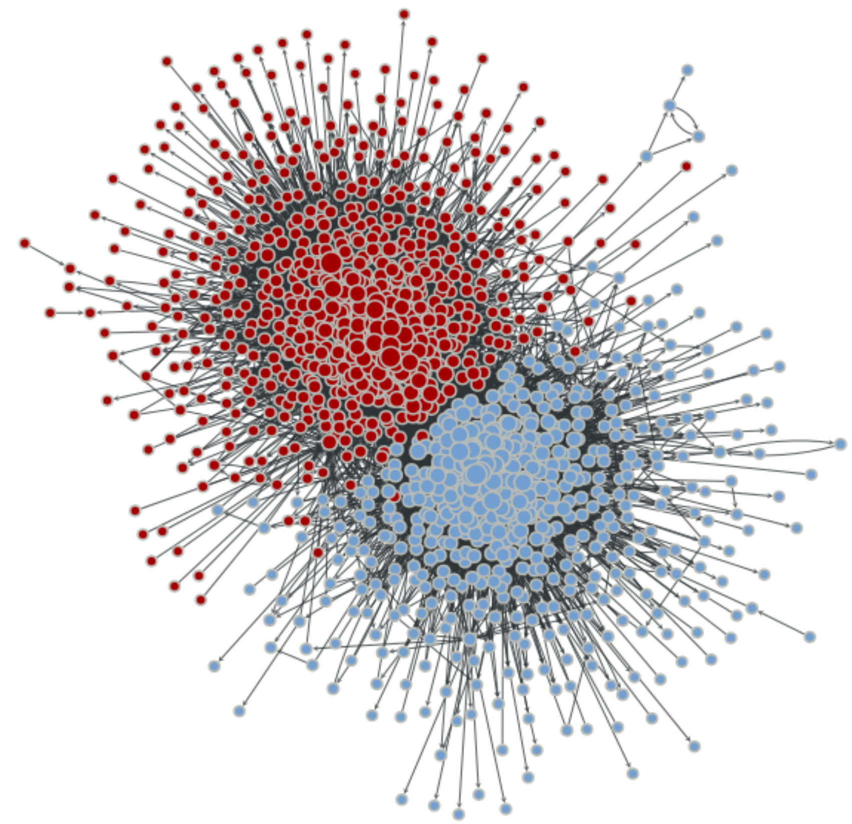


Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']	Semidefinite Progarmmin g [Cai/Li 13']	DCSBM BP	SCORE [Jin 15']
# Misclassified nodes	83	80	63	63	61	58

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges



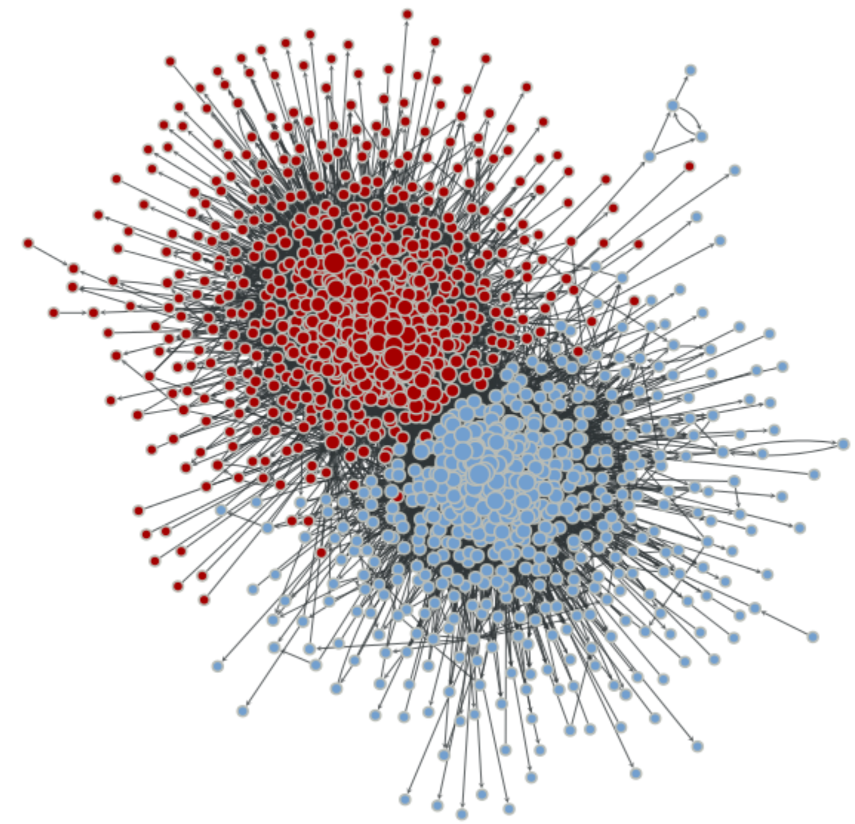
Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']	Semidefinite Progarmmin g [Cai/Li 13']	DCSBM BP	SCORE [Jin 15']
# Misclassified nodes	83	80	63	63	61	58

"Achieving Optimal Misclassification Proportion …"
[Gao et al. 15']

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges



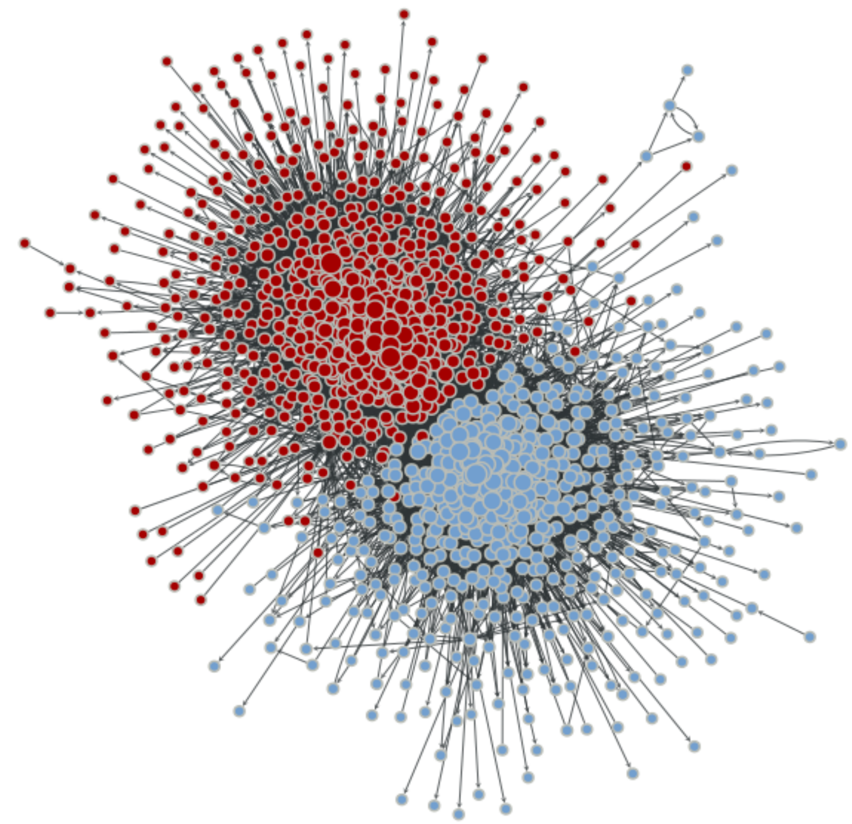
Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']	Semidefinite Progarmmin g [Cai/Li 13']	DCSBM BP	SCORE [Jin 15']	[Gao et al 15']
# Misclassified nodes	83	80	63	63	61	58	58

"Achieving Optimal Misclassification Proportion …"
[Gao et al. 15']

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges



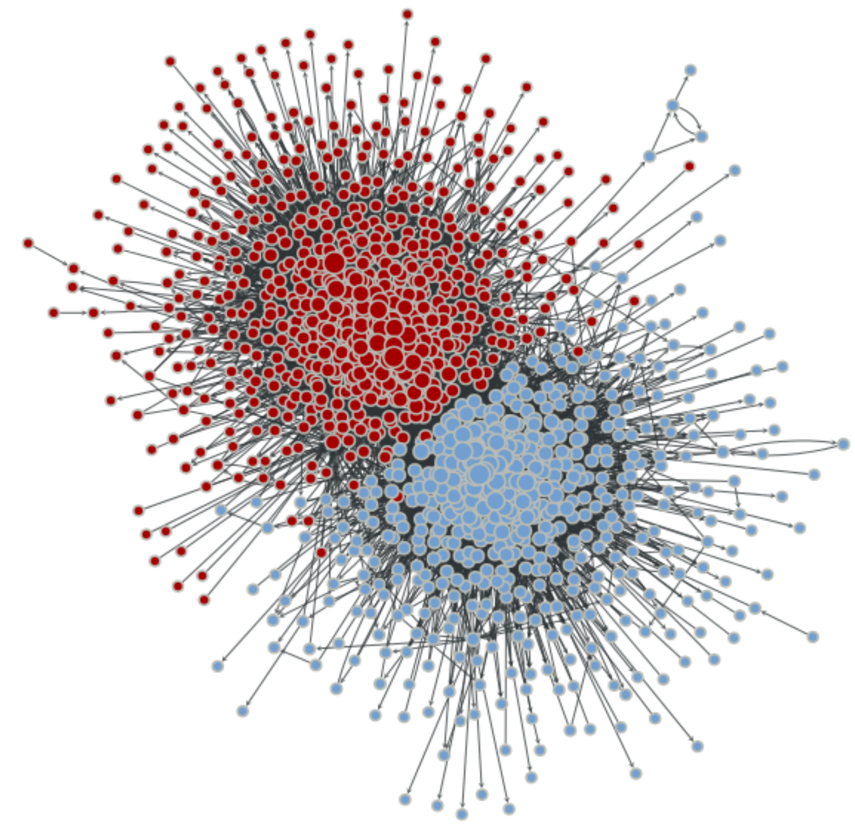
Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']	Semidefinite Progarmmin g [Cai/Li 13']	DCSBM BP	SCORE [Jin 15']	[Gao et al 15']	X-Laplacian
# Misclassified nodes	83	80	63	63	61	58	58	50

"Achieving Optimal Misclassification Proportion …"
[Gao et al. 15']

Community detection

Network of political blogs

[Adamic/Glance 05']
1222 nodes, 16714 edges



Method	Adjacency matrix	Reg.Spectral Clustering [Qin/Rohe 13']	Modularity BP [Zhang/Moore 14']	Semidefinite Progarmmin g [Cai/Li 13']	DCSBM BP	SCORE [Jin 15']	[Gao et al 15']	X-Laplacian
# Misclassified nodes	83	80	63	63	61	58	58	50

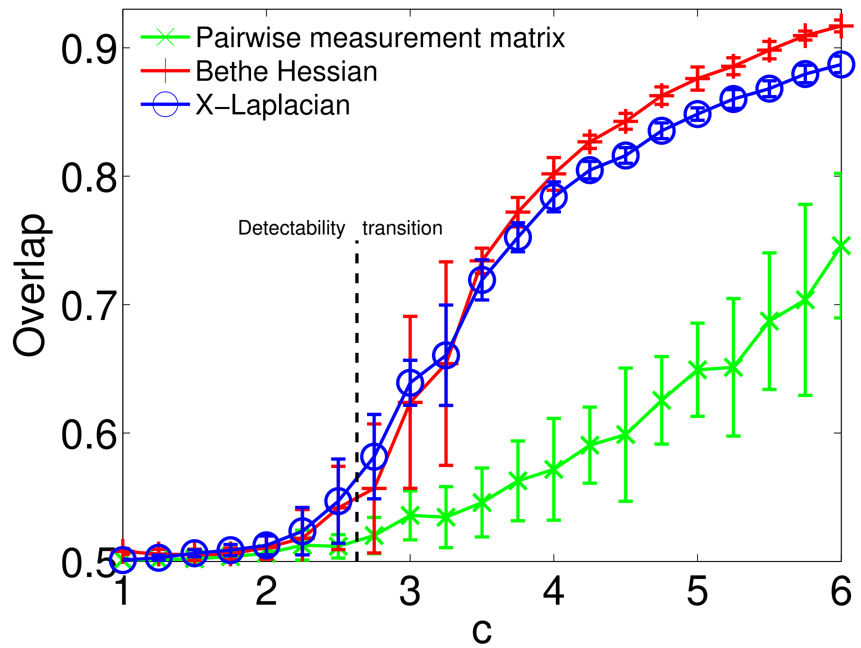
"Achieving Optimal Misclassification Proportion …"
[Gao et al. 15']

Clustering from sparse similarities

Model from [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']

Clustering from sparse similarities

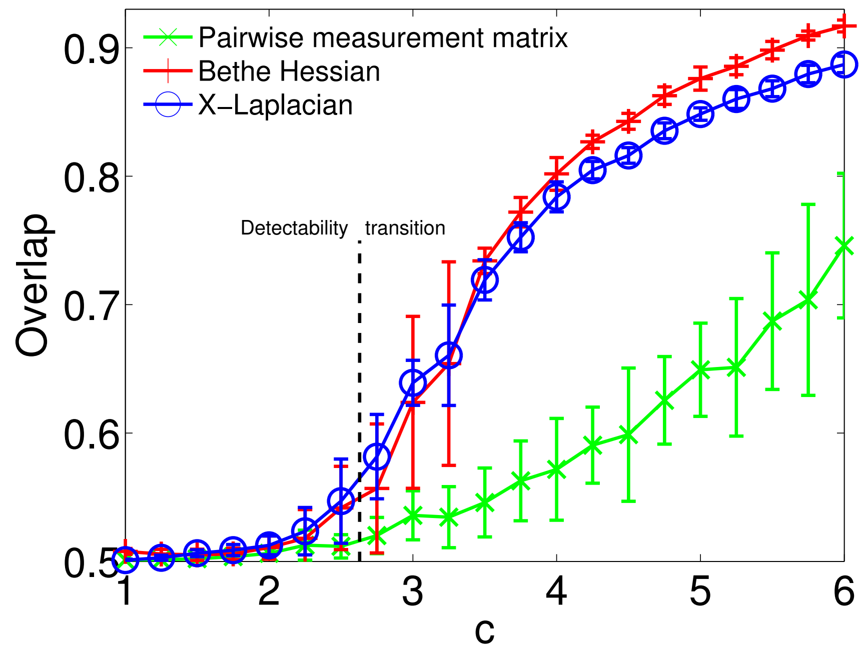
Model from [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']



n=10000, q=2, Gaussian variance: 0.75, -0.75

Clustering from sparse similarities

Model from [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']

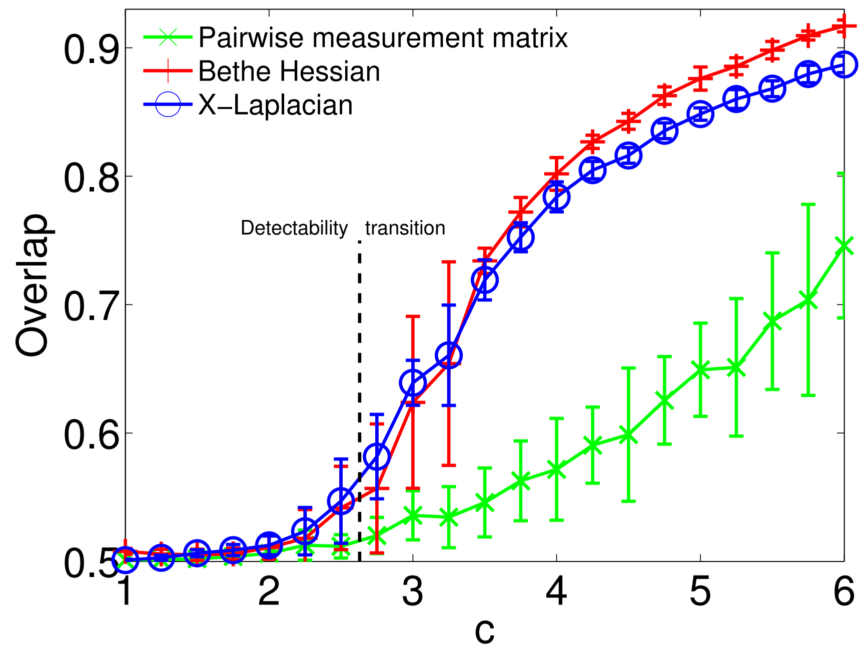


n=10000, q=2, Gaussian variance: 0.75, -0.75

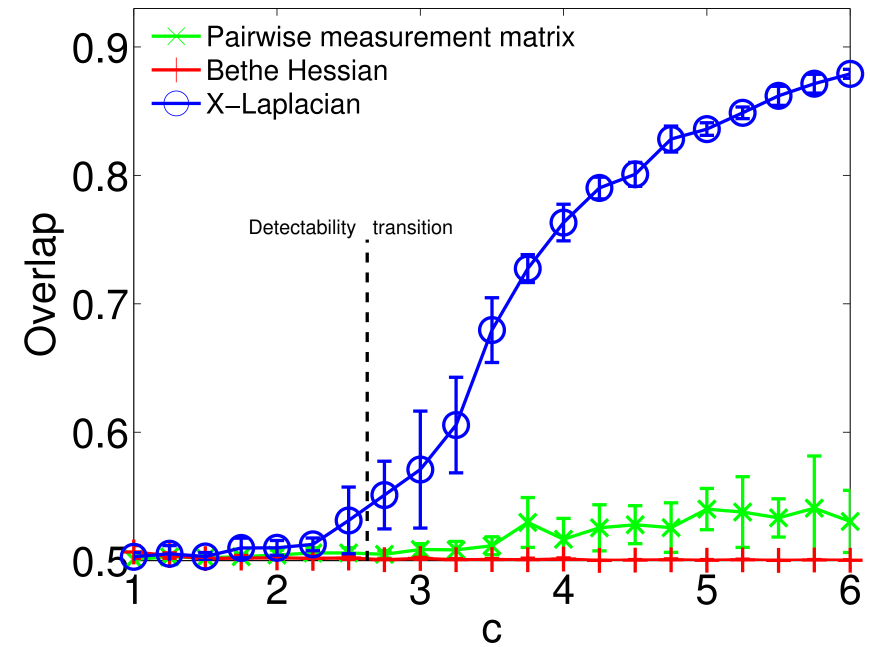
Bethe Hessian uses correct parameters, X-Laplacian does not.

Clustering from sparse similarities

Model from [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']



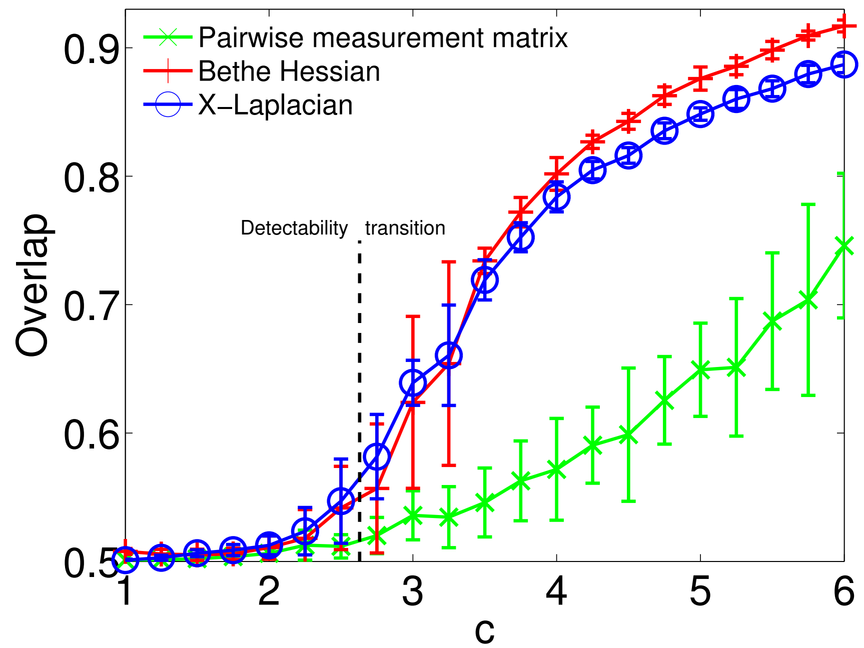
n=10000, q=2, Gaussian variance: 0.75, -0.75
Bethe Hessian uses correct parameters, X-Laplacian does not.



The same as left, but with cliques

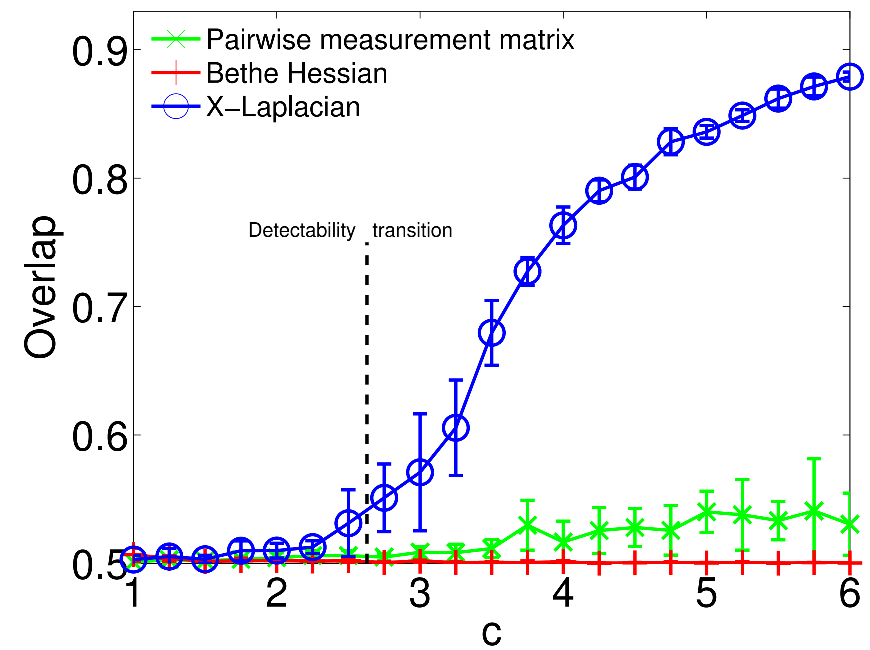
Clustering from sparse similarities

Model from [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']

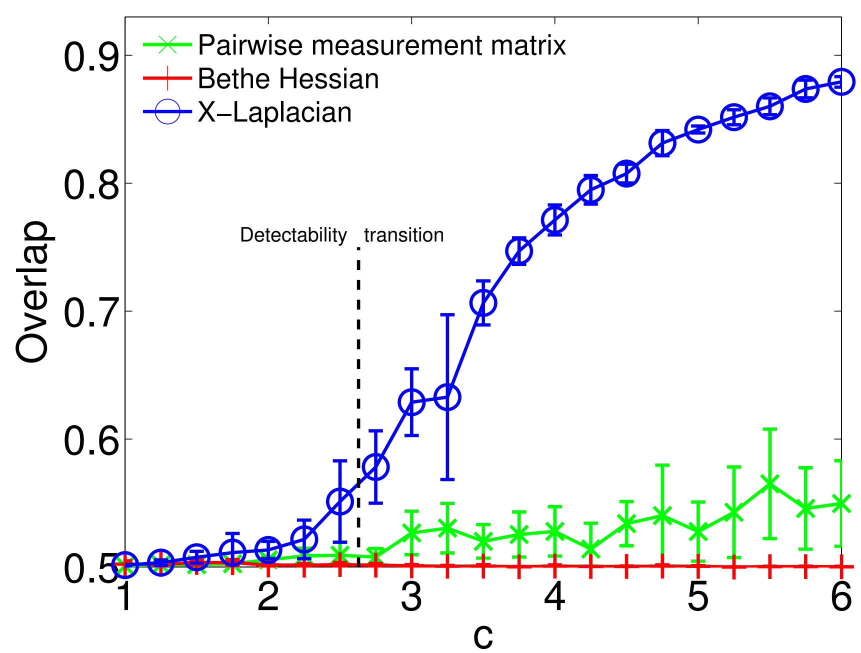


n=10000, q=2, Gaussian variance: 0.75, -0.75

Bethe Hessian uses correct parameters, X-Laplacian does not.



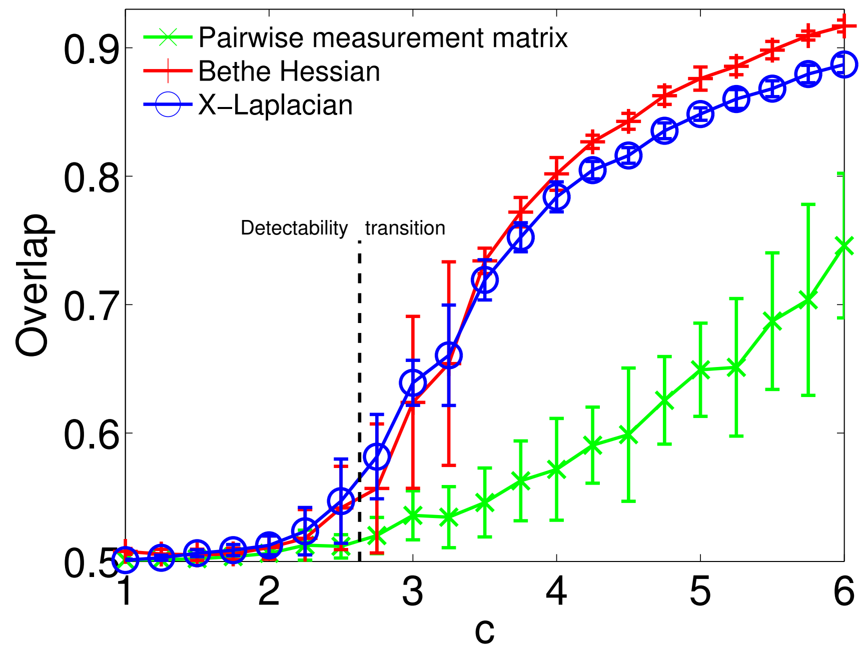
The same as left, but with cliques



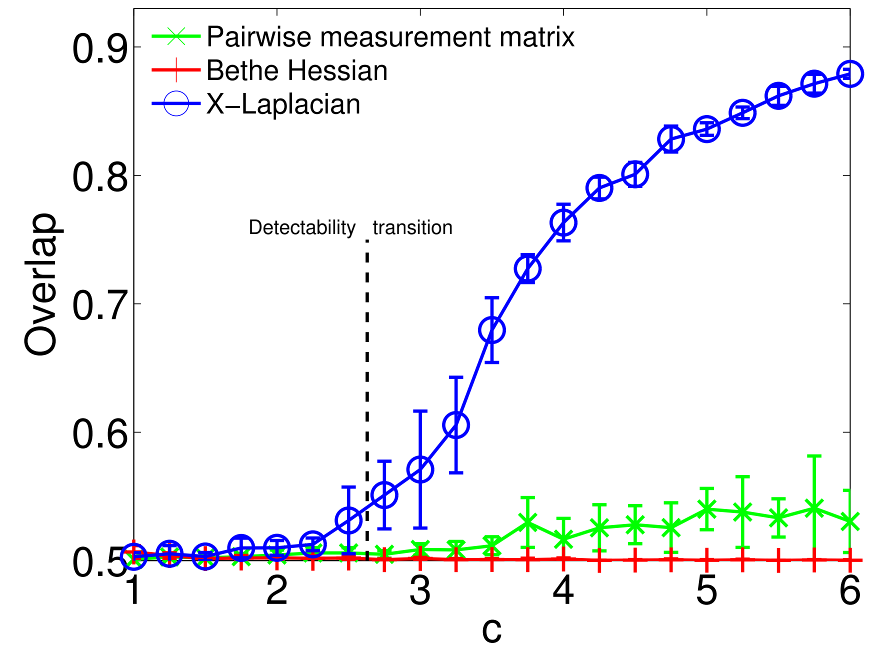
The same as top, but with hubs

Clustering from sparse similarities

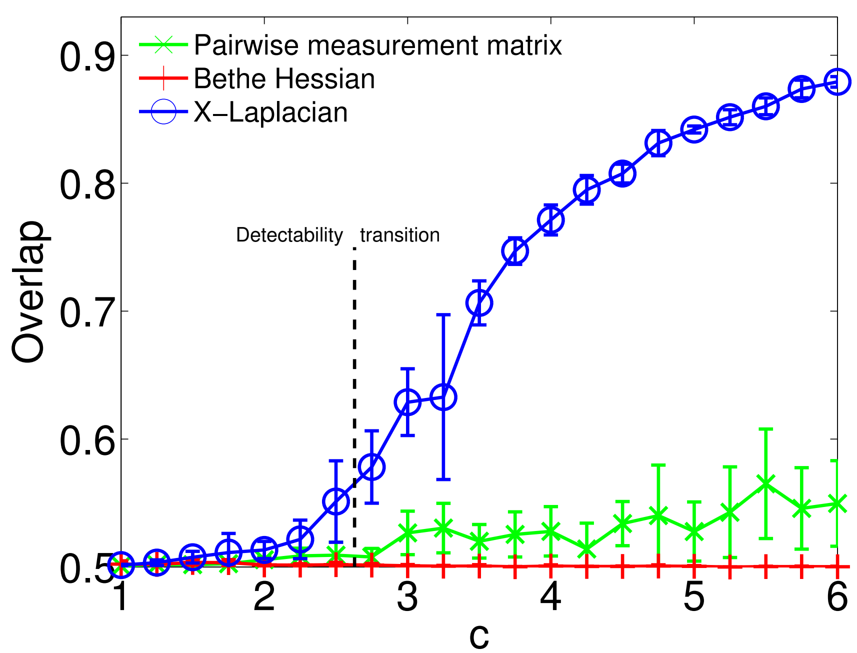
Model from [Saade/Lelarge/Krzakala/Zdeborová ISIT 16']



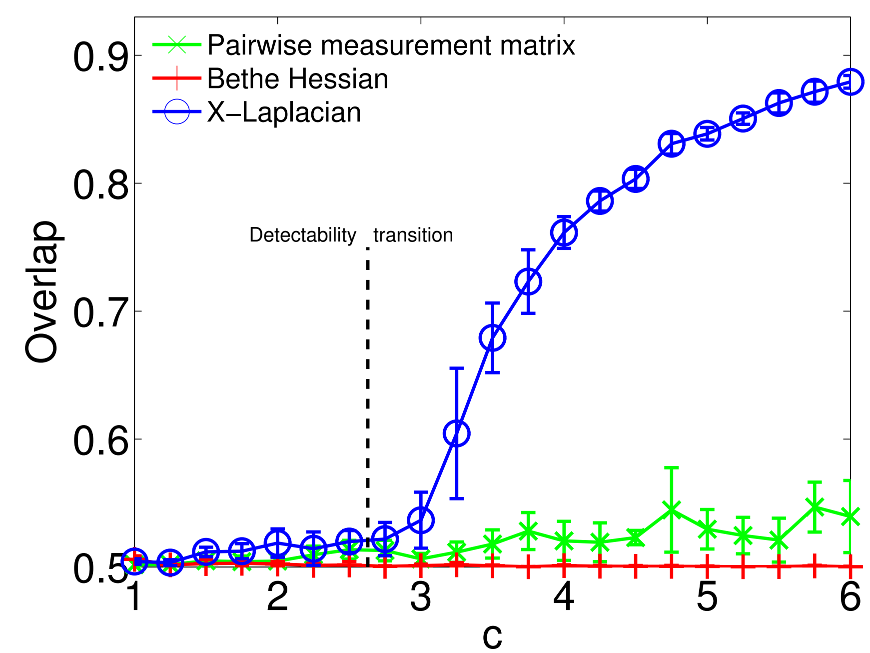
n=10000, q=2, Gaussian variance: 0.75, -0.75
Bethe Hessian uses correct parameters, X-Laplacian does not.



The same as left, but with cliques



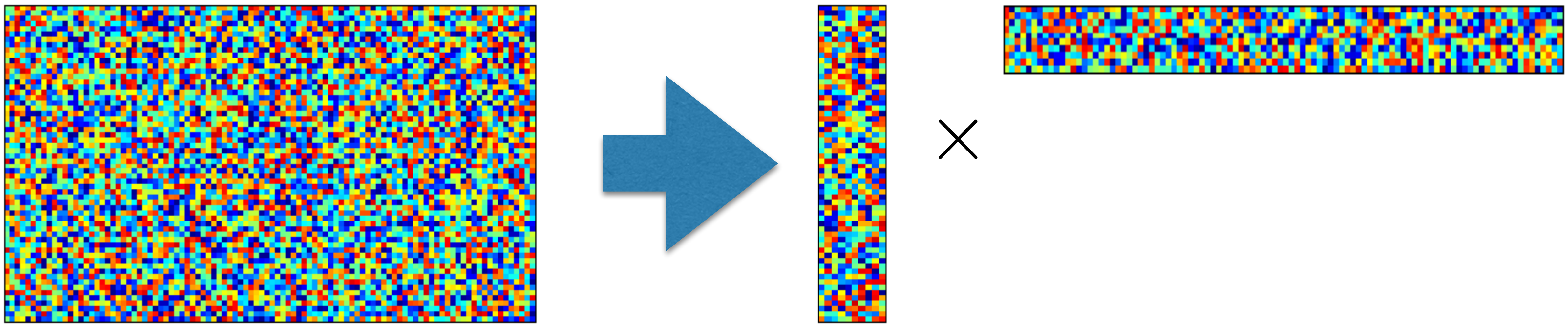
The same as top, but with hubs



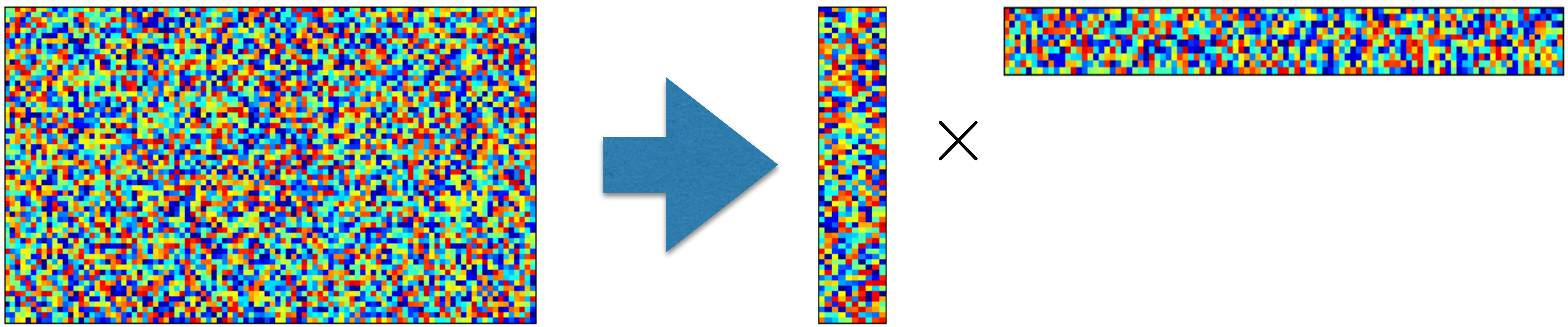
The same as top left, but with neighbors connected, as in [Javanmard/Montanari/Ricci-Tersenghi 16']

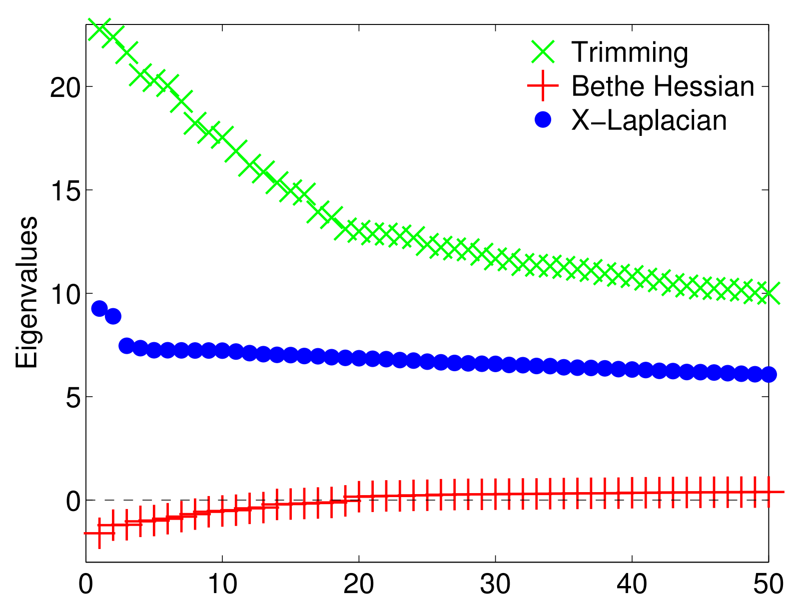
Rank estimation and matrix completion

Rank estimation and matrix completion



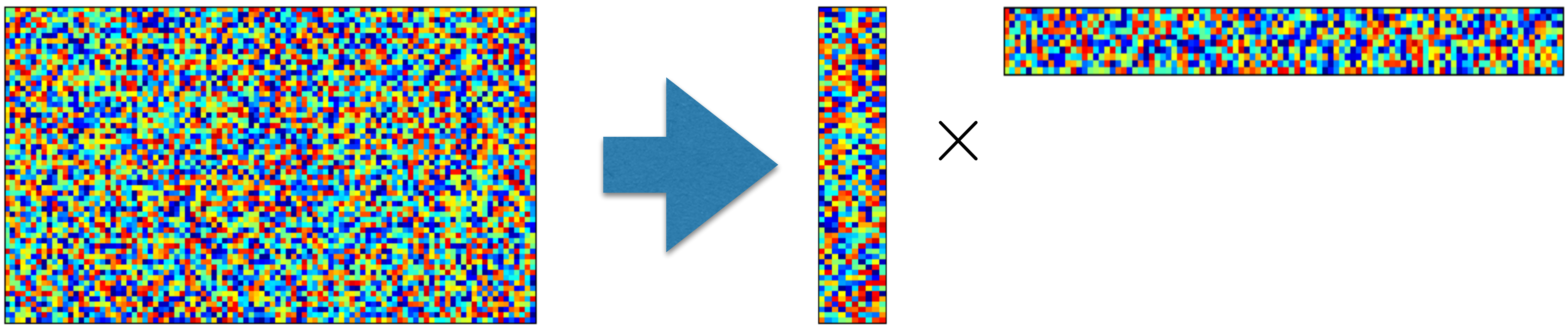
Rank estimation and matrix completion

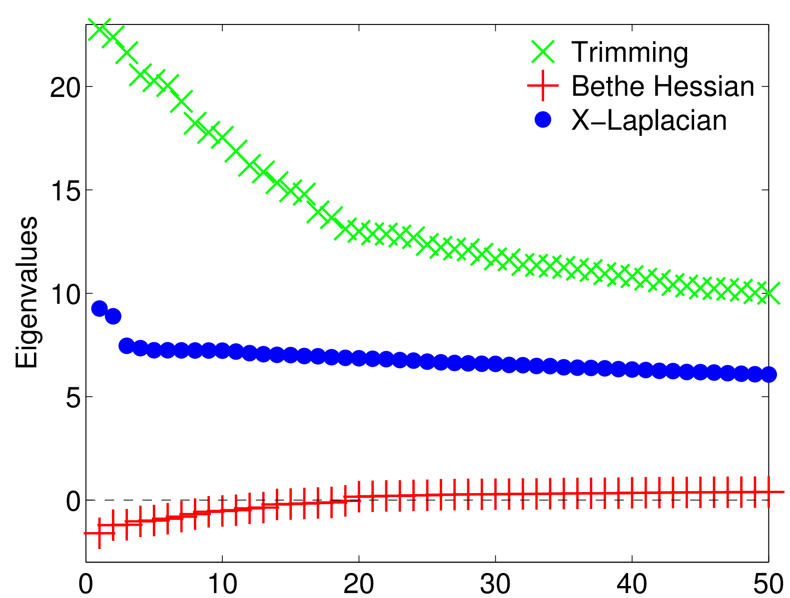




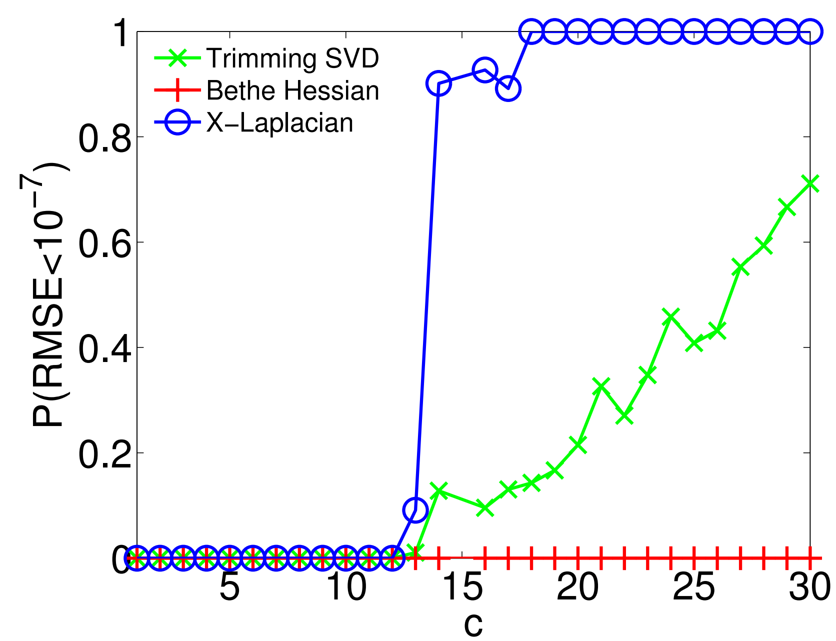
Original rank = 2
1000x10000 matrix, c=8, with 10 size-20 cliques

Rank estimation and matrix completion





Original rank = 2
1000x10000 matrix, c=8, with 10 size-20 cliques



Original rank = 3
1000x10000 matrix, with 10 size-20 cliques

Conclusions

Conclusions

- Sparsity and noise cause serious localization problems for spectral algorithms.

Conclusions

- Sparsity and noise cause serious localization problems for spectral algorithms.

- Many methods for solving localization problem, e.g. trimming, non-backtracking, Bethe Hessian,… can be seen as doing regularizations.

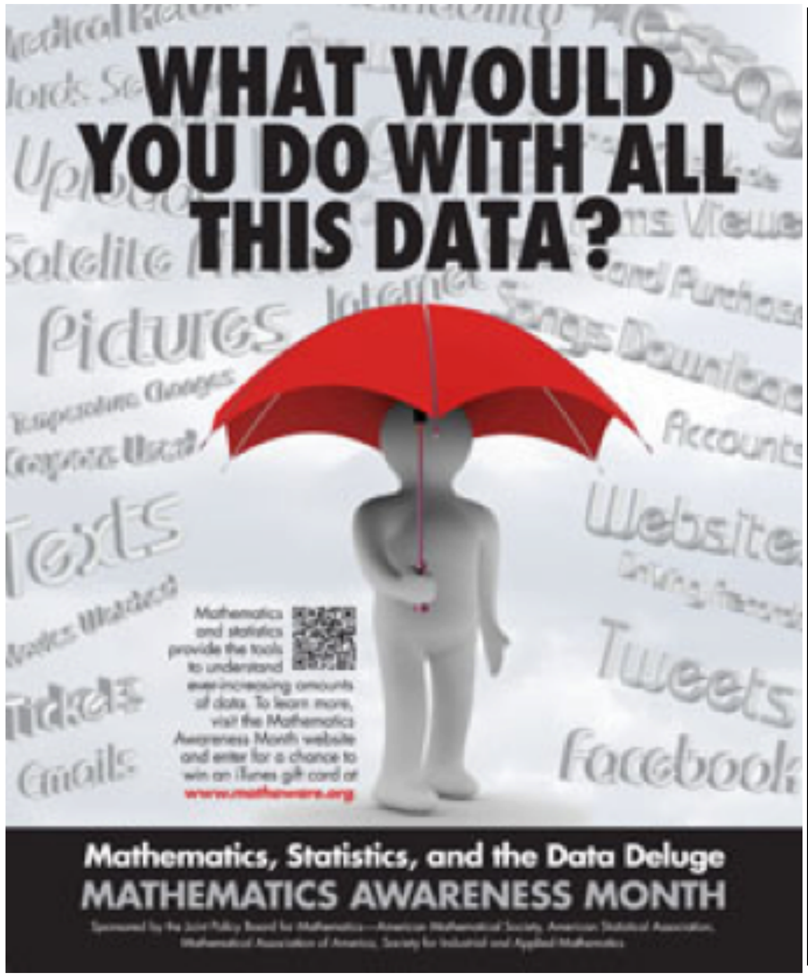
Conclusions

- Sparsity and noise cause serious localization problems for spectral algorithms.
- Many methods for solving localization problem, e.g. trimming, non-backtracking, Bethe Hessian,… can be seen as doing regularizations.
- Fixed-form regularization works only when the source of localization is known.

Conclusions

- Sparsity and noise cause serious localization problems for spectral algorithms.
- Many methods for solving localization problem, e.g. trimming, non-backtracking, Bethe Hessian,… can be seen as doing regularizations.
- Fixed-form regularization works only when the source of localization is known.
- Good regularizations can be learnt from the localized eigenvectors.
  (Demo of the X-Laplacian can be found at http://panzhang.net)

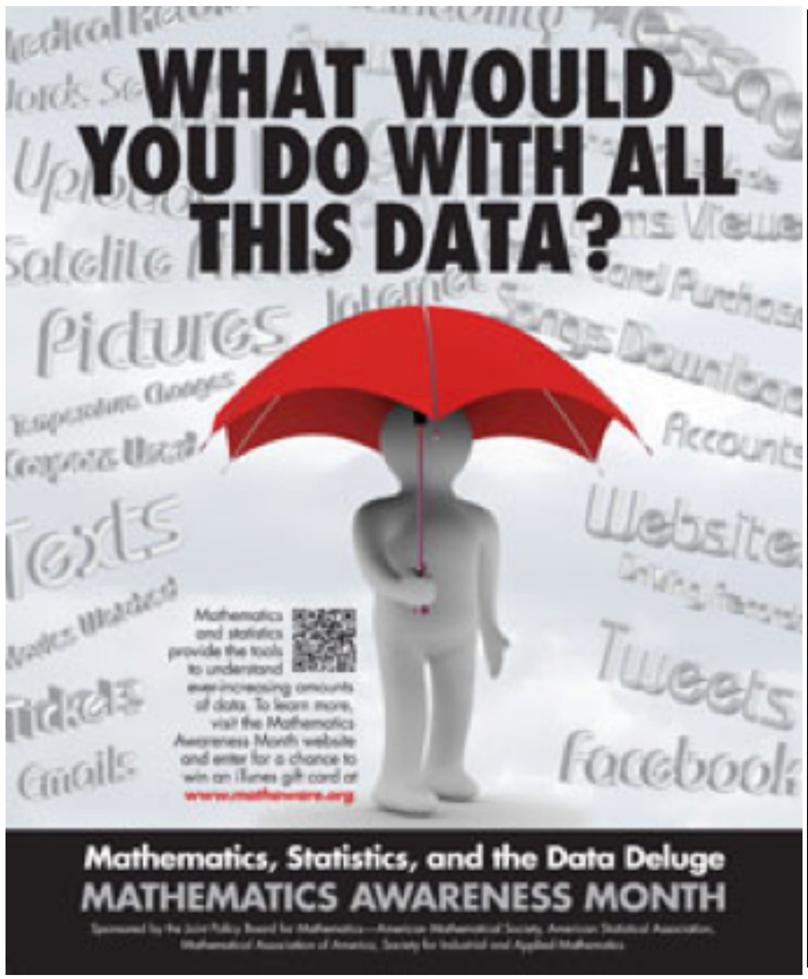
Thank you for your attention



We do it by Regularization!

Reference:
Pan Zhang, "Robust Spectral Detection of Global Structures in the Data by Learning a Regularization", *Advances In Neural Information Processing Systems 29*, 541 (2016)

Thank you for your attention



X-Laplacian

We do it by Regularization!

Reference:
Pan Zhang, “Robust Spectral Detection of Global Structures in the Data by Learning a Regularization”, *Advances In Neural Information Processing Systems 29*, 541 (2016)