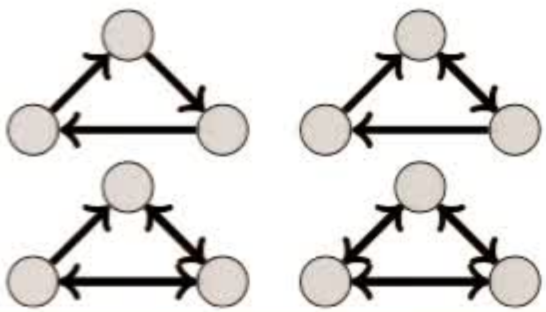
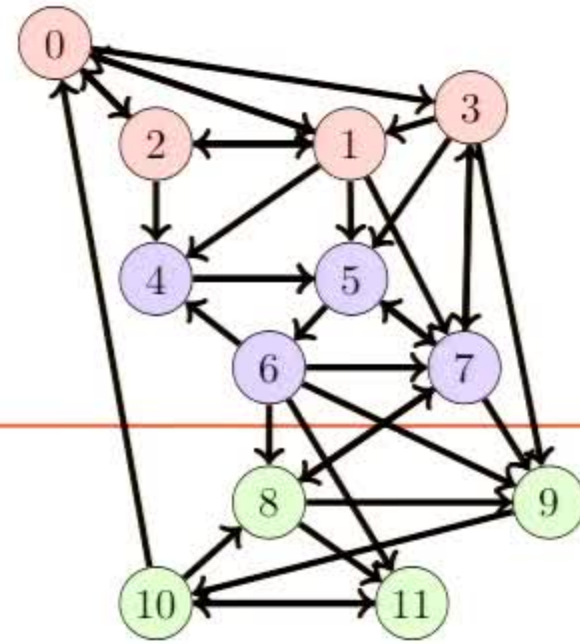
TENSOR SPECTRAL CLUSTERING
FOR PARTITIONING HIGHER-ORDER NETWORK STRUCTURES



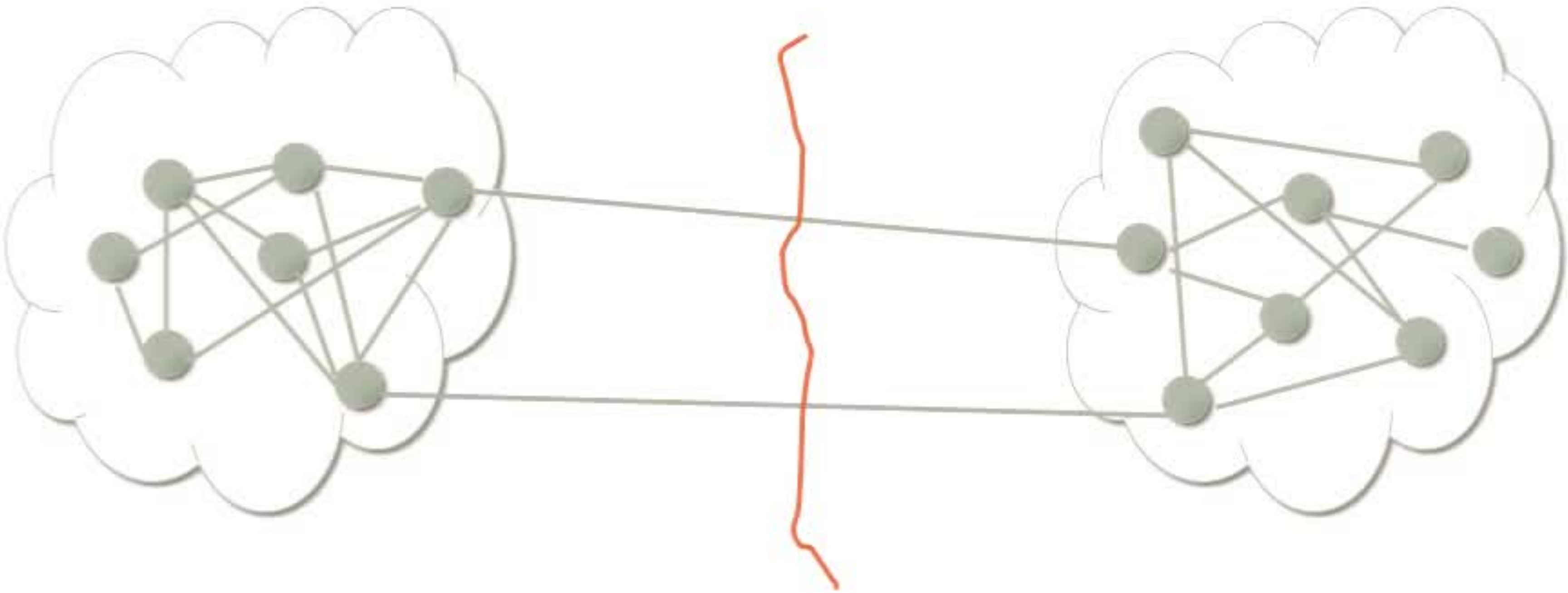


SIAM Data Mining 2015
Vancouver, BC

Austin Benson
ICME, Stanford University
arbenson@stanford.edu

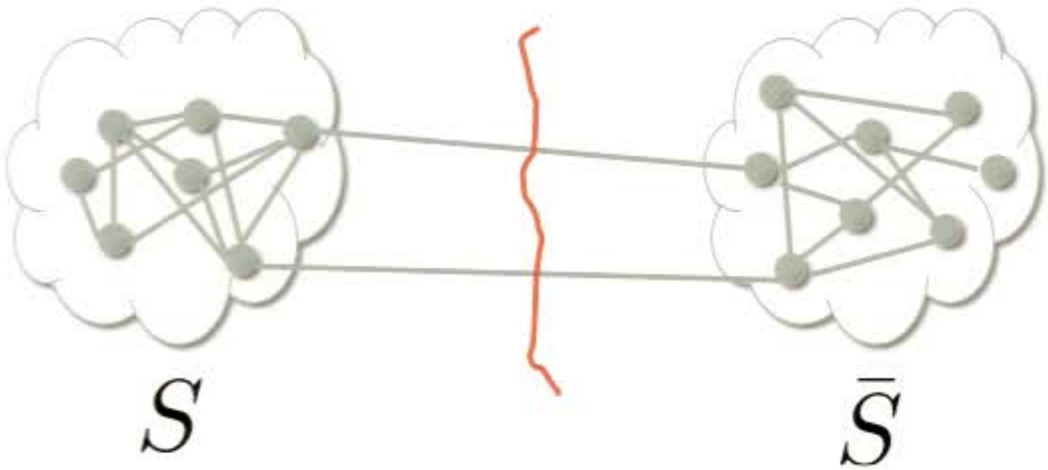
Joint work with
David Gleich, Purdue
Jure Leskovec, Stanford

Background: graph partitioning and applications



- Goal: find a ``balanced" partition of a graph that does not cut many edges.
- Applications: community structure in social networks, decompose networks into functional modules

Background: graph partitioning and clustering



A popular measure of the quality of a cut is *conductance*:

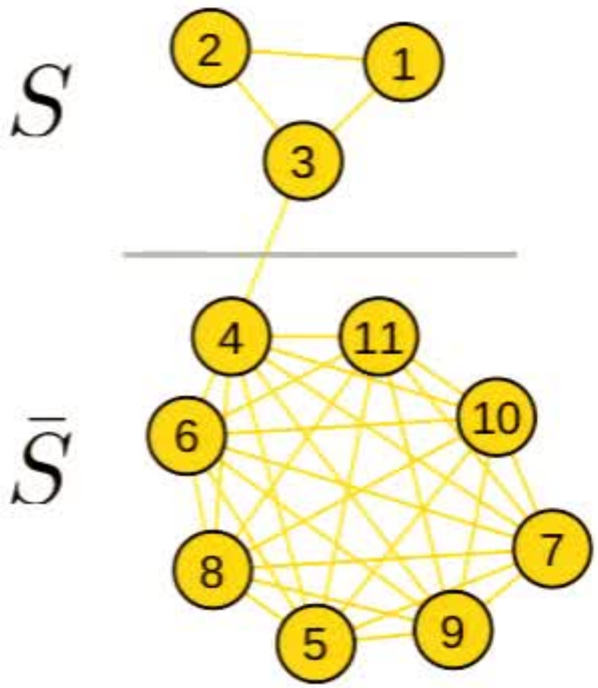
$$\min_S \phi(S) = \min_S \frac{\#(\text{edges cut})}{\min(\text{vol}(S), \text{vol}(\bar{S}))}$$

vol(S) is the number of edge end points in the set S

NP-hard in general, but there are approximation algorithms

Background: spectral clustering and random walks

$$\min_S \phi(S) = \min_S \frac{\#(\text{edges cut})}{\min(\text{vol}(S), \text{vol}(\bar{S}))}$$



$\mathbf{P}_{43} = \Pr(3 \rightarrow 4) = 1/3$

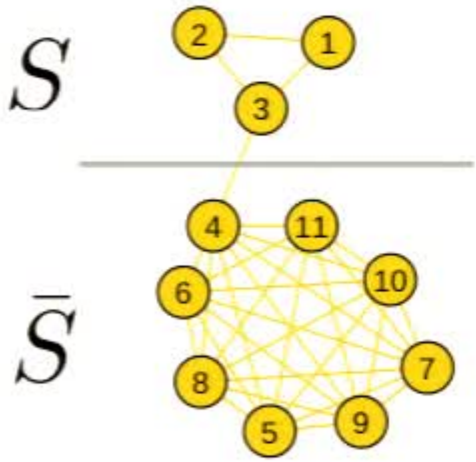
Central computation:

$$\mathbf{z}^T\mathbf{P} = \lambda_2 \mathbf{z}^T$$

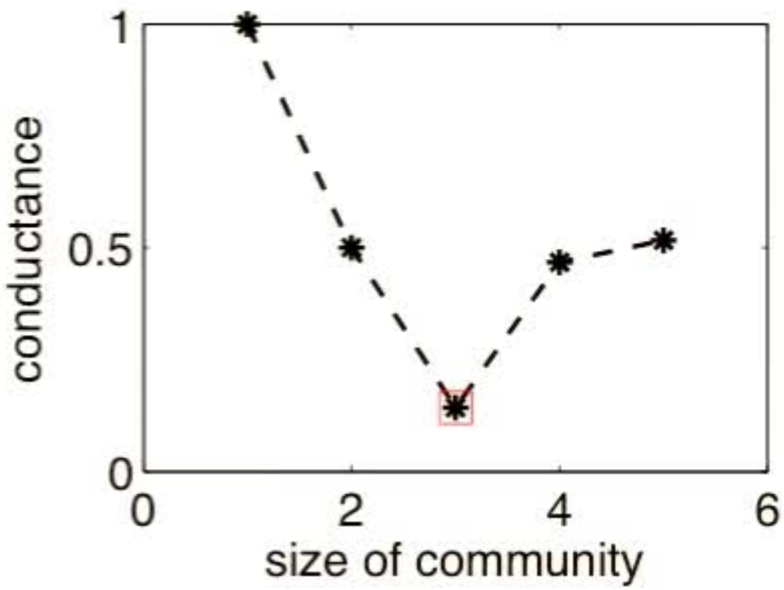
$$\mathbf{P} = \mathbf{A}^T\mathbf{D}^{-1}$$

- **P** is a transition matrix representing the random walk Markov chain.
- Entries of **z** used to partition graph.

Background: sweep cut



$$\mathbf{z}^{\mathbf{T}}\mathbf{P} = \lambda_2 \mathbf{z}^{\mathbf{T}}$$



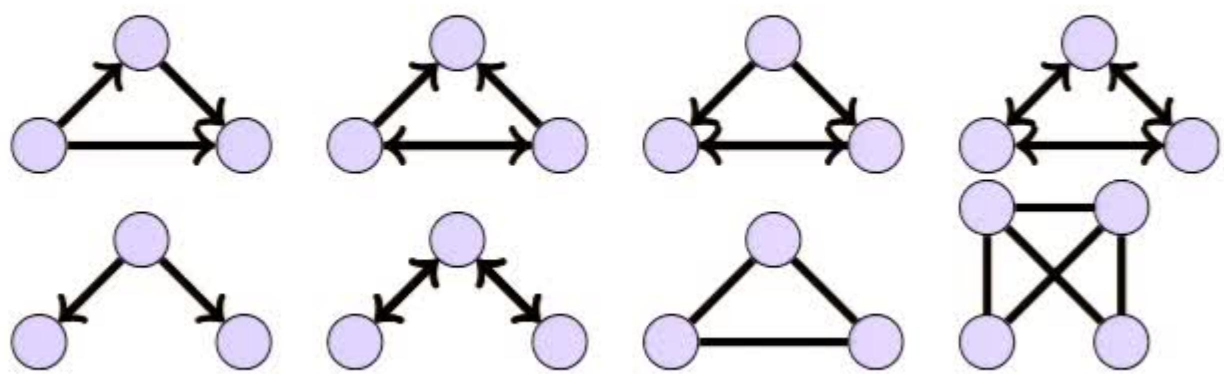
2	φ({2})
1	φ({2,1})
3	φ({2,1,3})
4	φ({2,1,3,4})
11	φ({2,1,3,4,11})
6	φ({2,1,3,4,11,6})
8	φ({2,1,3,4,11,6, 8})
10	φ({2,1,3,4,11,6,8,10})
9	φ({2,1,3,4,11,6,8,10,9})
7	φ({2,1,3,4,11,6,8,10,9,7})
5	φ({2,1,3,4,11,6,8,10,9,7,5})

Cheeger inequality guarantee on the conductance.

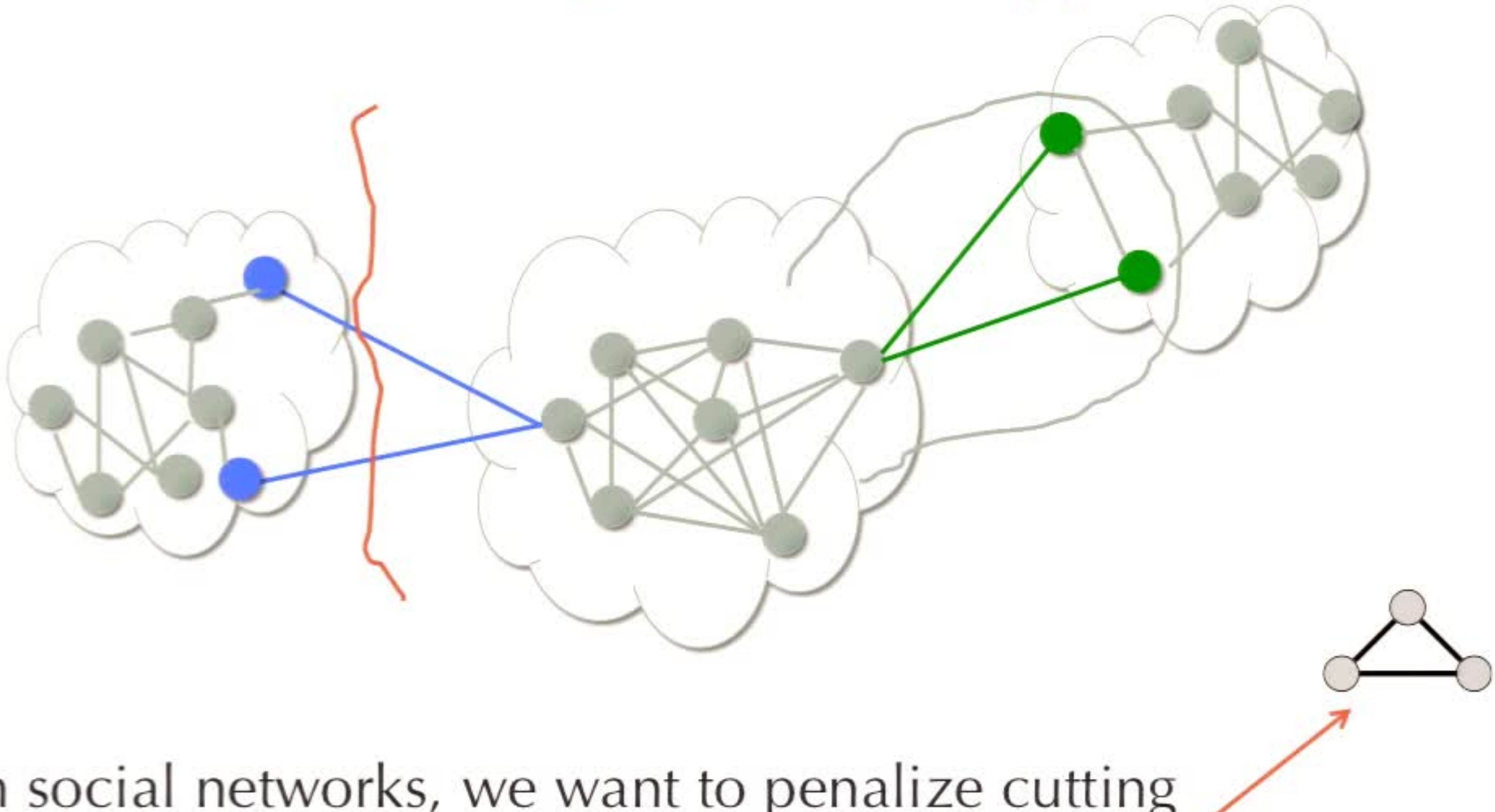
Problem: clustering methods are based on **edges** and do not use higher-order relations or **motifs**, which can better model problems.

Edges

Motifs

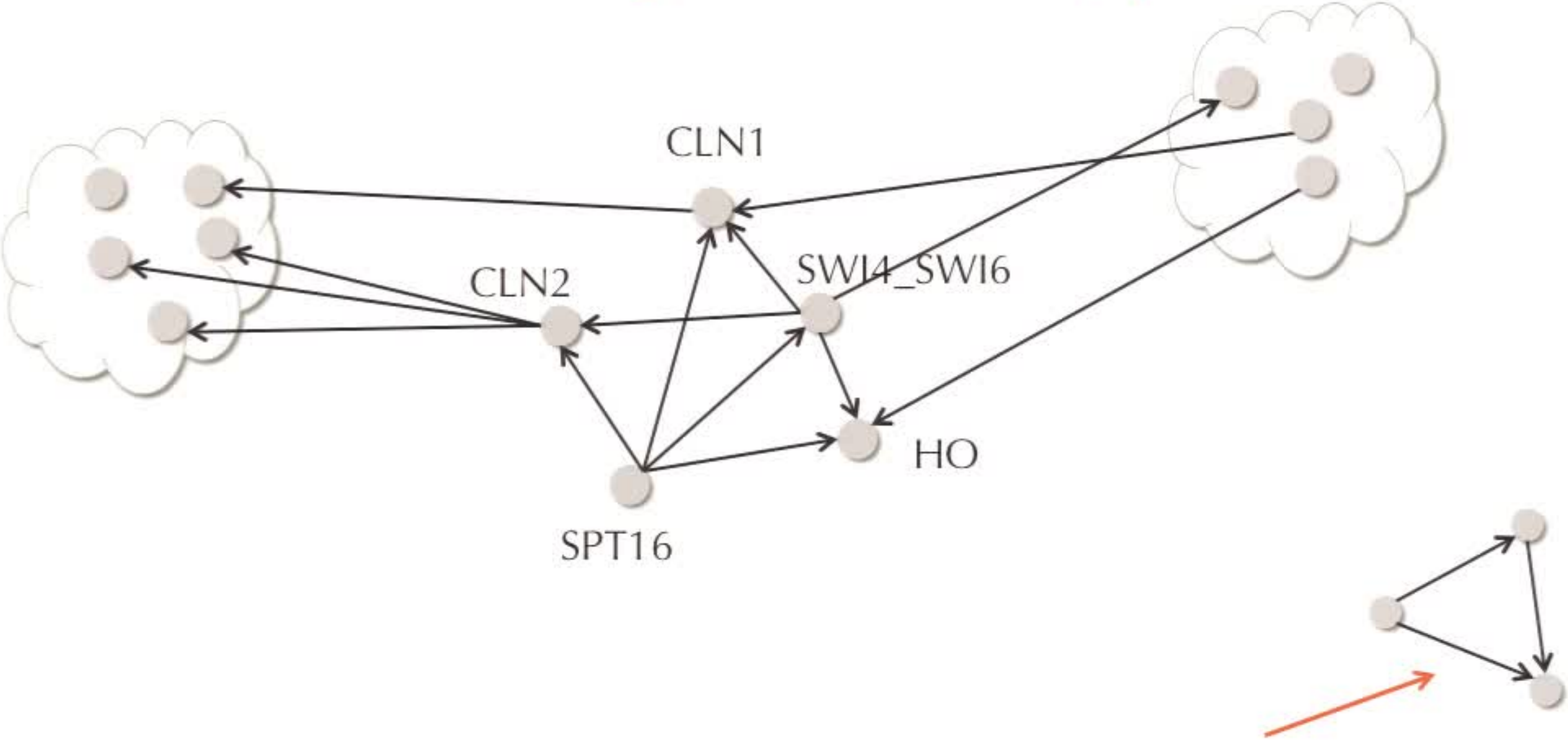


Problem: current methods only consider edges … and that is not enough to model many problems



In social networks, we want to penalize cutting triangles more than cutting edges. The triangle motif represents stronger social ties.

Problem: current methods only consider edges … and that is not enough to model many problems



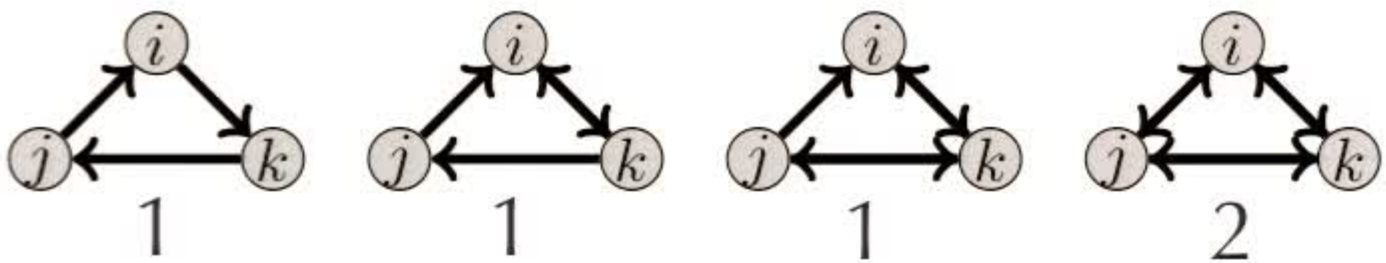
In transcription networks, the ``feedforward loop" motif represents biological function. Thus, we want to look for clusters of this structure.

Our contributions

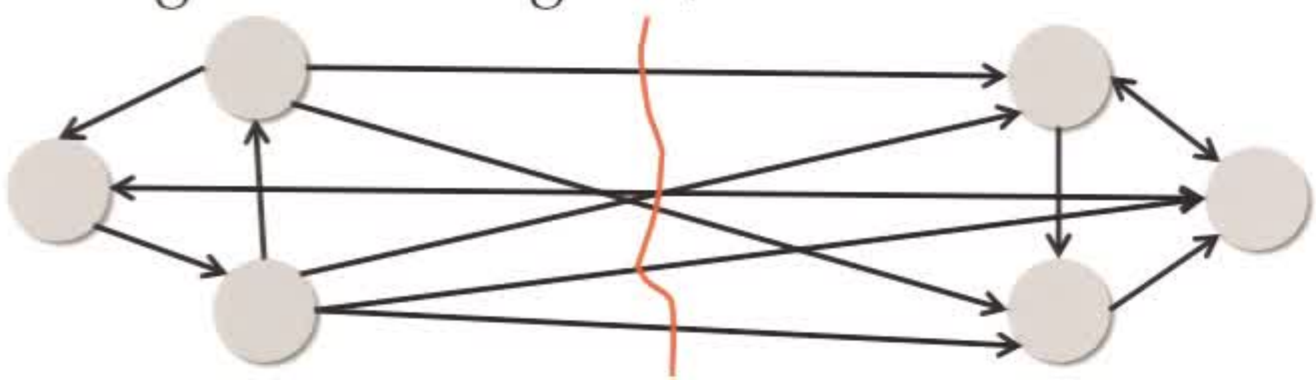
1. We generalize the definition of conductance for motifs.
2. We provide an algorithm for optimizing this objective:

Tensor Spectral Clustering (TSC) Algorithm:

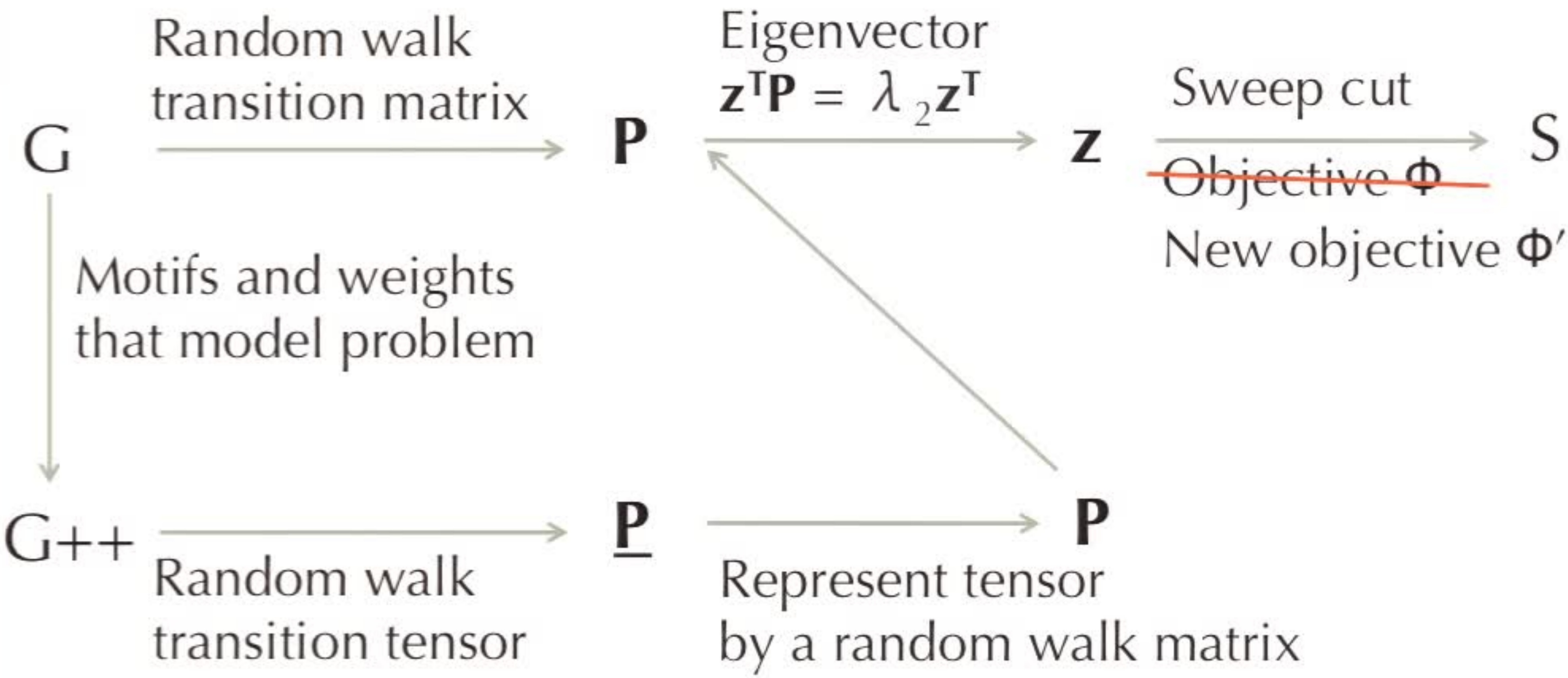
Input: set of motifs and weights



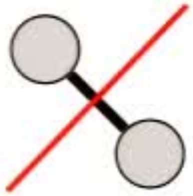
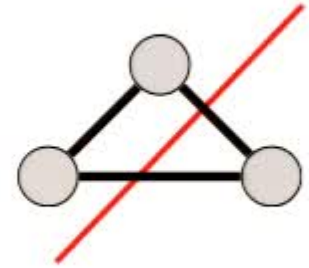
Output: Partition of graph that does not cut the motifs corresponding to the weights (and some normalization).



Roadmap of Tensor Spectral Clustering



Motif-based conductance

Edges cut  → Triangles cut 

vol(S) =
#(edge end points in S)

→

$\mathrm{vol}_3(S)$ =
#(triangle end points in S)

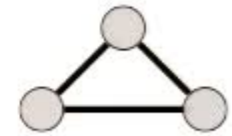
$$\phi(S) = \frac{\#(\text{edges cut})}{\min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))}$$

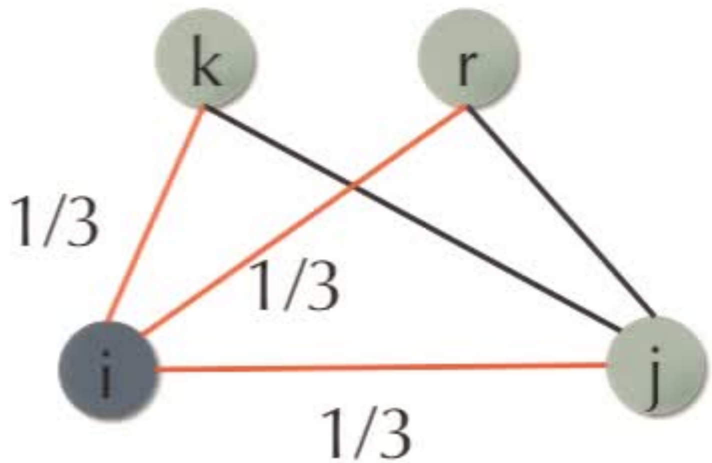
→

$$\phi_3(S) = \frac{\#(\text{triangles cut})}{\min(\mathrm{vol}_3(S), \mathrm{vol}_3(\bar{S}))}$$

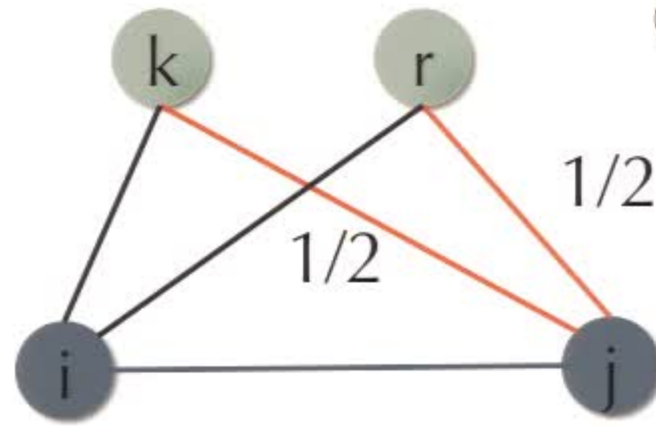
Our algorithm is a heuristic for minimizing this objective based on the random walk interpretation of spectral clustering.

First-order → second-order Markov chain

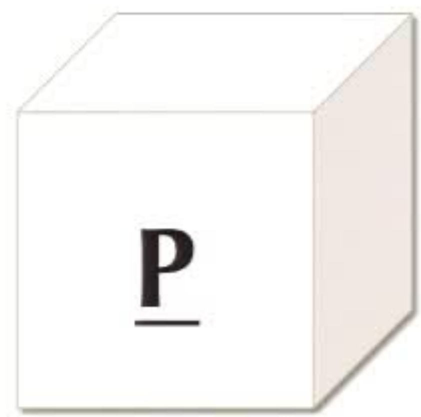




Prob(i → j) = 1/3



Prob((i, j) → (j, k)) = 1/2

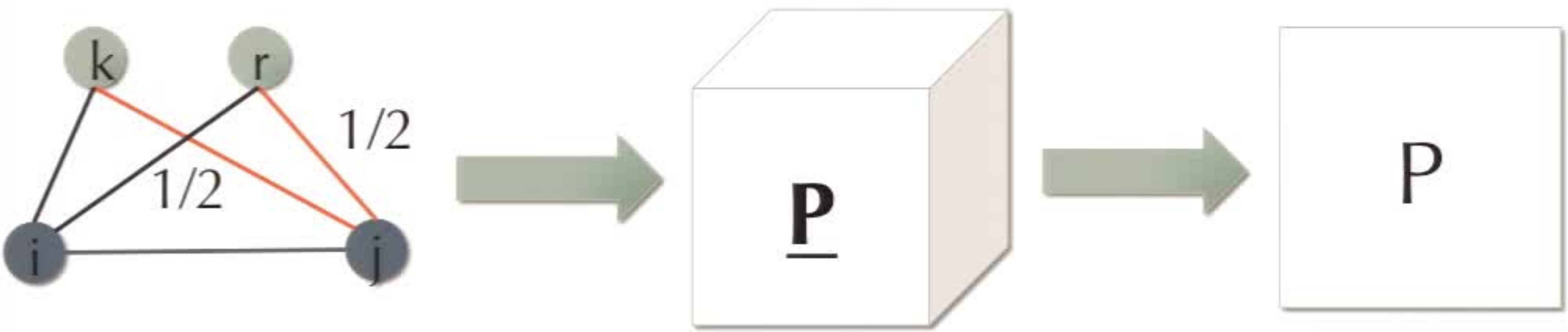


$\boldsymbol{P}(j,i) =$
$\Pr(S_{t+1} = j \mid S_t = i)$

$\underline{\boldsymbol{P}}(i,j,k) =$
$\Pr(S_{t+1} = k \mid S_t = j,\ S_{t-1} = i)$

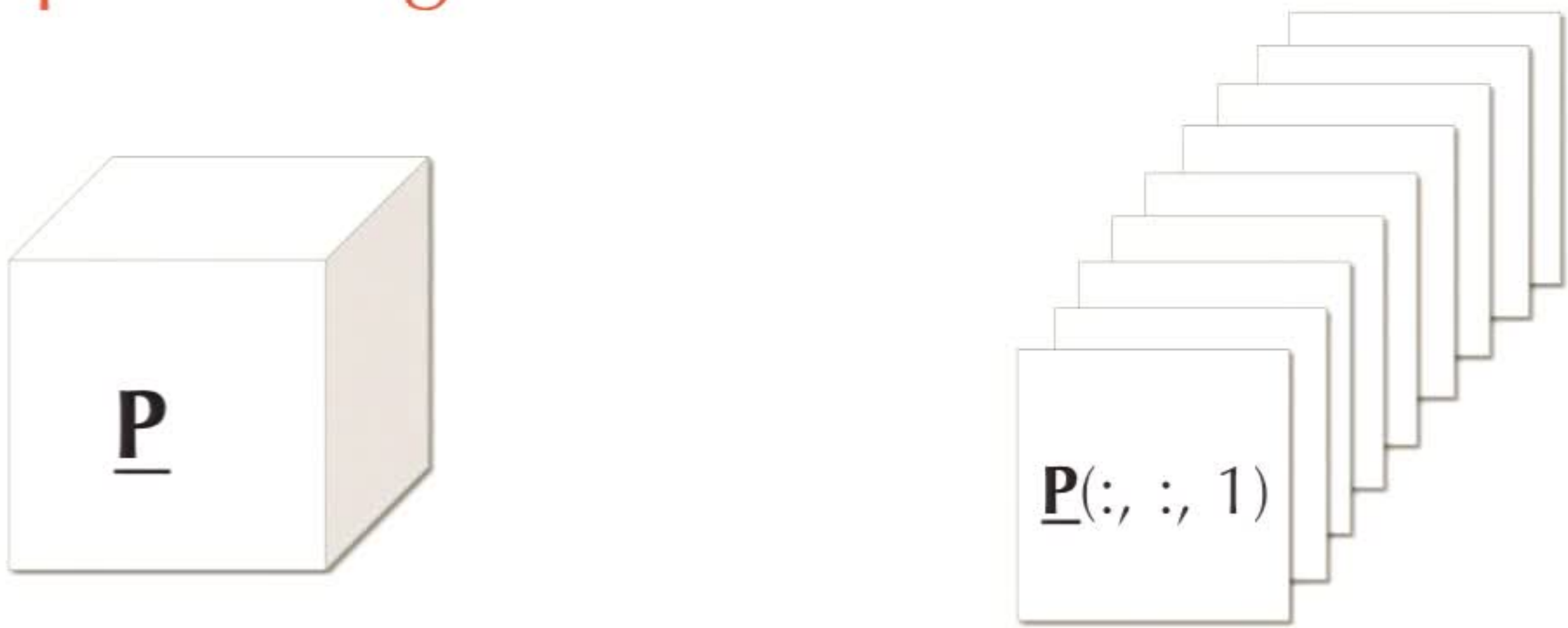
Representing the transition tensor

- Problem 1: Even stationary distribution of second-order Markov chain is $O(n^2)$ storage.
- Problem 2: Tensor eigenvectors are hard to compute.



- Idea: Represent the tensor as a matrix, respecting the motif transitions of the data. Then we can compute eigenvectors.

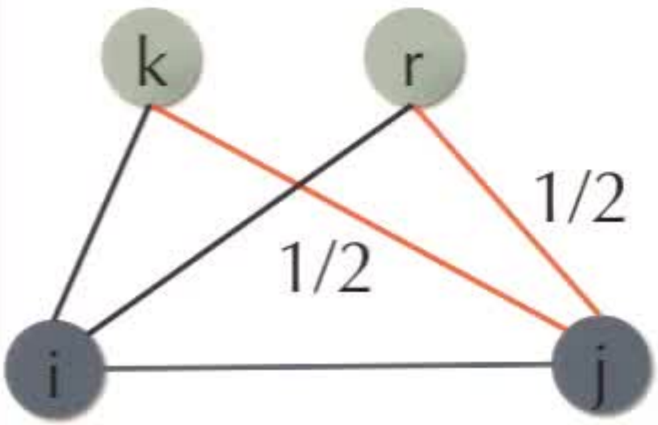
Representing the transition tensor



$$\underline{\boldsymbol{P}}(i, j, k) = \Pr\left(S_{t+1} = k \mid S_t = j,\ S_{t-1} = i\right)$$

- Each slice of transition tensor is a transition matrix.
- Convex combinations of these slices is a transition matrix.
- Which combination should we use?

Transition tensor → transition matrix



$$\underline{\boldsymbol{P}}(i,j,k) = \Pr\left(S_{t+1} = k \mid S_t = j,\ S_{t-1} = i\right)$$

$$\boldsymbol{R} = \left[\underline{\boldsymbol{P}}(:,:,1) \quad \underline{\boldsymbol{P}}(:,:,2) \quad \ldots \quad \underline{\boldsymbol{P}}(:,:,n)\right]$$

1. Compute tensor PageRank vector [Gleich+14]

$$\alpha \boldsymbol{R}\left(\boldsymbol{x} \otimes \boldsymbol{x}\right) + (1-\alpha)\boldsymbol{v} = \boldsymbol{x},\ x_k \geq 0,\ \boldsymbol{e}^{\mathsf{T}}\boldsymbol{x} = 1$$

2. Collapse back to probability matrix

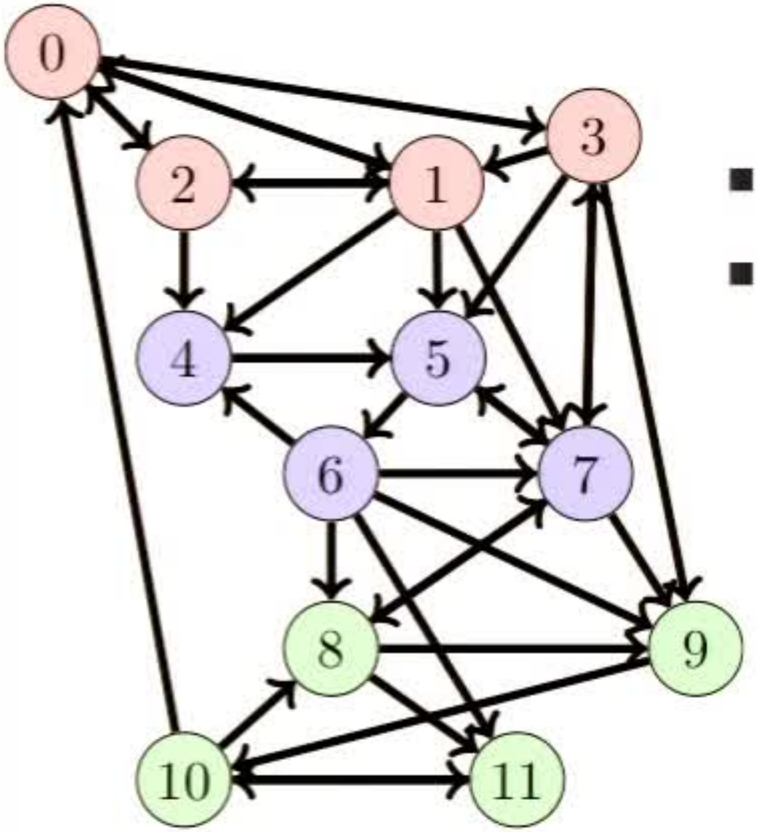
$$\boldsymbol{P}[\boldsymbol{x}] := \sum_{k=1}^{n} x_k \underline{\boldsymbol{P}}(:,:,k)$$

Convex combination of slices $\underline{\mathbf{P}}(:, :, k)$

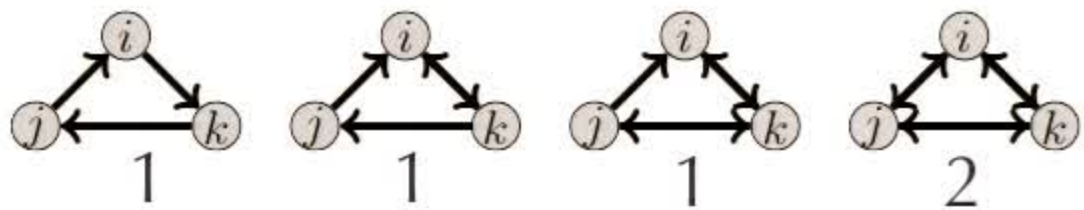
Theorem

Suppose there is a partition of the graph that does not cut *any* of the motifs of interest. Then the second left eigenvector of the matrix $\mathbf{P}[x]$ properly partitions the graph.

Layered flow network

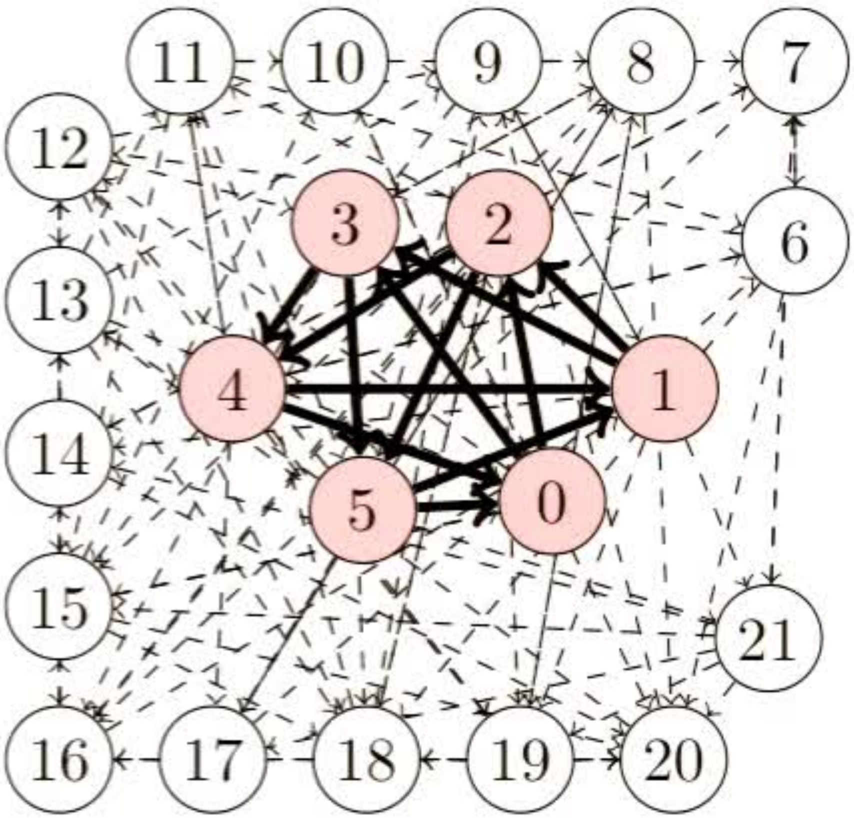


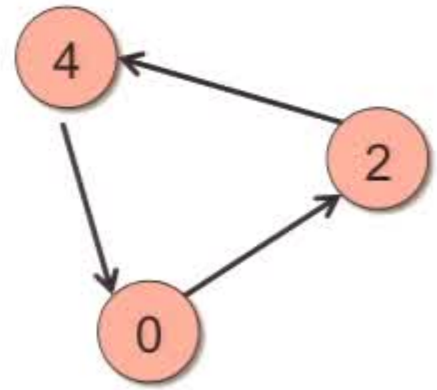
- The network "flows" downward
- Use directed 3-cycles to model flow:



- Tensor spectral clustering: {0,1,2,3}, {4,5,6,7}, {8,9,10,11}
- Standard spectral: {0,1,2,3,4,5,6,7}, {8,10,11}, {9}

Planted motif communities

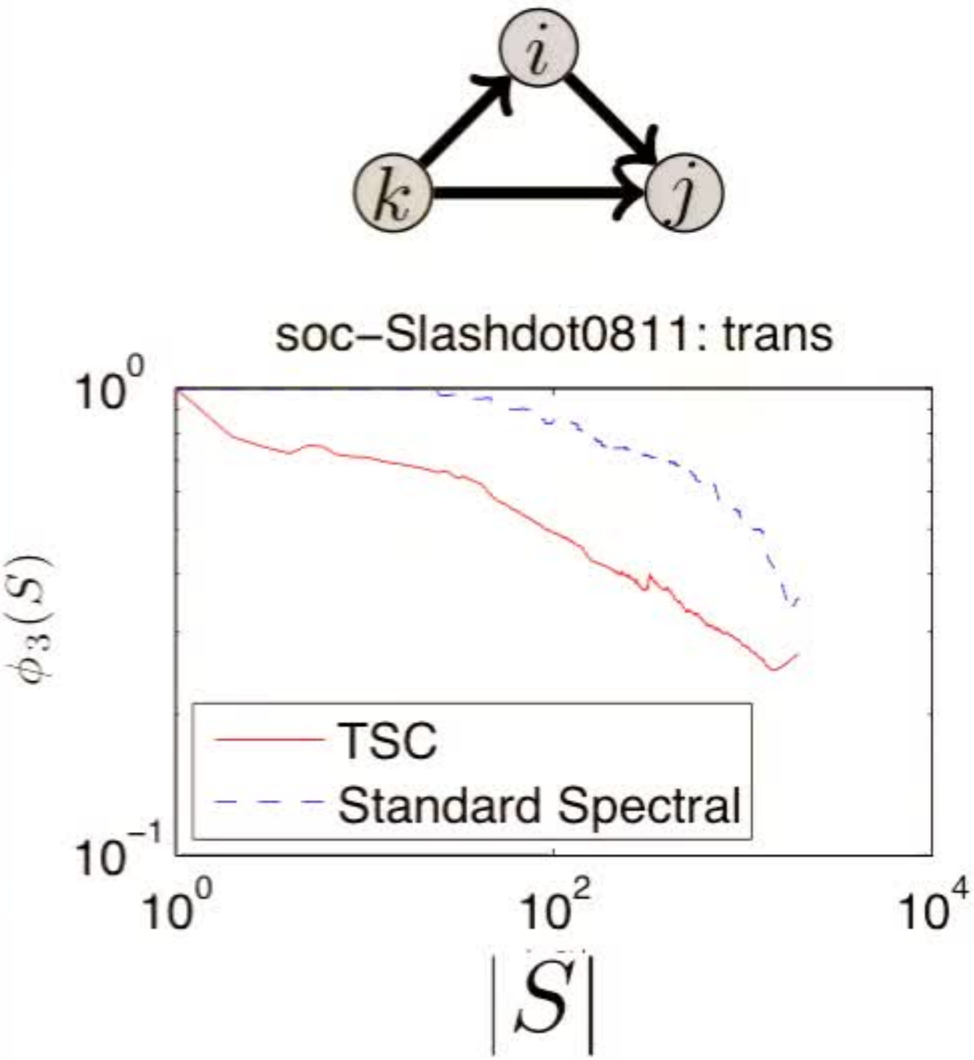


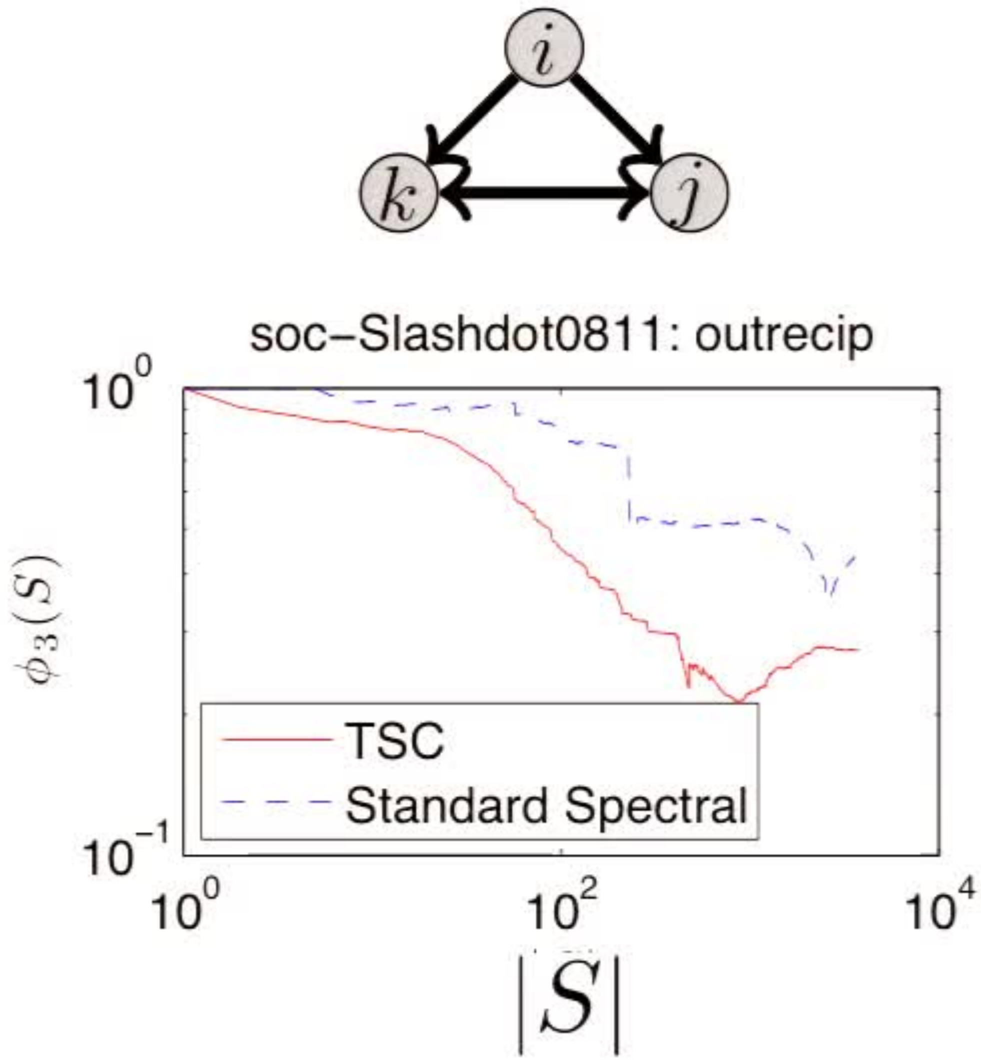


Plant a group of 6 nodes with high motif frequency into a random graph.

- Tensor spectral clustering: {0,1,2,3,4,5,12,13,16}
- Standard spectral: {0,1,4,5,9,11,16,17,19,20}

Some motifs on large networks





Summary of results

1. New objective function: motif conductance
2. Tensor Spectral Clustering algorithm that is a heuristic for minimizing motif conductance.
   Input: different motifs and weights
   Output: partition minimizing the number of motifs cut corresponding to the weights

More recent work: algorithm with Cheeger-like inequality for motif conductance.

Tensor Spectral Clustering
for partitioning higher-order network structures

Thanks!

arbenson@stanford.edu
github.com/arbenson/tensor-sc