Models and Algorithms for Time-Dependent Networks

Des Higham
Department of Mathematics and Statistics
University of Strathclyde

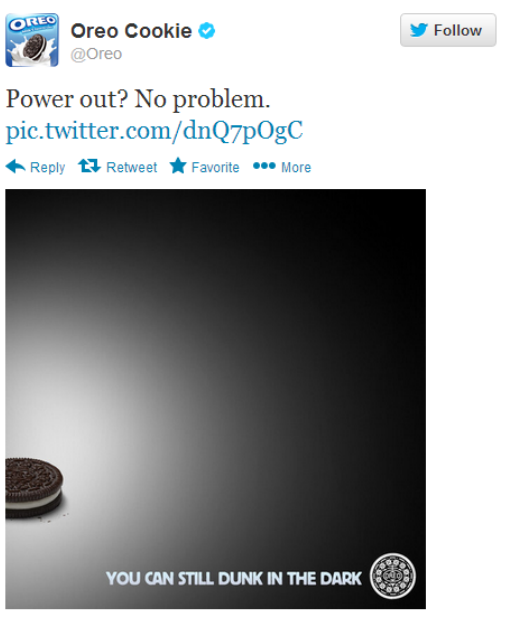
Joint work with
Fiona Ainley (Bloom), **Ernesto Estrada** (Strathclyde), **Peter Grindrod** (Oxford), **Peter Laflin** (Bloom), **Alex Mantzaris** (Strathclyde/Bloom), **Amanda Otley** (Bloom), **Mark Parsons** (Reading)

University of Strathclyde

Part 1, **Algorithms**

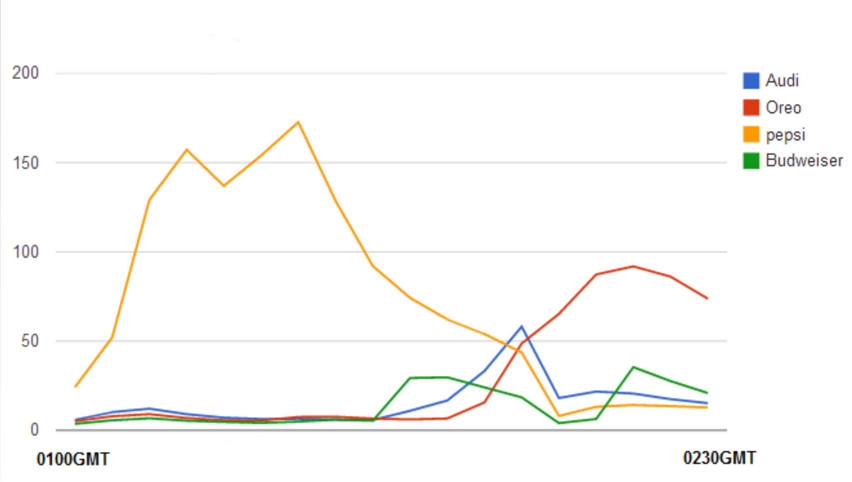
From Twitter’s homepage:
Businesses can also use Twitter to listen and gather market intelligence and insights

Oreo at Superbowl 2013



Audi did this slightly earlier

Influence over time



Applications

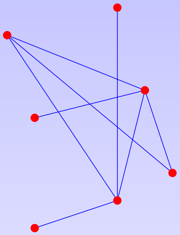
Large-scale, **dynamic** interaction data arises in, e.g.,

- **social media**
- **on-line business**
- **text, email, voicemail**
- **epidemiology**
- **neuroscience**

Some Algorithmic Challenges

- identify **communities**
- discover other types of **structure**
- **categorize** the roles of individuals
- **target** key players

Matrix Computation for Centrality



Unweighted, with N nodes
Adjacency matrix A

$\left(A^2\right)_{ij} \; := \; \sum_{p=1}^{N} a_{ip} a_{pj}$ counts **paths of length two** from node i to node j

Generally, $\left(A^k\right)_{ij}$ counts the number of **walks of length k** from node i to node j. Then

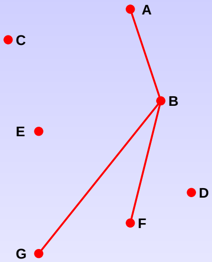
$\left(A + A^2/(2!) + A^3/(3!) + \cdots\right)_{ij}$ leads to $\left(e^A\right)$ and

$\left(A + \alpha^2 A^2 + \alpha^3 A^3 + \cdots\right)_{ij}$ leads to $(I - \alpha A)^{-1}$

to quantify how well **information can flow** from i to j

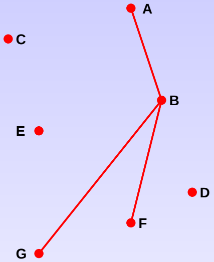
Time Ordered Sequence of Networks

Time 1 Time 2 Time 3

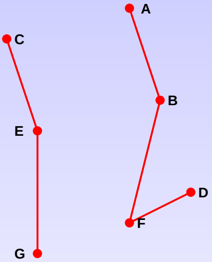


Time Ordered Sequence of Networks

Time 1

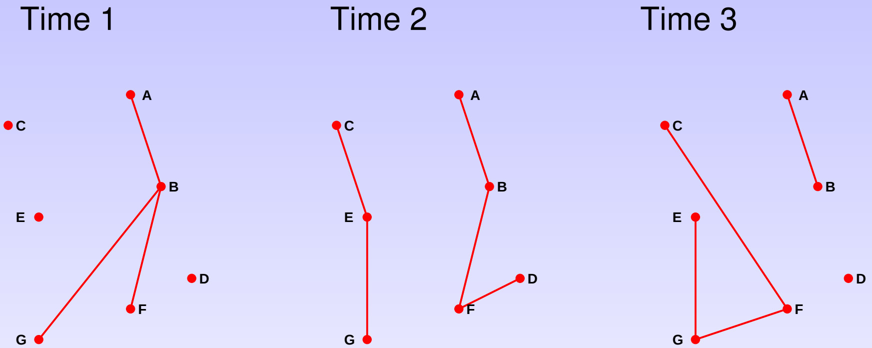


Time 2

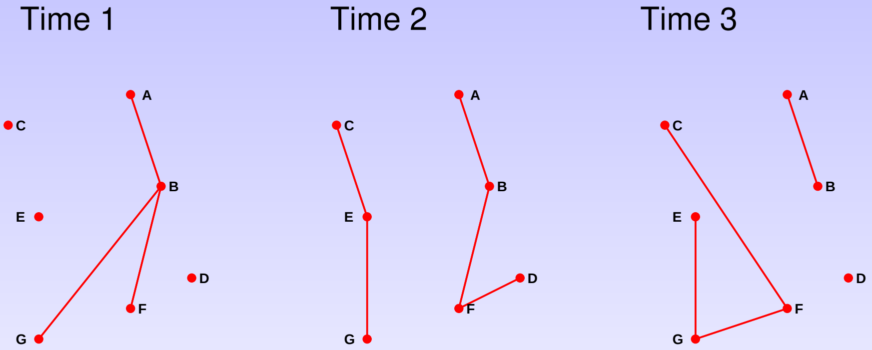


Time 3

Time Ordered Sequence of Networks

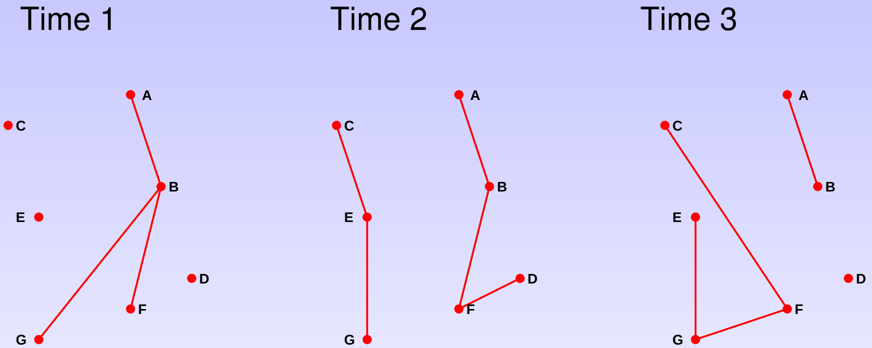


Time Ordered Sequence of Networks



- Lack of symmetry caused by time's arrow

Time Ordered Sequence of Networks



- Lack of symmetry caused by time's arrow
- Aggregation would also overestimate the spread

Dynamic Walks

Time points $t_0 < t_1 < t_2 < \cdots < t_M$

Adjacency matrices $A^{[0]}, A^{[1]}, A^{[2]}, \ldots, A^{[M]}$

Dynamic walk of length w from node i_1 to node i_{w+1}:
sequence of times $t_{r_1} \le t_{r_2} \le \cdots \le t_{r_w}$ and a
sequence of edges $i_1 \leftrightarrow i_2, i_2 \leftrightarrow i_3, \ldots, i_w \leftrightarrow i_{w+1}$,
such that $i_m \leftrightarrow i_{m+1}$ exists at time t_{r_m}

(Several variations are possible)

Use this to define **centrality** of a node,
generalizing Katz (1953)

New Algorithm

Grindrod, Higham, Parsons & Estrada, **Phys. Rev. E**, 2011

Key observation: the matrix product

$$A^{[r_1]} A^{[r_2]} \dots A^{[r_w]}$$

has i, j element that counts the number of dynamic walks of length w from node i to node j, where the mth step takes place at time t_{r_m}

Keep track of all such walks and discount by α^w

E.g. $\alpha^2 A^{[0]} A^{[1]}$, $\alpha^4 A^{[0]} A^{[2]} A^{[3]} A^{[7]}$, $\alpha^3 A^{[3]} A^{[3]} A^{[9]}$

This is achieved by

$$\mathcal{Q} := \left(I - \alpha A^{[0]}\right)^{-1} \left(I - \alpha A^{[1]}\right)^{-1} \cdots \left(I - \alpha A^{[M]}\right)^{-1}$$

Then $\mathcal{Q}_{ij}$ is our overall summary of how well information can be passed from node i to node j

Dynamic Centrality

We will call the row and column sums

$$\sum_{k=1}^{N} \mathcal{Q}_{nk} \qquad \& \qquad \sum_{k=1}^{N} \mathcal{Q}_{kn}$$

the **broadcast** and **receive** communicabilities

- generalizes **Katz centrality** in social networks
- involves **sparse linear solves**
- captures the **asymmetry** through non-commutativity of matrix multiplication

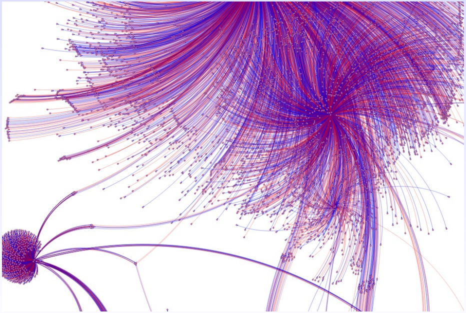
Twitter's Big Hitters

Laflin, Mantzaris, Grindrod, Ainley, Otley, Higham,
Social Network Analysis and Mining, 2013

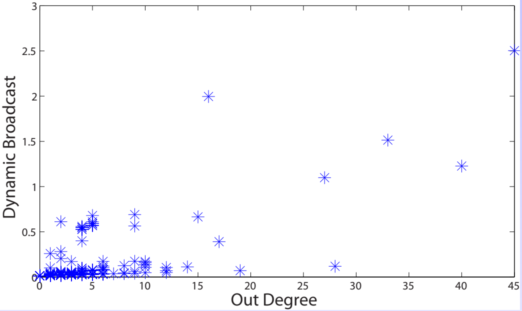
Listen to **tweets** containing the phrases
`city break`, `cheap holiday`, `travel`, `insurance`, `cheap flight` plus two brand names

From 17 June 2012 at 14:41 to 18 June at 12:41

0.5 Million Tweeters/Followers



Dynamic Broadcast Centralities



Twitter account with **fourth highest out degree** is a **very poor dynamic broadcaster**

Closer inspection $\Rightarrow$ an automated process

Five **social media experts** were given the Twitter data and asked to rank the accounts according to importance

We found that **dynamic centrality measures are hard to distinguish from human experts**

Downweighting over time

Grindrod & Higham, **SIAM Review** (Research Spotlights) 2013

Motivation: News goes stale, messages become irrelevant, viruses mutate, ... ***old information is less important***

The algorithm can be generalized naturally to

$$\mathcal{S}^{[k]} = \left(I + e^{-b\Delta t_k}\mathcal{S}^{[k-1]}\right)\left(I - \alpha\, A^{[k]}\right)^{-1} - I$$

Here, $\left(\mathcal{S}^{[k]}\right)_{ij}$ counts the number of dynamic walks from i to j up to time t_k, scaled by

- a factor α^w for **dynamic walks of length w**
- a factor e^{-bt} for **walks that begin t time units ago**

We have a new parameter, b:
$b = 0$ is the previous algorithm
$b = \infty$ is Katz on the current network

Continuous Time

Grindrod & Higham, **Proc. Roy. Soc. A**, 2014

Discretizing in time and binarizing is convenient, but

- Δt too large can **overlook** or **smear** events
- Δt too small may **introduce redundant computations** and give a **false impression** of accuracy

So use $A(t)$ over continuous time

Define communicability by taking $\Delta t \to 0$ limit

Key idea: use the scaling $(I - \alpha A(k\Delta t))^{-\Delta t}$

Justification: A constant $\Rightarrow$ correct behaviour

$$(I - \alpha A)^{-\frac{1}{2}\Delta t}(I - \alpha A)^{-\frac{1}{2}\Delta t} = (I - \alpha A)^{-\Delta t}$$

Continuous Time

$\mapsto$ Matrix ODE for the evolution of pairwise communicability

$$U'(t) = -b(U(t) - I) - U(t) \log \left(I - \alpha A(t) \right)$$

Here:

- $U(t)_{ij}$ is the communicability between nodes i and j
- log is the **matrix logarithm**

IEEE VAST 2008 Challenge (ficticious data)

Voice calls from a controversial socio-political movement

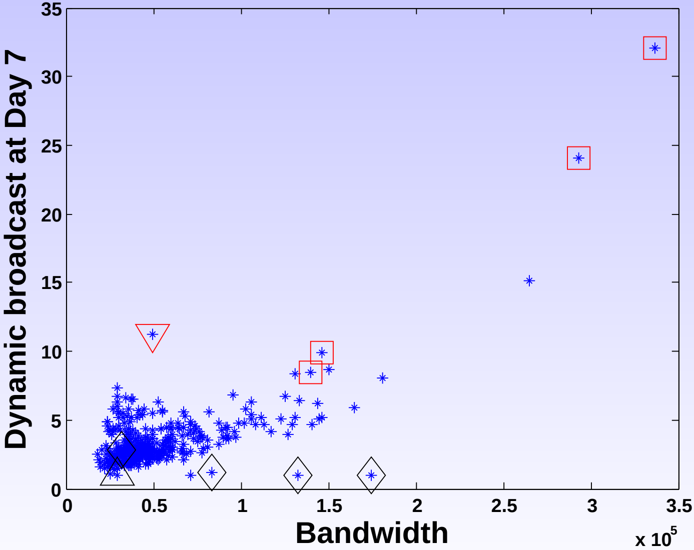
- 400 IDs
- 10 days
- each call: time stamp (hour/minute) plus duration (sec)
- a key inner circle exists with a single ringleader

Based on published challenge entries

- inner circle of five and ringleader have been identified
- these five people change IDs at end of day 6

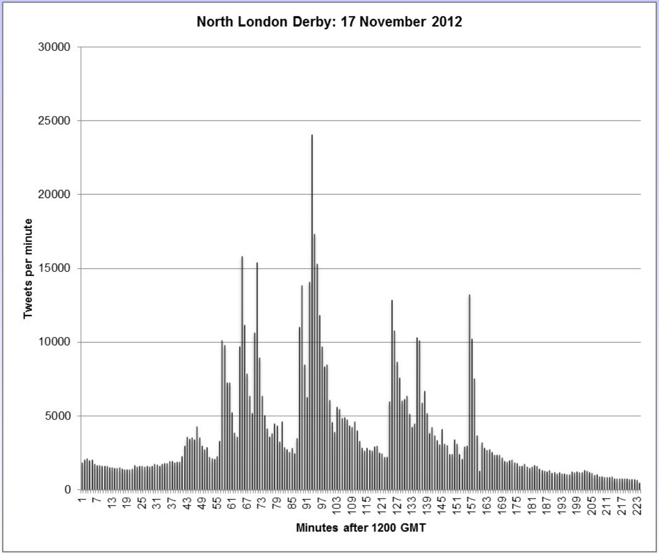
We take $A(t)$ to be symmetric and use MATLAB's `ode23`

Unsupervised Analysis: up to Day 7

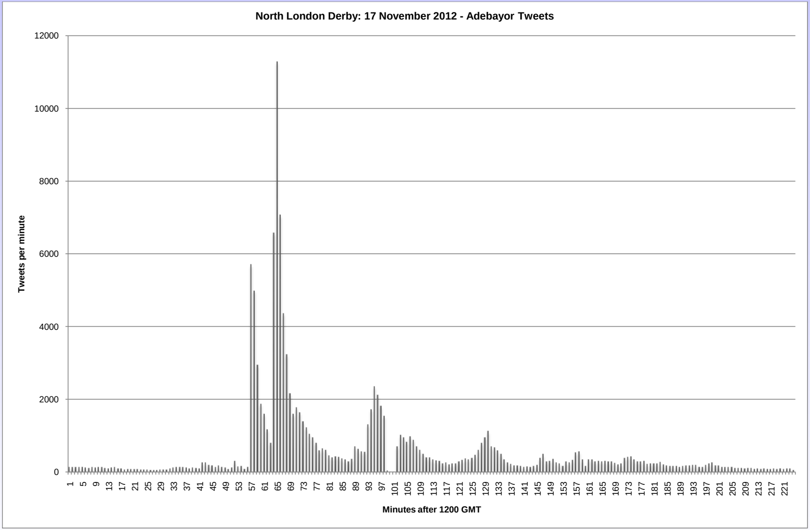


Arsenal 5 - 2 Spurs, Nov. 2012, kick off 12:45

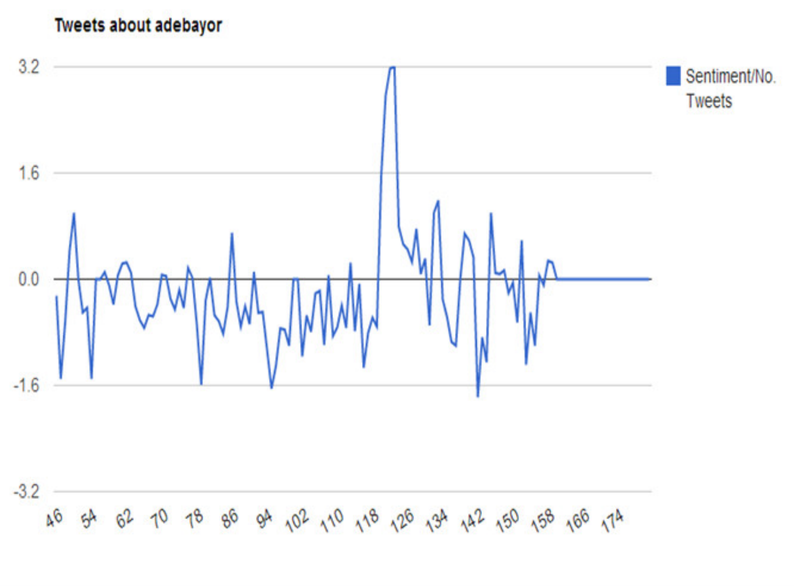
850,000 tweets in 4 hours. Up to 24,000 tweets per minute



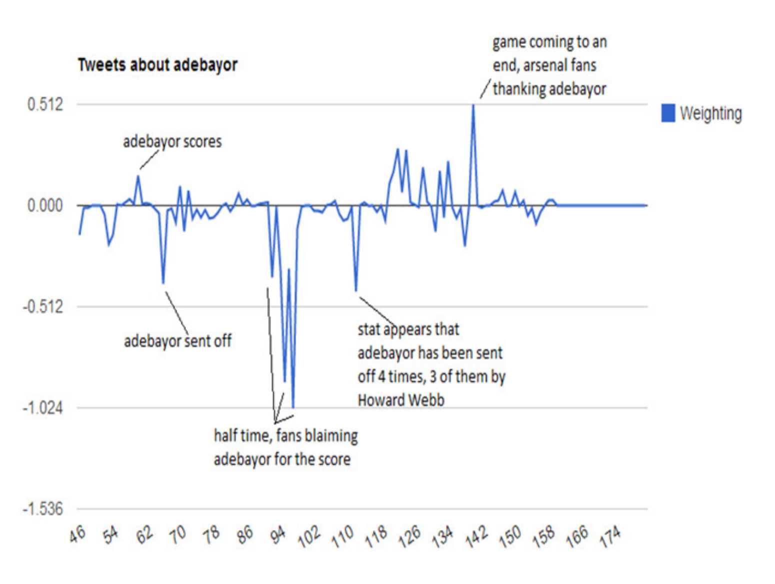
Adebayor: volume of tweets



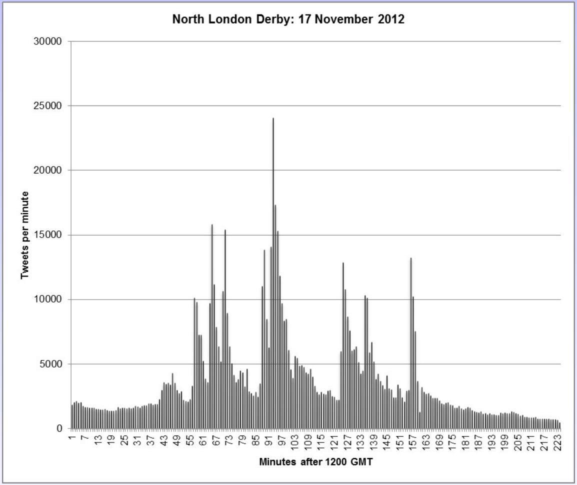
Adebayor: sentiment across time



Adebayor: sentiment weighted by influence

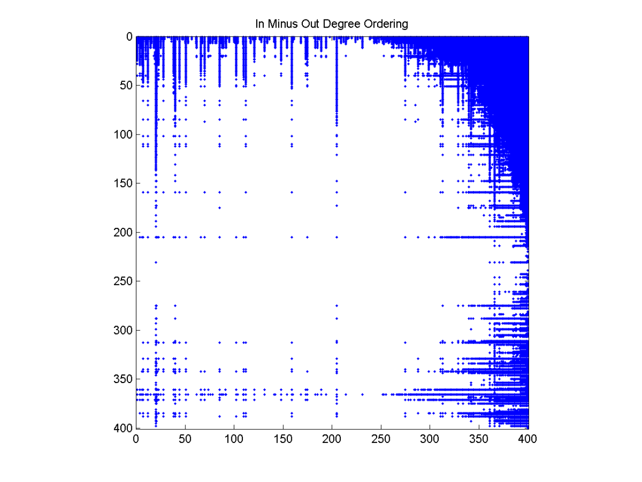


Current Challenge: Spikes in activity



Estimate **who will be busy** during a spike
Explain the **10-20 minute half-life**

Current Challenge: Discovering Hierarchy



Part 2, Modelling: **Next Time ...**

Part 2 will look at **mathematical models** for dynamic networks

Part 2: Network Models

Part 2: Network Models

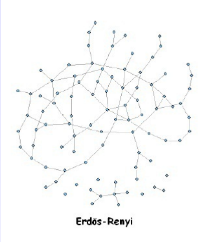
First: static network models

Model: typically a **probabilistic rule** for joining pairs of nodes.

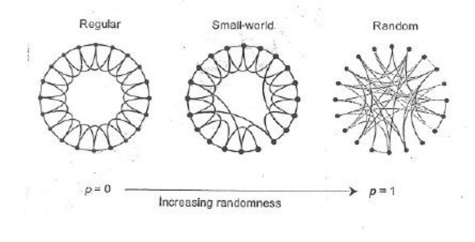
Useful if we want to know whether a given network has some **unusual structure**

Influential Random Network Models

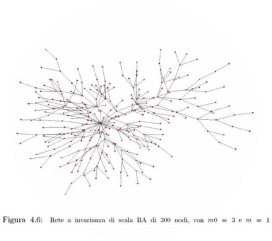
Erdös-Rényi, 1959,   ER(p)



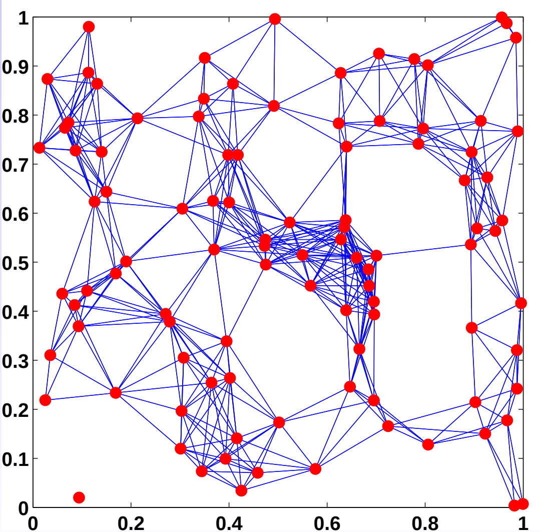
Watts-Strogatz, 1998



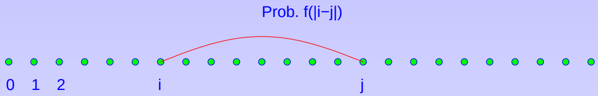
Barabási-Albert, 1999

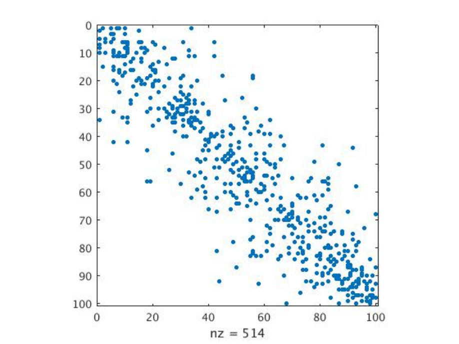


Geometric, e.g,. Penrose 2003



Range-Dependent, Grindrod 2002





Models as Test Matrices

Taylor & Higham, **ACM Trans. Math. Software**, 2009

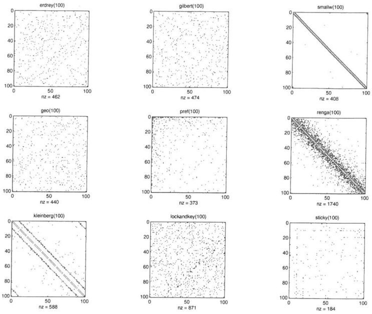


Fig. 1. Spy plots showing nonzero patterns for a 100×100 sample from each of the nine models.

3. MODELS

In this section, we give brief descriptions of the nine models implemented, and show how to use the corresponding MATLAB functions. In each case, the output argument, `A`, is a sparse, symmetric, zero-diagonal matrix of dimension `n`, with `n` being the first of the input arguments. The remaining input arguments take default

Dynamic Network Models

Some modelling challenges. . .

- understand **mechanisms**
- **calibrate** parameters and compare models
- characterize **business as usual**
- **forecast** future behaviour
- simulate **what-if** scenarios

Models for Dynamic Networks

Motivated by digital human interaction,

- **fixed** set of nodes
- edges that may **appear and disappear** over time

Further, assume that over discrete time points

- dynamics are **Markovian**
- each edge can be treated **independently**

Triadic Closure

Grindrod, Higham & Parsons, **Internet Mathematics**, 2012

Friends of friends become friends

Given N people, "friending" and "unfriending"
Let $A^{[k]}$ be the adjacency matrix at time k

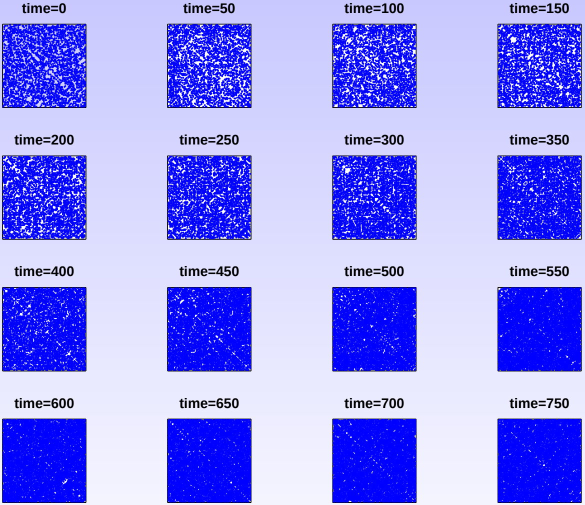
Edge death probability is a constant $\omega \in (0, 1)$
Edge birth probability between nodes i and j given by

$$\delta + \epsilon \left(\left(A^{[k]} \right)^2 \right)_{ij}$$

where $0 < \delta \ll 1$ and $0 < \epsilon(N - 2) < 1 - \delta$

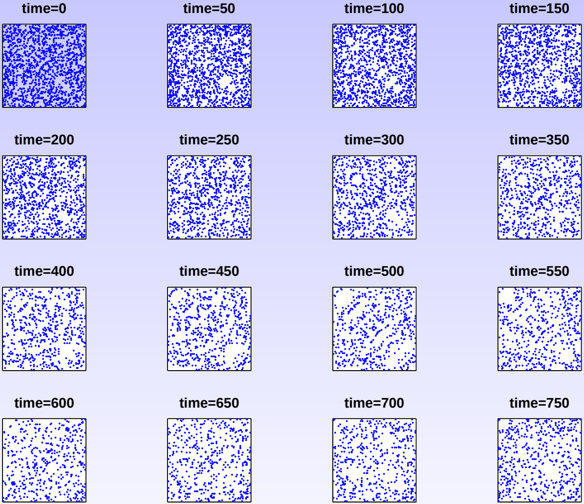
Consider $N = 100$, $\omega = 0.01$, $\epsilon = 5 \times 10^{-4}$, $\delta = 4 \times 10^{-4}$

Triadic closure: start with ER(0.3)



Edge density at time 750 is 0.712

Triadic closure: start with ER(0.15)



Edge density at time 750 is 0.051

Mean field analysis for $\delta + \epsilon\left(\left(A^{[k]}\right)^2\right)_{ij}$

Ergodicity and **symmetry** $\Rightarrow$ Erdös-Rényi limit: every edge present with probablity $p^\star$

Heuristic **mean field** approach: insert the ansatz "$A^{[k]} = \mathrm{ER}(p_k)$" into the model to obtain

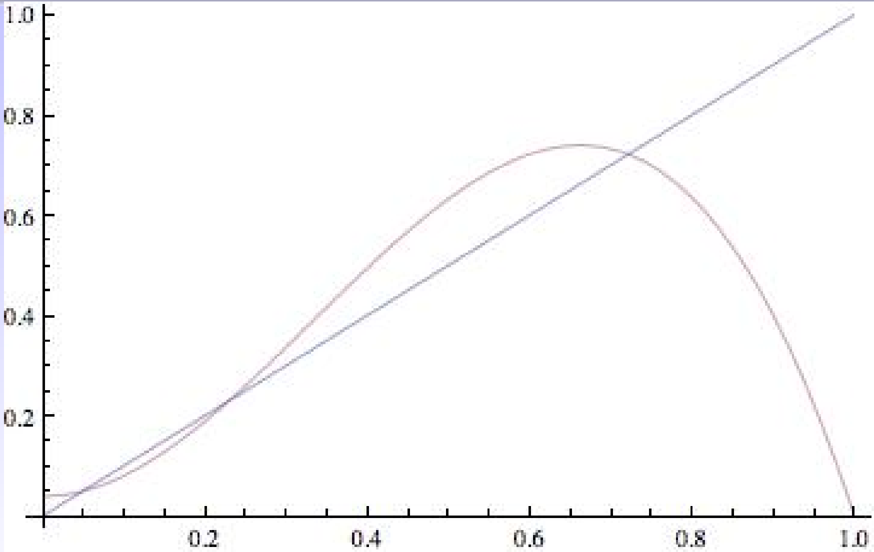
$$p_{k+1} = (1-\omega)p_k + (1-p_k)(\delta + \epsilon(N-2)p_k^2)$$

Generically: **three real roots**

Two are stable, **one is unstable**

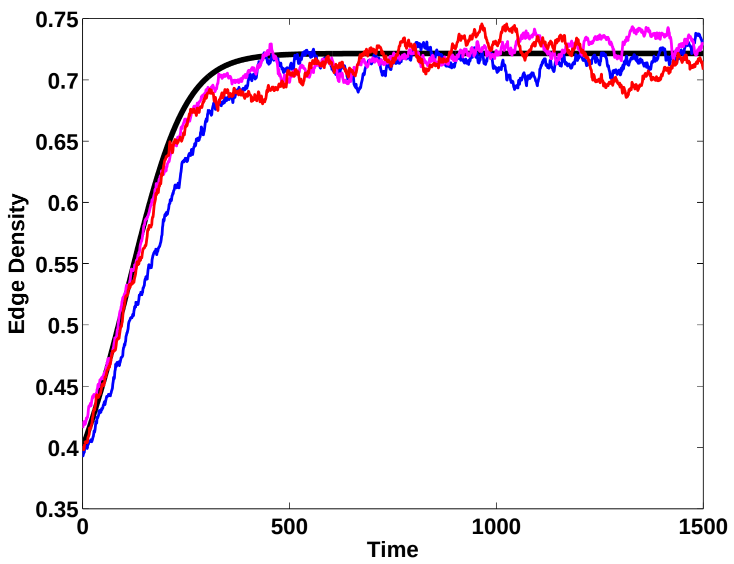
$N = 100$, $\omega = 0.01$, $\epsilon = 5 \times 10^{-4}$, $\delta = 4 \times 10^{-4}$

$N = 100, \omega = 0.01, \epsilon = 5 \times 10^{-4}, \delta = 4 \times 10^{-4}$

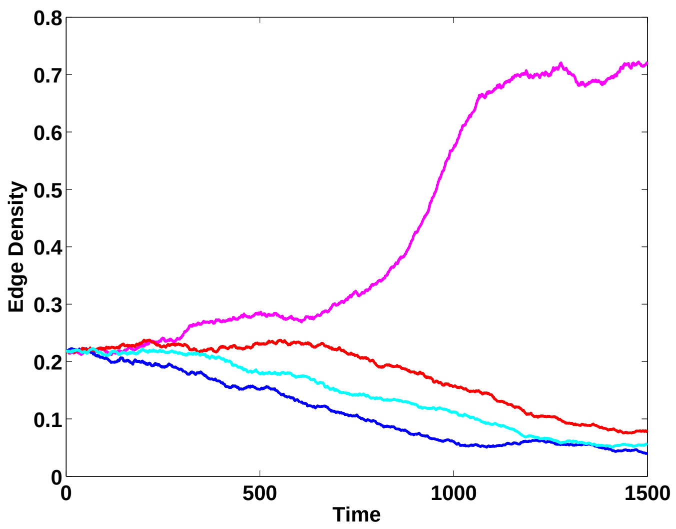


Stable fixed points at $p^\star = 0.049$ and $p^\star = 0.721$
Unstable fixed point at $p^\star = 0.229$

Mean-field vs. simulation from $\mathrm{ER}(0.4)$



Four simulations from $\mathrm{ER}(0.23)$



Stable fixed points 0.049 & 0.721 Unstable 0.229

Calibration/Inference

Mantzaris & Higham, in *Temporal Networks*, Springer, 2013, edited by P. Holme and J. Saramäki

Given model parameters, we can compute the probability of observing the data: **likelihood**

Tests on synthetic data show that we can correctly infer the triadic closure effect

Wealink data from Hu and Wang, Phys. Lett. A, 2009.
26 Million time stamps, over 841 days
0.25 Million nodes
No edge death

Growth of Edges

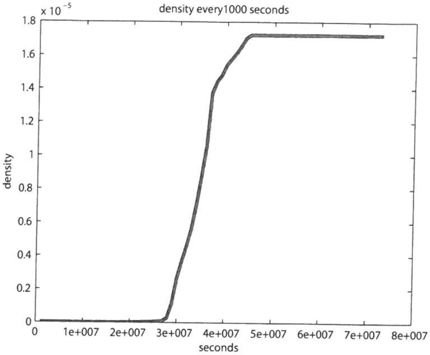


Fig. 11: Density of the Wealink online social network as a function of time in seconds.

Evidence for $\epsilon > 0$ in this dynamic network

18 Alexander V. Mantzaris and Desmond J. Higham

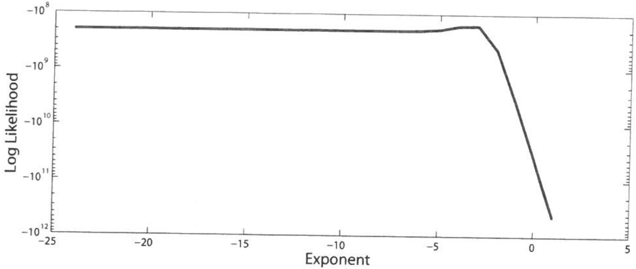


Fig. 13: Log likelihood of the triadic closure model as a function of triad closure strength, ε. The basal edge birth is fixed at 9.95×10^{-8}. The x-axis shows the log base 10 values used for ε in the search for the largest loglikelihood.

Although likelihood-based calibration and model comparison is conceptually straightforward for stochastic, Markov chain based, models of the type used here,

Other Edge Dynamics?

- higher order **motifs**
- **centrality** (rich get richer)
- **hierarchy** (chain of command)
- **gravity** models (relative location)

Could also couple topology with state of node

- **homophily** (like-minded nodes associate)
- **heterophily** (opposite of homophily)
- **social balance** (my enemy's enemy is my friend)

In principle, **model comparison** can be used to determine which one best fits the data

What's Next?

Algorithms

- efficient communicability computation
- detection of communities over time

Modelling/Calibration/Prediction

- explain 20 minute half-life for Twitter spike-decay
- compare social science hypotheses (triadic closure, homophily, heterophily, hierarchy)

Thanks to
CAPITA
EPSRC/RCUK Digital Economy programme
Leverhulme Trust
National Physical Laboratory
Royal Society
Wolfson Foundation
EPSRC/Strathclyde Impact Acceleration Account

Bloom Agency