Multiscale Analysis of Dynamic Graphs

Mauro Maggioni
Department of Mathematics and Computer Science
Duke University

ICIAM, Vancouver - 7/22/2011

Joint work: J. Lee (Duke, Stanford)

Partial support: DARPA, NSF, ONR, Sloan

Random walks on graphs and data

Given a weighted graph (G, E, W): vertices represent data points, edges connect x_i, x_j with weight $W_{ij} := W(x_i, x_j)$, when positive (or above a threshold). Let $D_{ii} = \sum_j W_{ij}$ and

$$\underbrace{P = D^{-1}W}_{\text{random walk}}, \quad \underbrace{T = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}}_{symm.\ \text{``}random\ walk''} \quad , \quad \underbrace{L = I - T}_{norm.Laplacian} \quad , \underbrace{H = e^{-tL}}_{Heat\ kernel}$$

Let $1 = \lambda_0 \geq \lambda_1 \geq \ldots$ and φ_i the eigenval.'s and eigenvec.'s of T. We shall consider the map $G \ni x \mapsto (\varphi_1(x), \ldots, \varphi_m(x)) \in \mathbb{R}^m$.

Random walks on graphs and data

Given a weighted graph (G, E, W): vertices represent data points, edges connect x_i, x_j with weight $W_{ij} := W(x_i, x_j)$, when positive (or above a threshold). Let $D_{ii} = \sum_j W_{ij}$ and

$$\underbrace{P = D^{-1}W}_{\text{random walk}}, \quad \underbrace{T = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}}_{symm.\ \text{“}random\ walk''} \ , \quad \underbrace{L = I - T}_{norm. Laplacian} \ , \quad \underbrace{H = e^{-tL}}_{Heat\ kernel}$$

Let $1 = \lambda_0 \geq \lambda_1 \geq \ldots$ and φ_i the eigenval.'s and eigenvec.'s of T. We shall consider the map $G \ni x \mapsto (\varphi_1(x), \ldots, \varphi_m(x)) \in \mathbb{R}^m$.

We may construct weighted graphs from data: given

. Data $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^D$.

. Local similarities via a kernel function $W(x_i, x_j) \geq 0$.

Simplest example: $W_\sigma(x_i, x_j) = e^{-||x_i - x_j||^2/\sigma}$.

Dynamic Graphs

J. Lee, MM, Thesis, 2010,
Proc. SampTA, 2011

Given: time series of graphs G_t. Objective: to analyze this time series. Desiderata:

. Sensitive to large and small significative changes in the network, and to their location.

. Should capture both topological and quantitative geometric changes.

. Should yield measures of change: want to do analysis, statistics...

. Robust to "noisy" perturbations of the network.

We are used to performing similar tasks on real-valued time series, how do we extend similar tools to a "space of graphs"? How can we quantify change? How do we construct multiple resolutions on graphs?

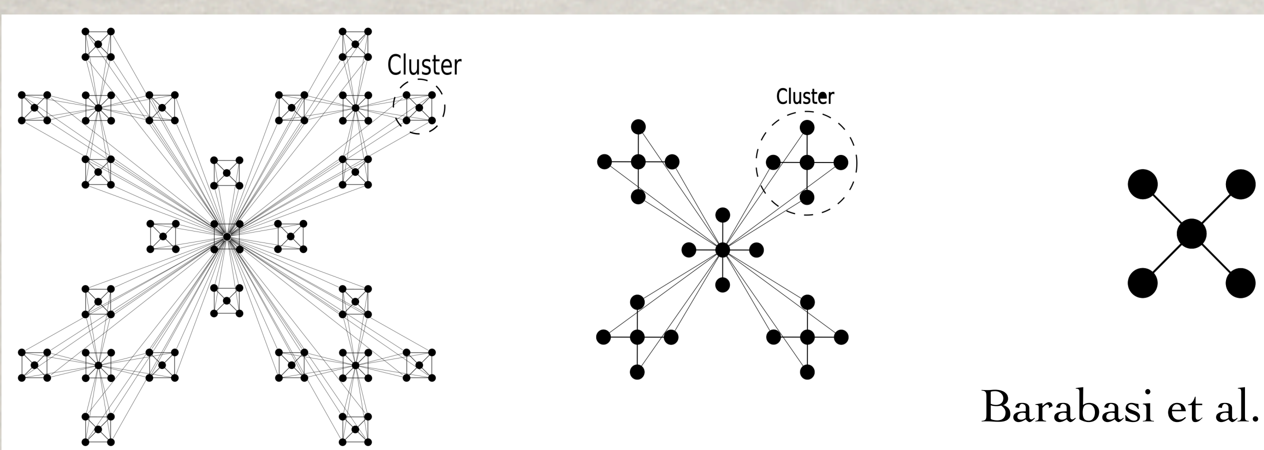
Multiscale Graphs - Toy Model

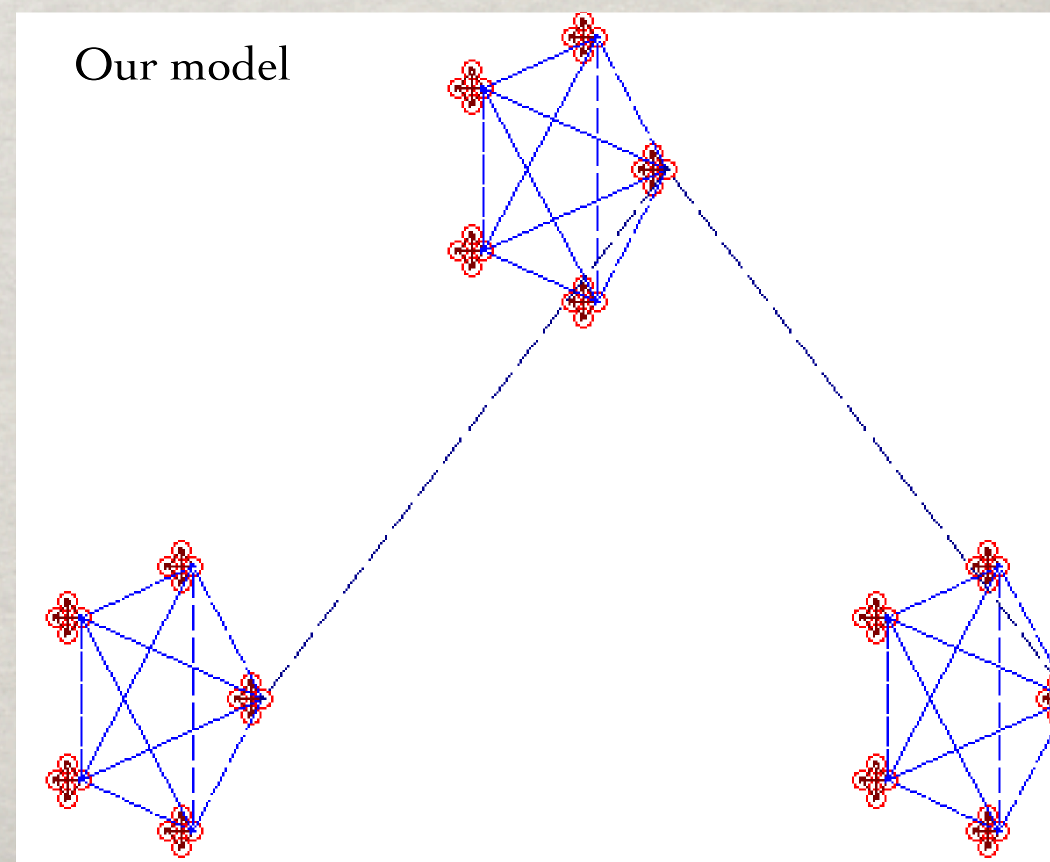
The graphs we start with will have a strong multiscale structure. They are constructed as follows: we start with a dictionary $\mathcal{D}$ of graphs, including for example complete graphs, trees, paths, loops, etc... Then:

(i) Pick J graph flavors $F_1, \ldots, F_J$, from $\mathcal{D}$, i.e. on flavor per scale

(ii) Let the graph at the coarsest scale be of flavor G_1, with edges with weight $O(2^1)$

(iii) For $j = 2, \ldots, J$, replace each vertex of G_{j-1} with a graph G_j of flavor F_j and edges of weight $O(2^j)$.



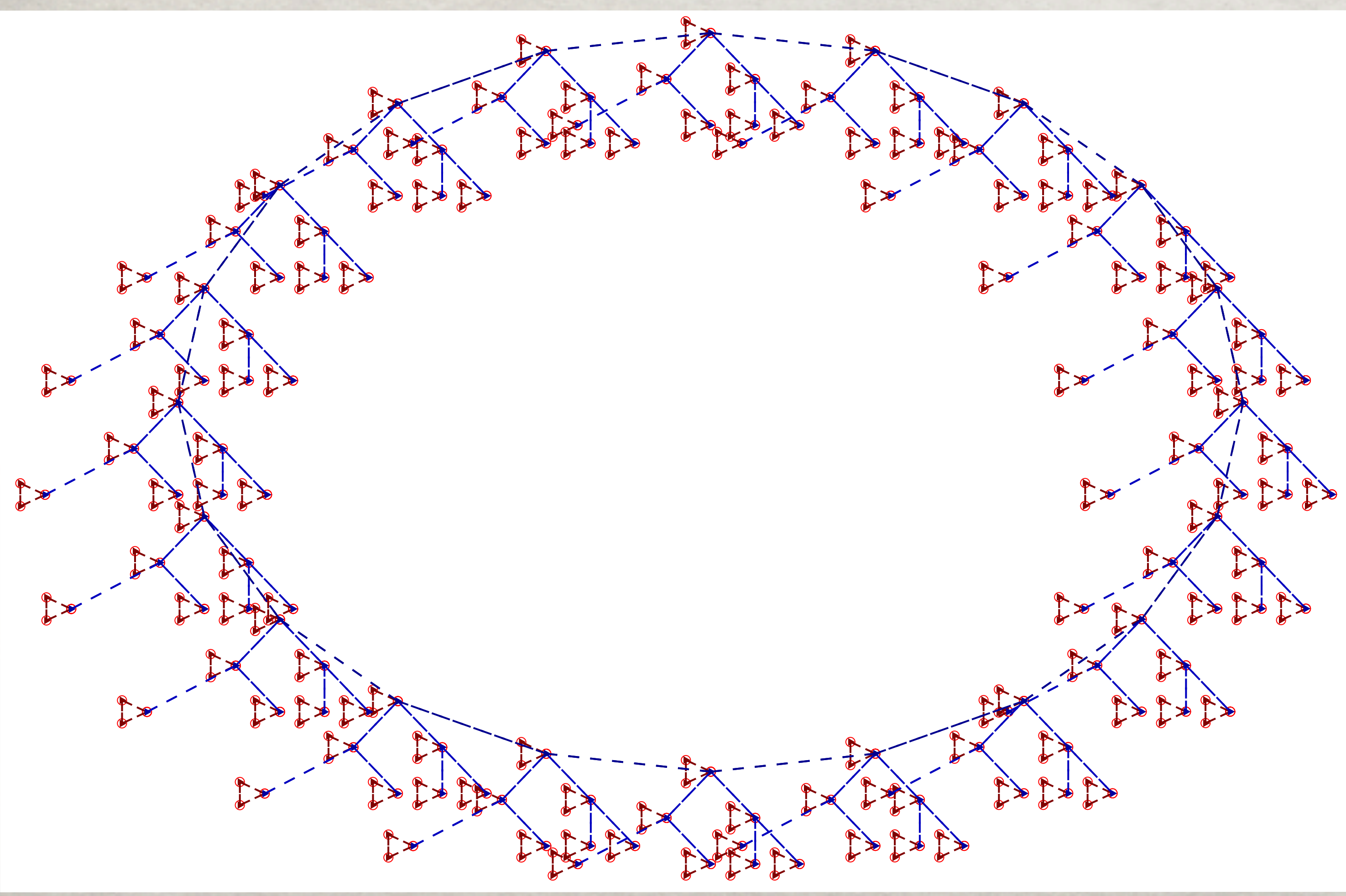


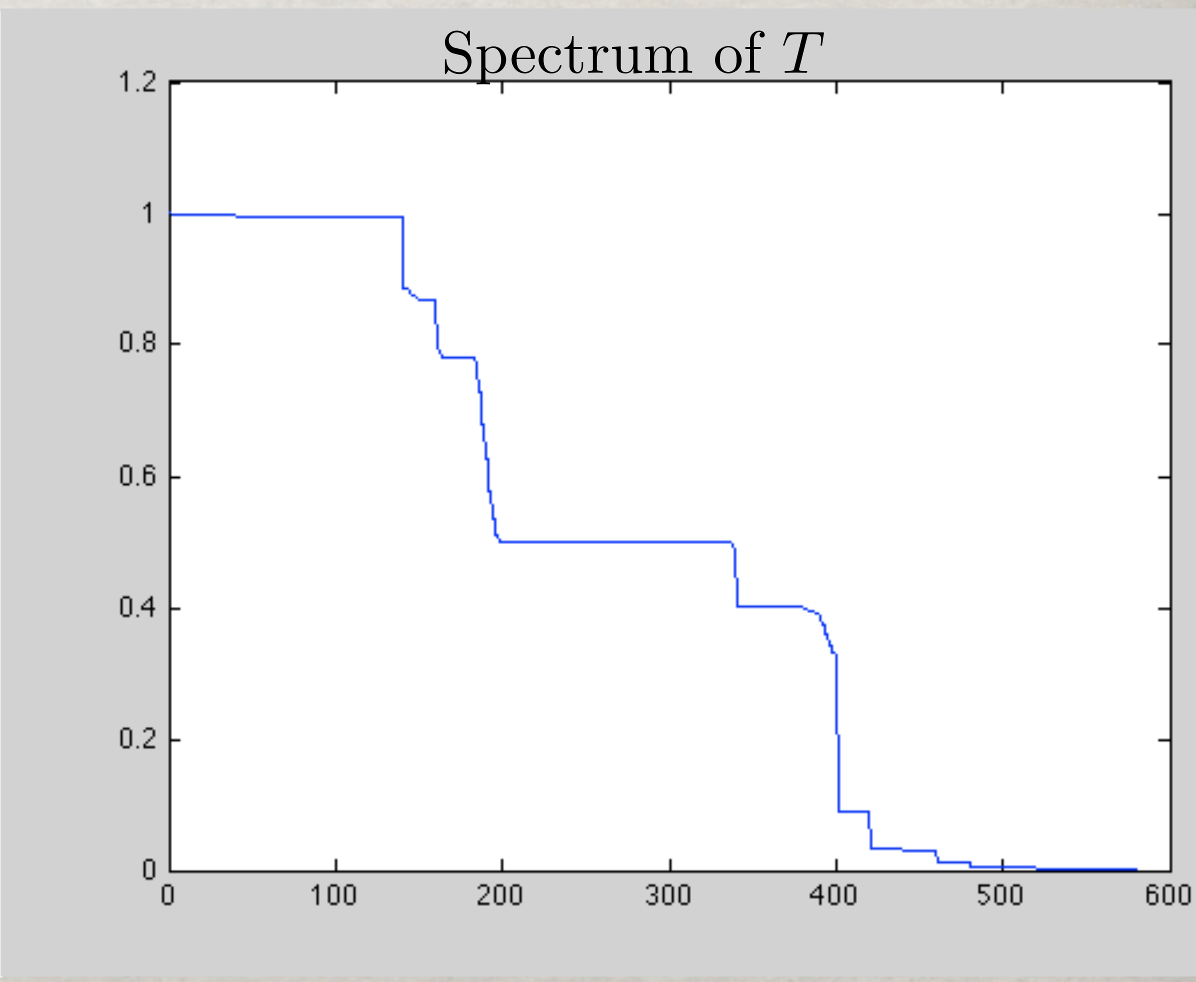
Multiscale Graphs - Toy Model

The generalizations below are *trivially* handled by the algorithms that follow, that do not depend on the symmetries above. For example different models used at different locations at different scales.

What matters is that the random walk has multiple time scales.

Intuitively: there is clear definition of scales, and what it means to “look at the graph at a certain scale”.





Dynamic Graphs - Toy Model

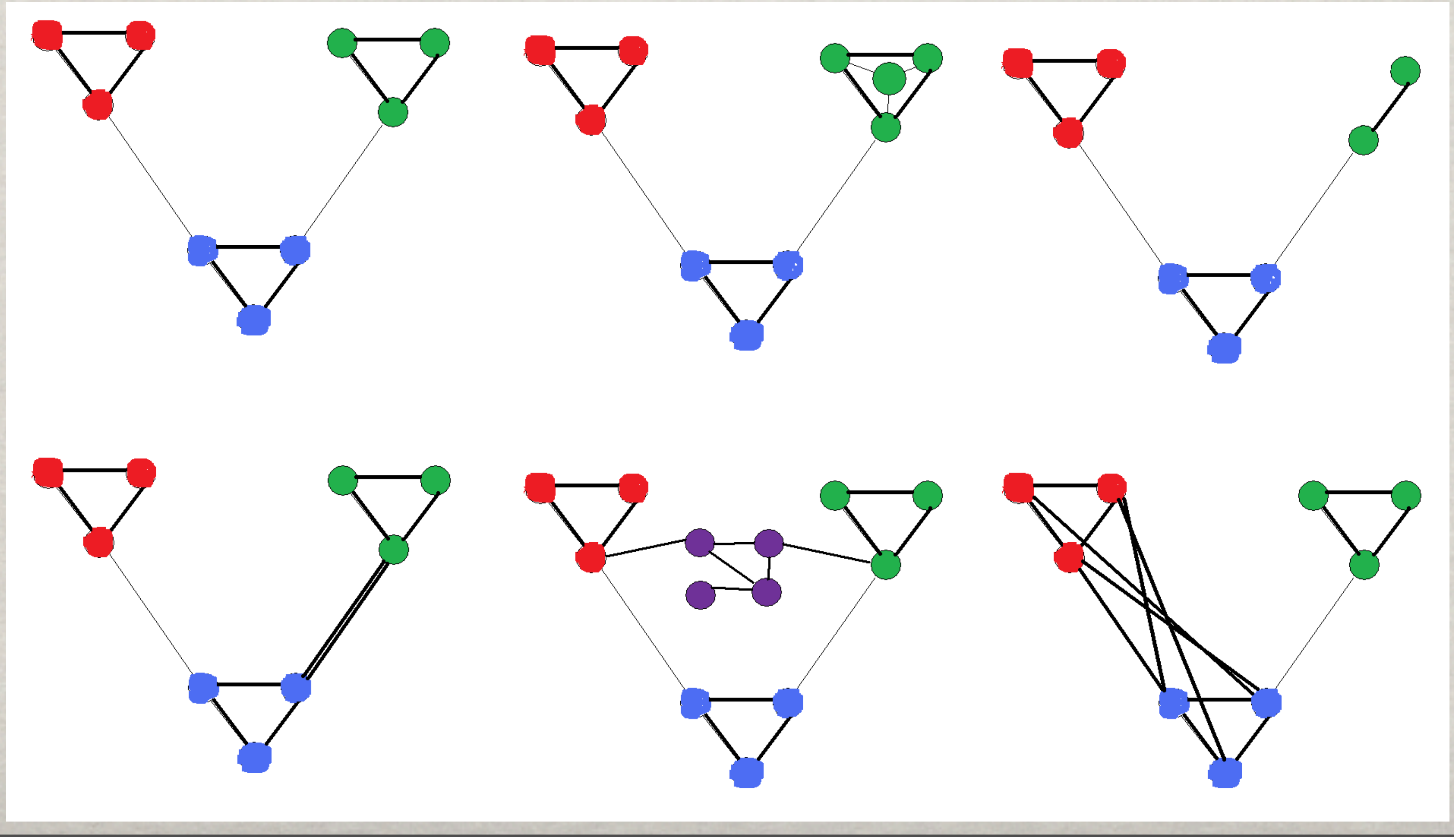
We consider a small set of "elementary moves", including:

(i) adding or removing a vertex

(ii) adding or removing an edge

(iii) changing an edge weight

The scale of the change is important.



Multiscale analysis of Dynamic Graphs

Our construction consists of 3 steps:

- Construct multiscale partitions of G_t
- Map G_t to a feature vector whose coordinates are operators, more specifically versions of P^t compressed on the multiscale partitions
- Compare G_{t+1} and G_t based on the feature vectors constructed above: these are now vectors of operators, and we know how to compare operators.

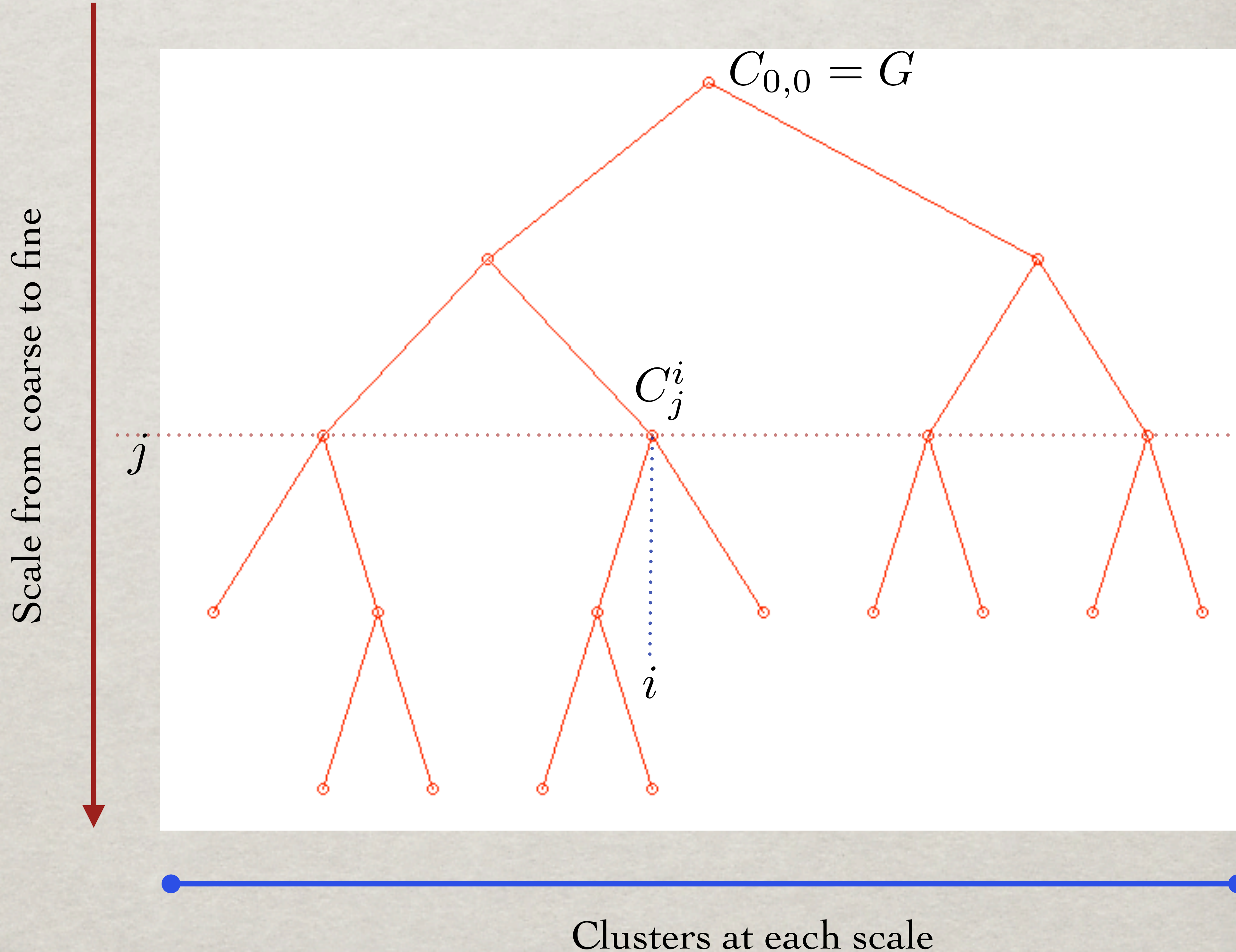
Algorithm: Multiscale Partitions

Recursive cuts lead to a multiscale partition

...

...

we one such at each time snapshot



Algorithm: Multiscale Partitions

Edges leaving the cluster

Total number of edges in cluster

Alternative: instead on N-cut use a conductance based criterion and local partitioning methods (D. Spielman, F. Chung, ...). This has stronger theoretical guarantees, and a good option for very large graphs.

Algorithm: Multiscale Partitions

Algorithm for constructing partitions. What is a good partition? Ncut should be small:

Edges leaving the cluster

$$Ncut(C^1,\dots,C^k) = \frac{1}{2}\sum_{i=1}^{k}\frac{W(C^i,(C^i)^c)}{vol(C^i)}$$

Total number of edges in cluster

where $W(A,B) = \sum_{x\in A, y\in B} W(x,y)$ and $vol(A) = \sum_{x\in A}\sum_y W(x,y)$.
NP-Hard to minimize, approximation algorithms exist [Arora]. Oftentimes relaxed to spectral clustering on $P = D^{-1}W$: compute top $k+1$ eigenvectors of P: $\varphi_1,\dots,\varphi_{k+1}$, embed the graph in Euclidean space by $x \mapsto (\varphi_i(x))_{i=1}^{k+1}$, and perform k-means in this space.

At any time t, we perform the above recursively, with $k=2$, obtaining a multiscale family of partitions $\{C_{j,t}\}$ organized in a binary tree.

Alternative: instead on N-cut use a conductance based criterion and local partitioning methods (D. Spielman, F. Chung, ...). This has stronger theoretical guarantees, and a good option for very large graphs.

Algorithm: Multiscale Dynamics

For every t and every scale j, compress P_t^τ: it is a matrix of size $|C_{j,t}| \times |C_{j,t}|$:

τ-step r.w. on G_t compressed at scale j

$$(P_{t,j,\tau})_{il} := \sum_{x \in C^i_{j,t}} \frac{1}{C^i_{j,t}} \sum_{y \in C^l_{j,t}} P^\tau(x,y)$$

probability mass transported from a starting point in $C^i_{j,t}$ to some point in $C^l_{j,t}$

t indexes time

j indexes scale in the partition tree

τ refers to internal averaging time

i, l index clusters at scale j

Algorithm: Multiscale compression

For every t and every scale j, compress P_t^τ: it is a matrix of size $|C_{j,t}| \times |C_{j,t}|$:

$$(P_{t,j,\tau})_{il} = \sum_{x \in C_{j,t}^i} \frac{1}{C_{j,t}^i} \sum_{y \in C_{j,t}^l} P^\tau(x,y)$$

for i, l varying in the index of the sets in the partition $C_{j,t}$.

In fact we can do much more, since we can localize this changes, at every scale, and across scales, and develop a full multiscale ("wavelet-like") time series analysis for graphs.

Algorithm: Multiscale compression

For every t and every scale j, compress P_t^τ: it is a matrix of size $|C_{j,t}| \times |C_{j,t}|$:

$$(P_{t,j,\tau})_{il} = \sum_{x \in C_{j,t}^i} \frac{1}{C_{j,t}^i} \sum_{y \in C_{j,t}^l} P^\tau(x,y)$$

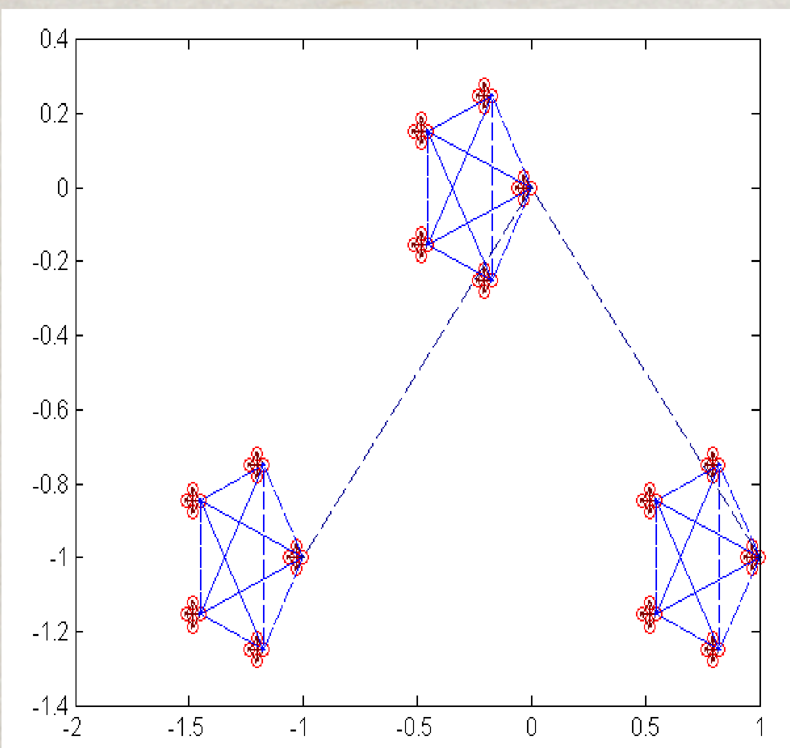
for i, l varying in the index of the sets in the partition $C_{j,t}$.

At this point we can measure variations in $P_{t,j,\tau}$ as a function of t, for the scale j and "internal time scale" τ, for example by measuring $||P_{t,j,\tau} - P_{t+1,j,\tau}||_F$.

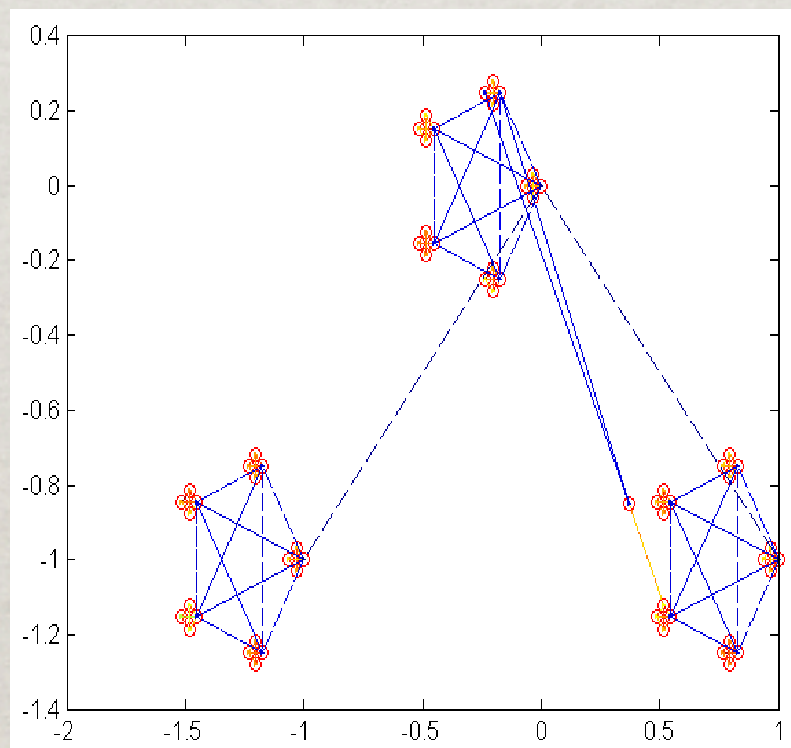
In fact we can do much more, since we can localize this changes, at every scale, and across scales, and develop a full multiscale ("wavelet-like") time series analysis for graphs.

A Simple Example

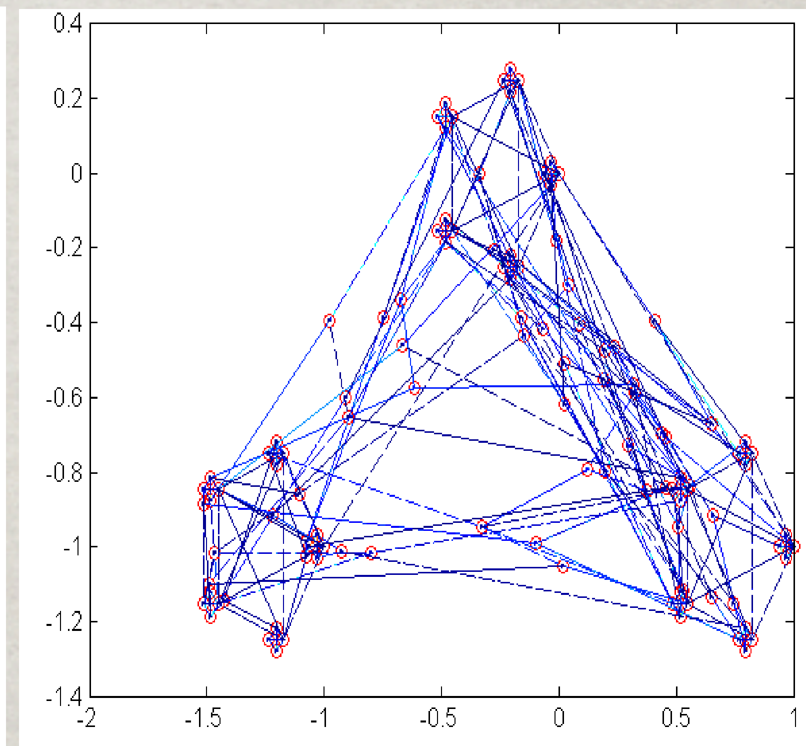
A time series of 51 graphs was constructed with a single vertex and edge noise added after each graph. The initial graph has 3 layers, and 60 nodes. The edge weights on the scales from finest to coarsest are 64, 8, and 1. The initial partitions are fed to the algorithm and we update the partitions using the aforementioned online algorithm. At each step, a new vertex with 3 edges is introduced.



$t = 0$

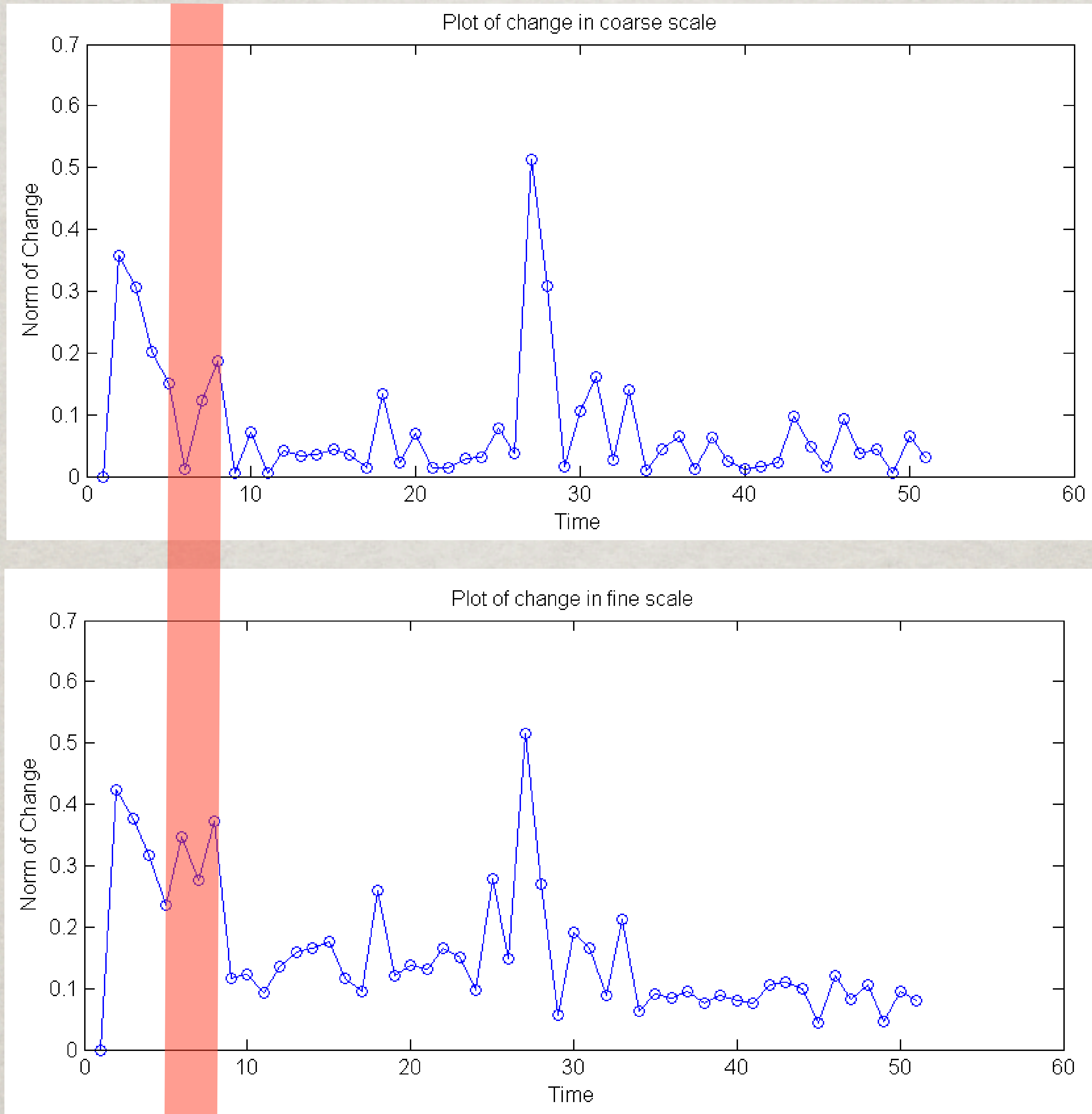


$t = 1$

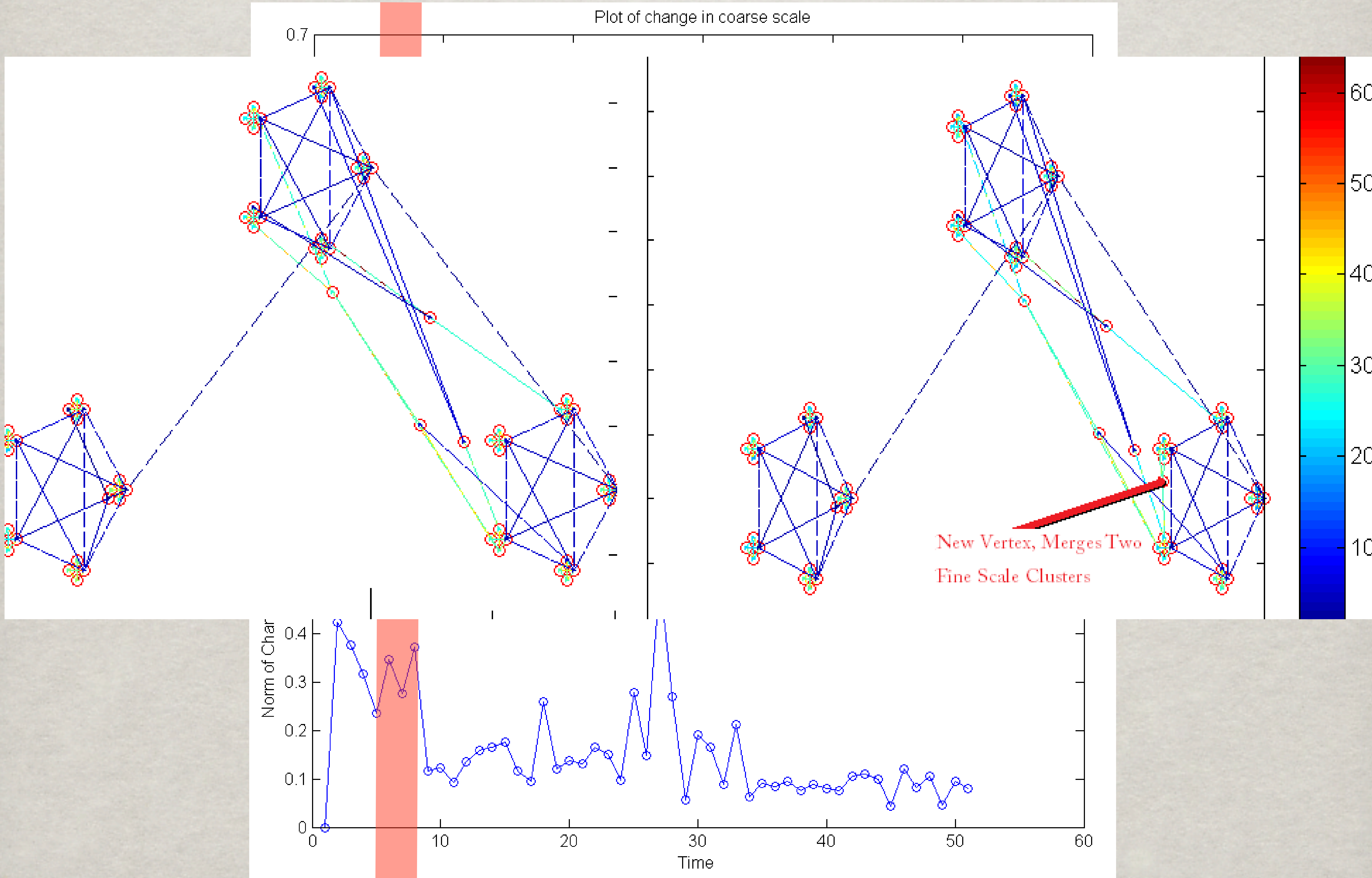


$t = 51$

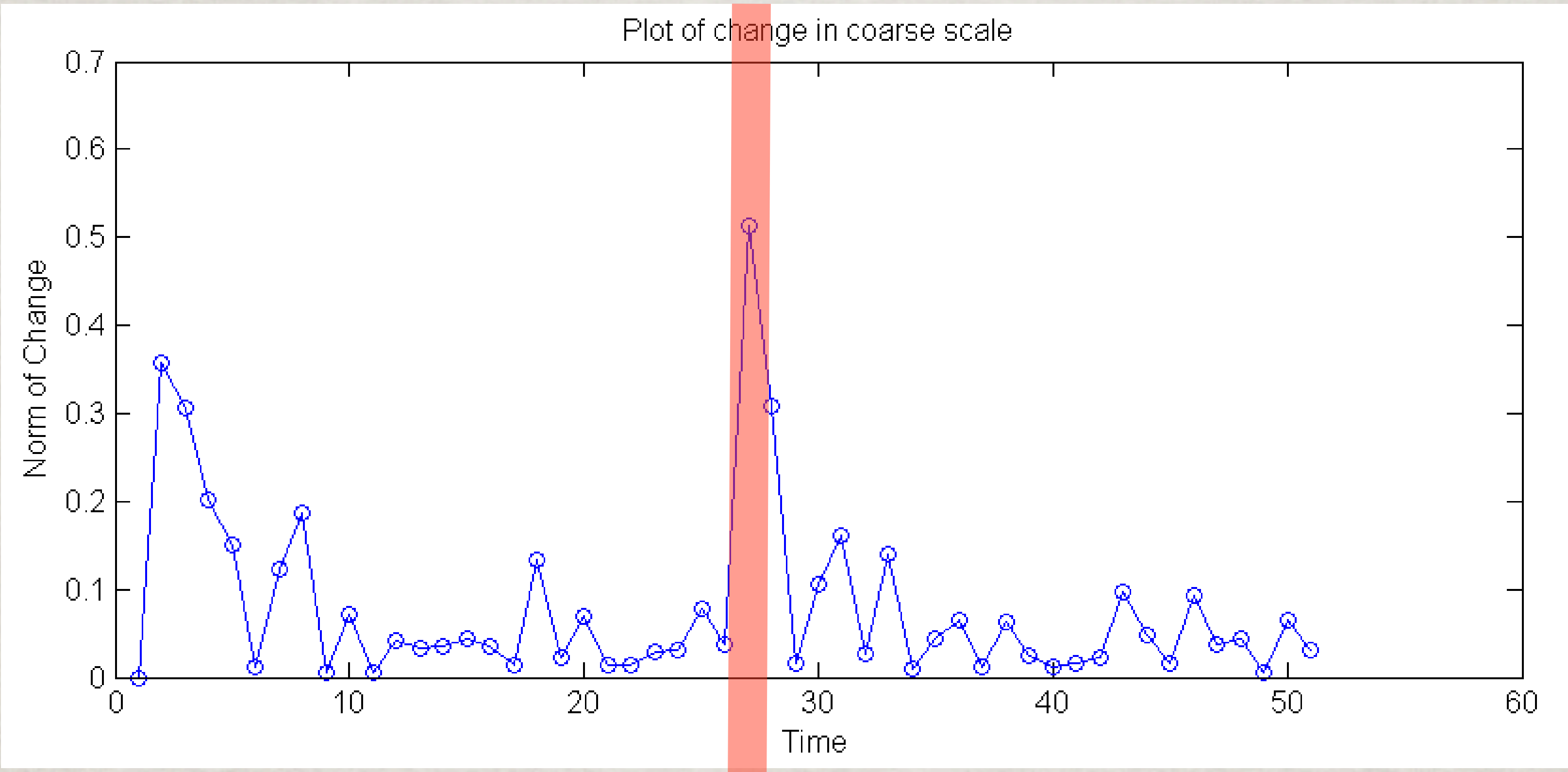
A Simple Example

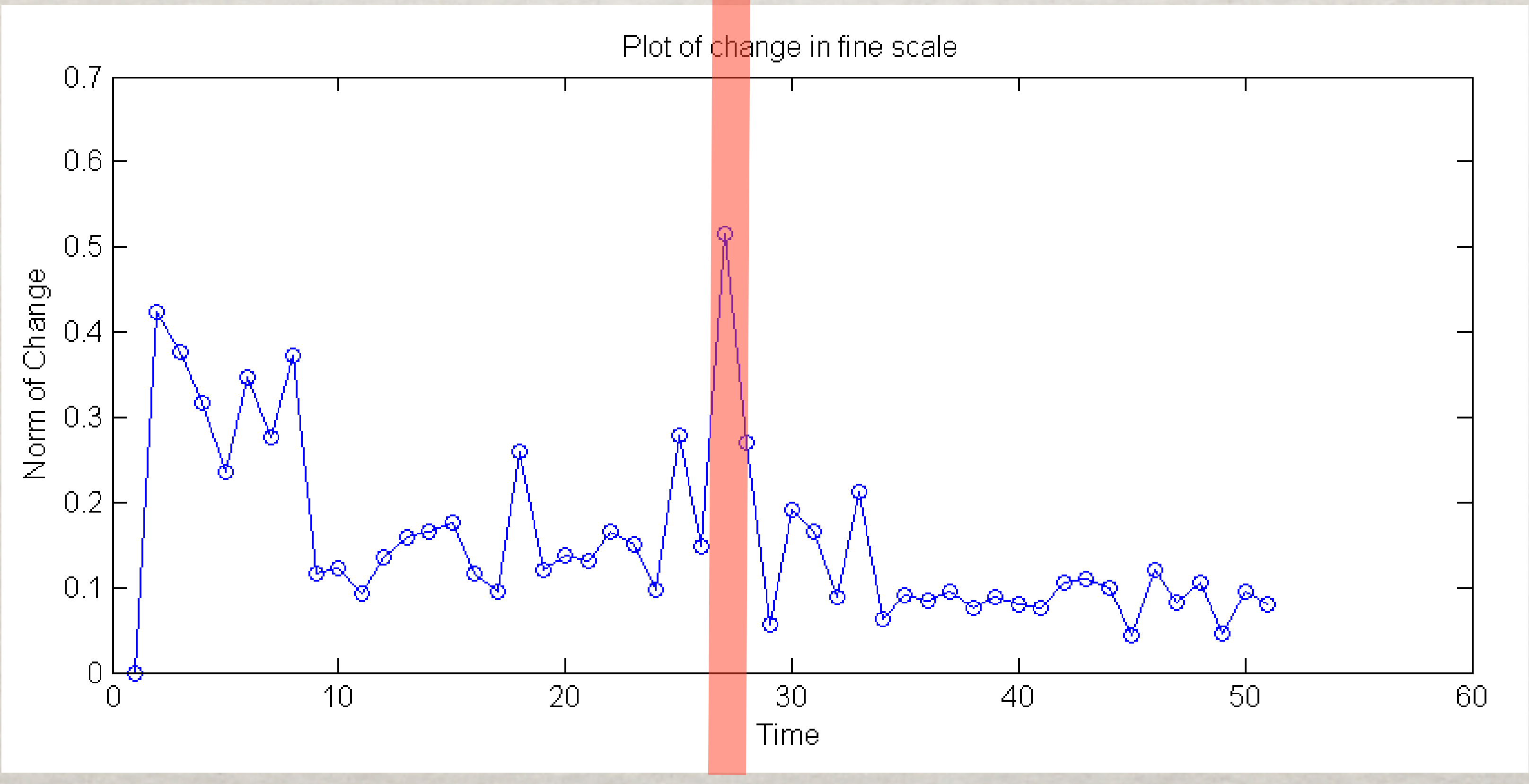


A Simple Example

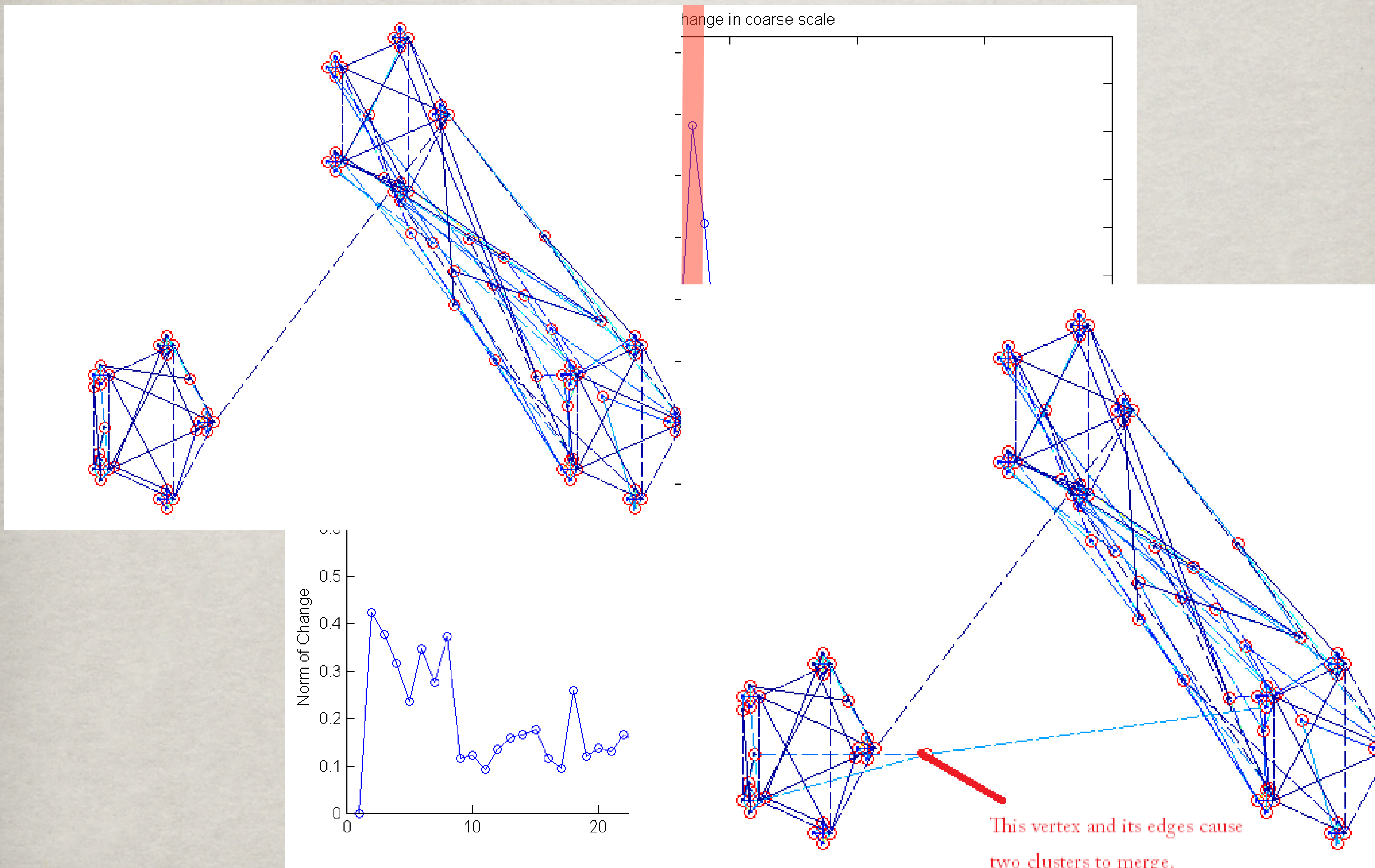


A Simple Example



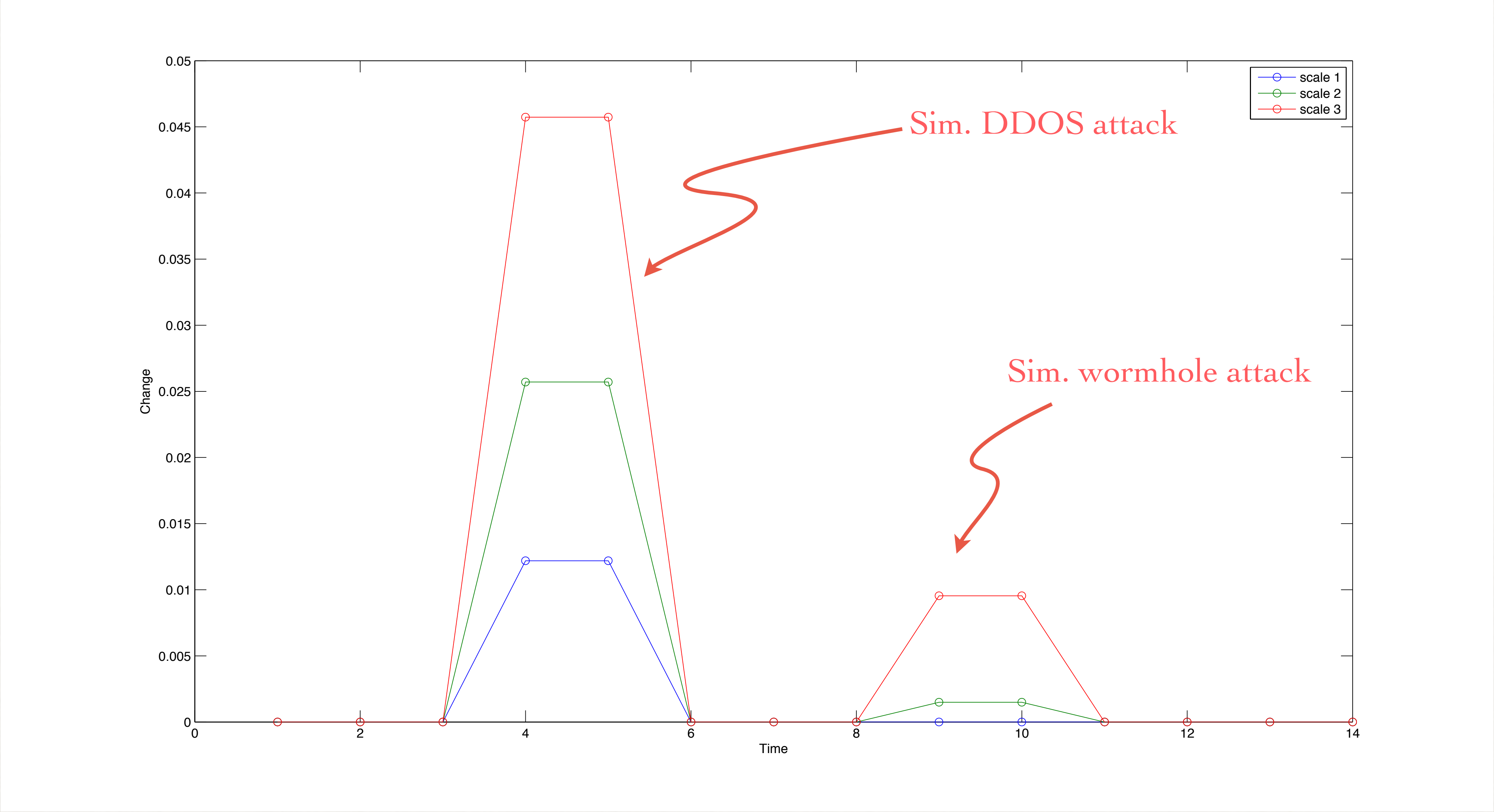


A Simple Example



Simulated "attacks"

Network of political blogs, 1400 nodes and 19000 edges. We simulate two attacks: a DDOS attack at time 4, when one random vertex is connected to 100 random vertices, till time 6, and then a wormhole attack at time 8, when the two farthest vertices are connected by a heavy edge.



L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US Election", in Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem (2005)

A much finer version

Joint with P. Balachandran and J. Mattingly

- We consider continuous time, and one scale only. Let $C_1, \ldots, C_k$ be clusters, constructed according to a conductance criterion, so that a r.w. started in C_i almost mixes inside C_i before significant mass leaks to C_i^c.

- *Compressing the clusters.* Let $\partial^- C_i$ and $\partial^+ C_i$ be the entry and exit points of C_i: we encode the probability of first leaving C_i through $y \in \partial C_i^+$ having entered at $x \in \partial C_i^-$, and approximate (in TV-distance) the distribution of the corresponding first exit time $\tau_{y|x}$.

- *Homogenized process.* It state space is $\partial^- C_i \cup \partial^+ C_i$, and the transitions are as follows: at $x \in \partial^- C_i$ toss a coin to pick be the exit point $y \in \partial^+ C_i$, and drawing a corresponding first exit time $\tau_{y|x}$. Then use the edges and transition times on the original graphs to move from $y \in \partial^+ C_i$ to $y \in \partial^- C_l$.

- *Some properties.* The homogenized process is semi-Markov (Markov at jump times), with a time-dependent transition kernel $\overline{P}_t(x, y)$. Then in a suitable, strong sense (TV distance), $\overline{P}_t(x, y)$ is close to $P_t(x, y)$.

- The above may be implemented by efficient algorithms.

A few notes

- Processes on a graph are the focus, not the graph itself
- We replaced a graph by a process on it (e.g. the random walk), and then studied that process at multiple scales. If the graph changes in time, we have a family of processes.
- Eigenfunctions capture large scale/long time dynamics, but expect interesting graphs to have rich multiscale spatial/temporal scales.
- Functions on graphs are also important, e.g. in machine learning.
- Spectral graph theory, multiscale analysis, stochastic processes all play fundamental roles.
- The construction of the processes from data is domain dependent, informed by physical/statistical/geometric models.
- We know how to construct multiscale basis functions (with fast transforms) on graphs; these may be used to compress processes such as the r.w.

Open problems & future dir.'s

Too many!

- Efficient and good local clustering algorithms.
- Properties of eigenfunctions of Laplacian/r.w. on graphs.
- Notions of (multiscale?) geometric stability and its relationships with stability of eigenfunctions, clusters, r.w.'s...
- How to study more general processes where diffusion is not the main component?

Collaborators: D. Brady (EE, Duke), R. Brady (CS, Duke), C. Clementi (Che, Rice), J. Mattingly (Math, Duke), E. Monson (CS, Duke), S. Mukherjee (Stats, Duke), R. Rajae (EE, Oregon), M. Rohrdanz (Che, Rice), R. Schul (Math, Stony Brook), W. Willinger (AT&T)

Partial support: DARPA, DOE, NSF, ONR, Sloan Foundation

THANK YOU!

www.math.duke.edu/~mauro