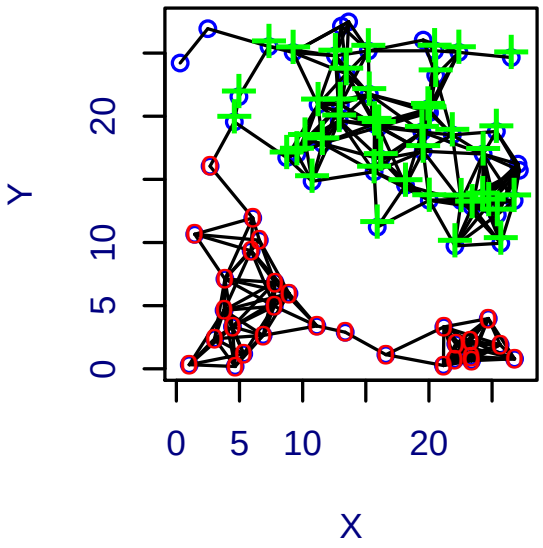
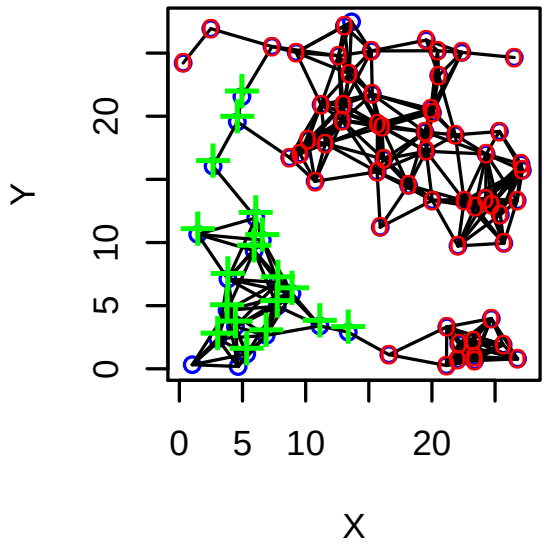
MAE 298, Lecture 7

April 25, 2006

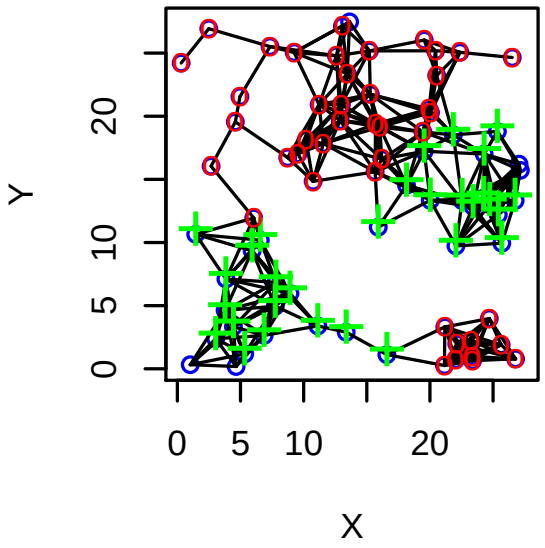
Timescale, $\tau_1 = 5314\ t_o$



Timescale, $\tau_2 = 1092\ t_o$



Timescale, $\tau_3 = 157\ t_o$



"Partitioning Networks (Spectral Methods)"

Review (class so far)

- Random graphs:
  – basic graph properties (diameter, degree distribution, ...)
  – phase transitions

- Power laws:
  – signatures of phase transitions
  (size distribution of components at the critical point)
  – seen pervasively in degree distribution of real-world networks
  – for $2 < \gamma < 3$, finite mean but infinite standard deviation.

- Preferential attachment (PA)/ Cumulative advantage

- Robustness of power law random graphs

- Optimization as an underlying mechanism for PA
  – also gives rise to saturation of "preferential" attachment

Last time

- Software tools.
- Especially important for connecting *data* with the structure of networks.

Partitioning networks

Challenge: Finding groups of vertices that have a high density of edges within them, with a lower density of edges between groups.

(Find partitions that divide the graph into smaller pieces, while cutting the minimum number of edges possible).

Partitioning networks: approaches

From Graph Theory (Spectral methods):

- Identifying sub-clusters
- Mixing times for information
- “Betweenness”

Incorporating data attributes:

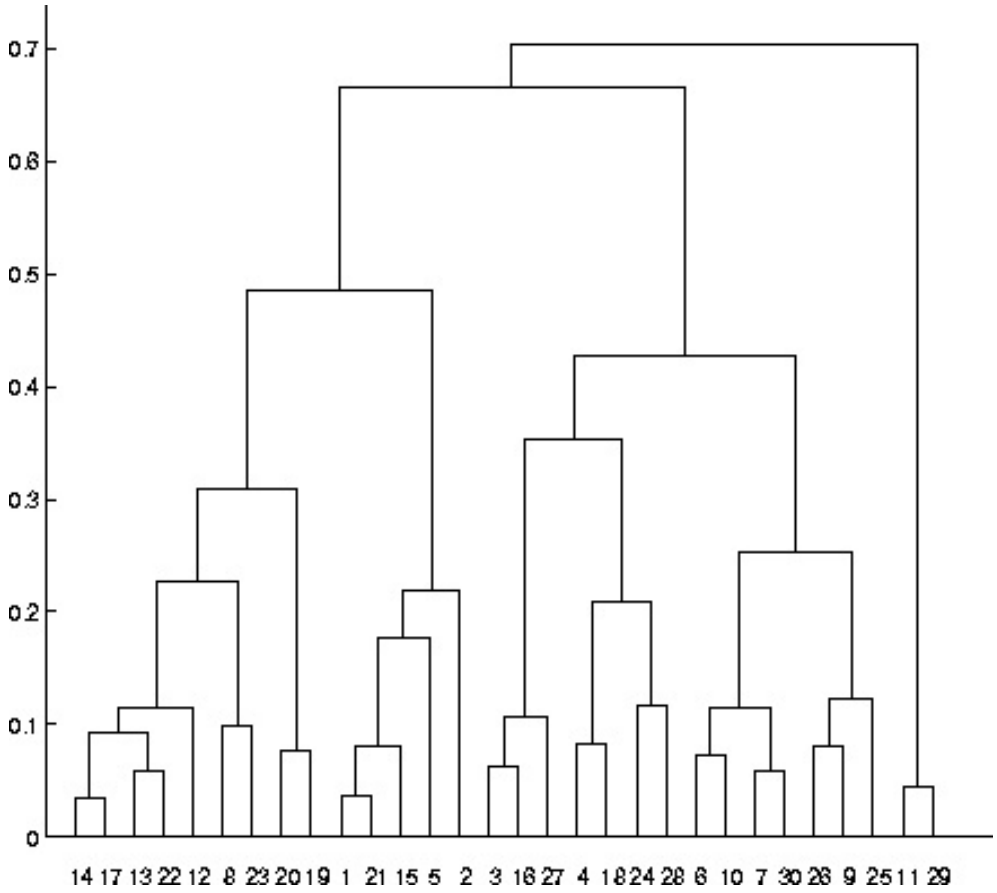
- “Mixing” (of links amongst different data types)
- Community structure

Graph Theory: Spectral Methods

- Looks only at graph structure without taking into account node or edge attributes.
- Based on random walks on the graph.
- Extremely useful for quantifying properties of information flow on networks and algorithms *(cover time, mixing time, etc)*.
- Quantitative (no subjectivity)

Why are other methods subjective?

e.g., Dendrograms



Iterate from bottom up.

Stop when have desired number of clusters.

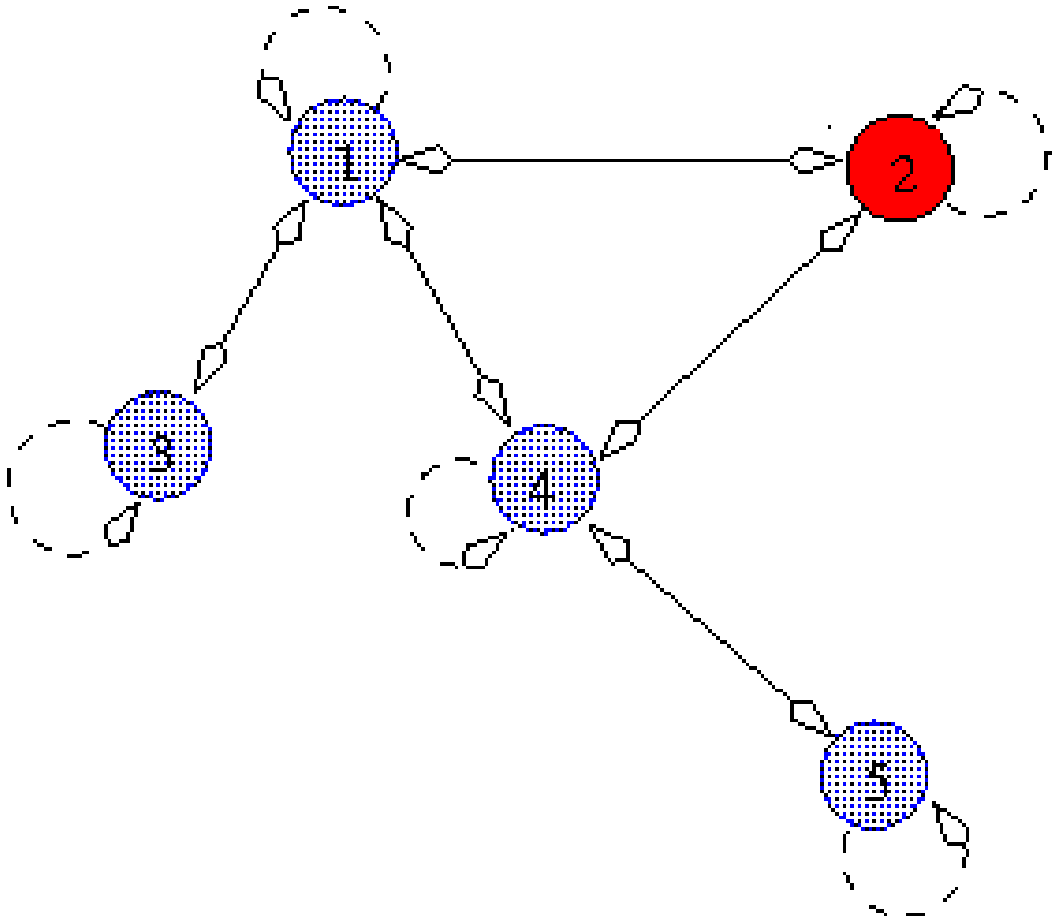
Spectral methods

- Considers the eigenvalues and eigenvectors of the graph.
- Take adjacency graph and consider a random walk on that graph.
- For now impose constraint that adjacency matrix be the *symmetric*. (Need this to ensure real eigenvalues).

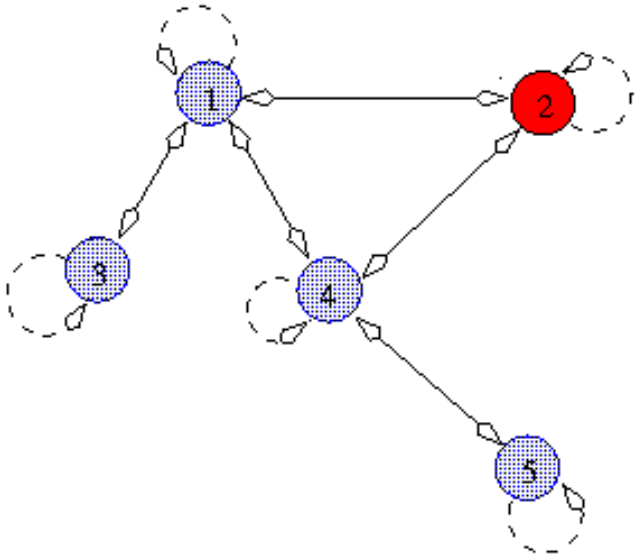
For now, three important measures from random walks:

- Cover time
- Mixing time
- Relaxation time

Sample graph structure



Random walk: State Transition Matrix (Column-normalize the adjacency matrix)



$$M = \begin{pmatrix} 1/4 & 1/3 & 1/2 & 1/4 & 0 \\ 1/4 & 1/3 & 0 & 1/4 & 0 \\ 1/4 & 0 & 1/2 & 0 & 0 \\ 1/4 & 1/3 & 0 & 1/4 & 1/2 \\ 0 & 0 & 0 & 1/4 & 1/2 \end{pmatrix}$$

M will have a basis set of eigenvectors $\{\vec{u}_i\}$ and corresponding eigenvalues λ_i.

Spectral values and graph isomorphisms

- Consider two networks with the same exact topology, but different labeling of edges (i.e., two *isomorphic* networks).

- Two isomorphic graphs will have the same eigen-spectra (same eigenvalues and eigenvectors).

- The eigenvalues and eigenvectors can be used to determine is two networks are *not* isomorphic.
  – If eigen-spectra different can guarantee graphs not isomorphic.
  – But if eigen-spectra the same cannot guarantee graphs are isomorphic (but it's a good clue, especially for large graphs).

Eigenvalues

- Let the vector $\vec{v}$ denote the probability of finding the random walk on any node (i.e., v_i is probability of finding walker on i-th node).
- We can write $\vec{v} = \sum_j a_j \vec{u}_j$ (sum over the basis vectors).
- Apply the state transition matrix to each eigenvector:

$$M\vec{u}_j = \lambda_j \vec{u}_j$$

- Applying it t-times:

$$MMM \cdots M\vec{u}_j = M^t \vec{u}_j = (\lambda_j)^t \vec{u}_j$$

Eigenvalues — what does it mean?

- Steady-state $\lambda_1 = 1$. The state transition matrix will have *at least* one eigenvalue $\lambda_1 = 1$.

$$M\vec{u}_1 = \lambda_1\vec{u}_1$$

equivalently:

$$M^t\vec{u}_1 = (\lambda_1)^t\vec{u}_1 = \vec{u}_1$$

- In fact the number of eigenvalues with $\lambda = 1$ equals the number of components! (Each component relaxes to its unique, independent steady-state solution).

Spectral gap

- When only one component, the largest eigenvalue $\lambda_1 = 1$. All other λ's are smaller.
- The difference $\lambda_1 - \lambda_2$ is the *spectral gap*.
- It tells us how effectively we can partition graphs into separate pieces. The larger the spectral gap, the worse the partitioning (cuts too many edges).
- Identify bottle necks.
- Mixing time! (related to λ_2).
- Isoperimetric number (the best cut), also related to λ_2.

Mixing time/relaxation time

Mixing time: Start a random walker at any arbitrary node. How long before it forgets, on average, where it started from?

Consider a "dumbbell" graph. Long memory of whether started on left or right hand side. (But very easy to partition in two w/o cutting many edges).

A closely related measure is the *relaxation time* . Time for an original amplitude to decay to $1/e$:

$$M^t \vec{v}_i = \frac{1}{e} \vec{v}_i.$$

Relaxation time

Want to calculate the relaxation time of the eigenvectors:

$$M^t \vec{u}_i = \tfrac{1}{e}\vec{u}_i = (\lambda_i)^t \vec{u}_i.$$

$$\implies t \ln(\lambda) = \ln(1/e) = -\ln(e) = -1.$$

$$\boxed{t = -1/\ln(\lambda)}$$

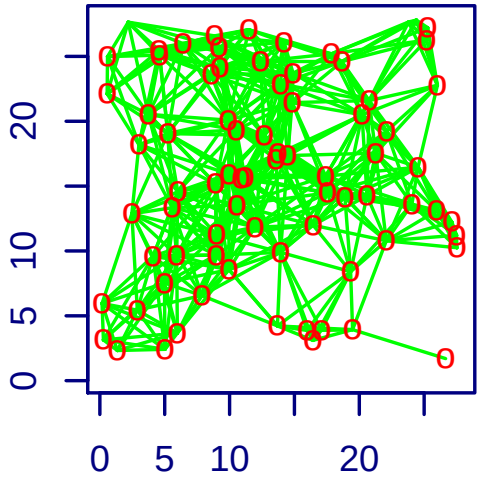
Thus, the largest relaxation time in the system:

$$\boxed{t_{max} = -1/\ln(\lambda_2)}$$

Example topologies

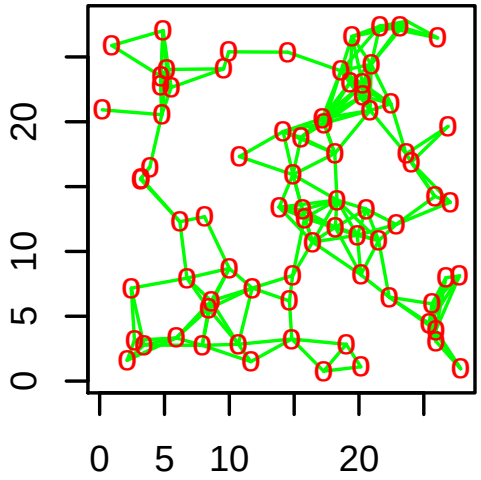
$\tau_{[min]}$ = 109 t_o

R = 7.513



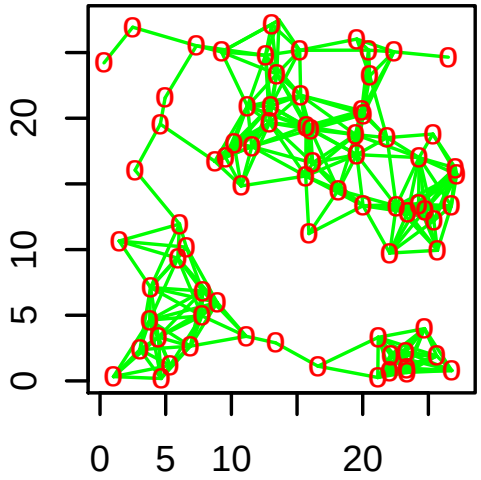
$\tau_{[med]}$ = 604 t_o

R = 4.61

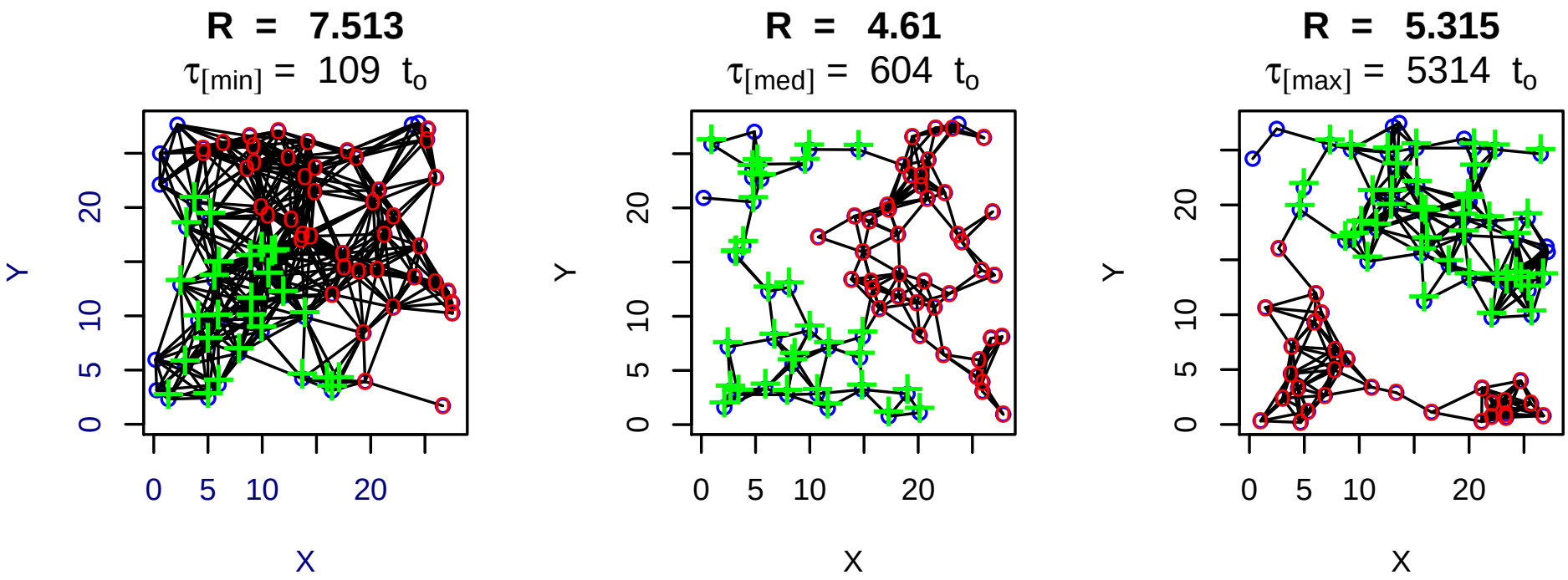


$\tau_{[max]}$ = 5314 t_o

R = 5.315



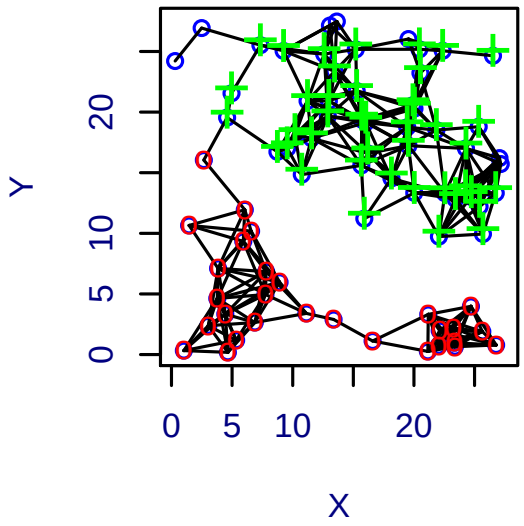
Mixing times and partitioning into two components



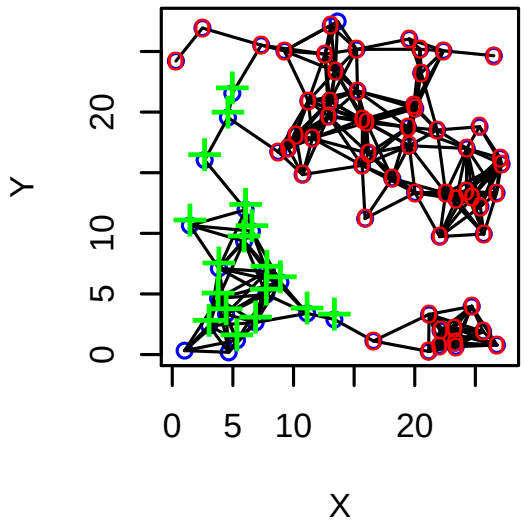
The larger the mixing time, the better the cut.

Higher order modes

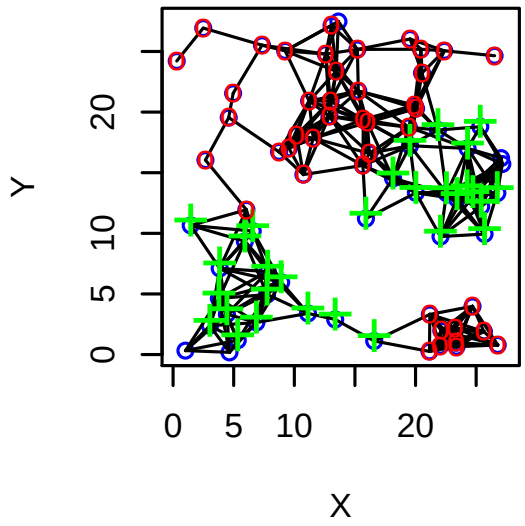
Timescale, τ_1 = 5314 t_o

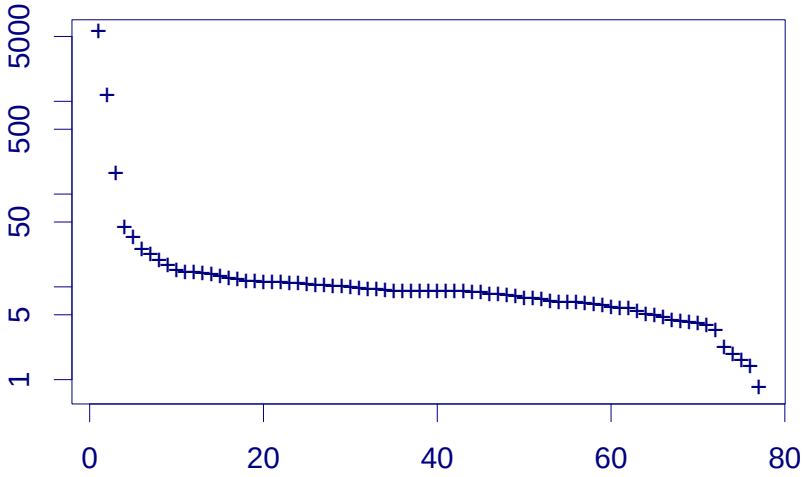


Timescale, τ_2 = 1092 t_o



Timescale, τ_3 = 157 t_o





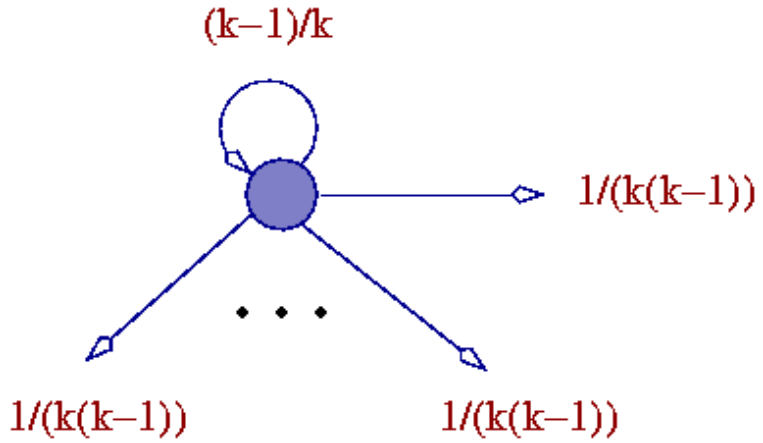
Applications to sensor networks

Standard state transition matrix (column normalize):

- Fastest mixing 1) complete graph, 2) star network.
- Smallest cover time: star network.

Applying this to wireless networks (next time):

- But in real-world often communication overhead (only one conversation at a time).
- Weighting the random walks:



Asymmetric (directed) graphs

- Can have complex (real + imaginary) eigenvalues.
- Not fully known how to use spectral methods for such graphs (an open problem).

Summary: spectral methods, eigen-spectrum

- If two distinct graphs have the same eigen-spectrum, they are likely isomorphic (esp for large graphs).
- Eigenvalues: degeneracy of $\lambda = 1$ tells us how many disconnected components in the graph.

Summary: spectral methods, measures

- Mixing time (time to forget where the walk started)
- Relaxation time (related to mixing time, gives bounds)
- Cover time (time to occupy each node)
- Spectral gap: the largest mixing time, $t_{\max} = -1/\ln(\lambda_2)$

  – the larger $t_{\max}$ the longer it takes for a random walk to cover the graph.

  – the larger $t_{\max}$ the more accurately a graph can be partitioned into two pieces.