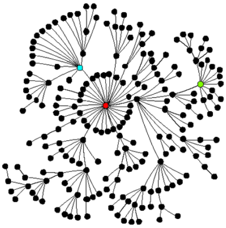
On likelihood models for evolving graphs



Richard G. Clegg (richard@richardclegg.org)
Dept. of Electronic and Electrical Engineering, UCL
Help from Raul Landa and Miguel Rio (UCL), Uli Harder (Imperial), Hamed Haddadi (QMUL), Ben Parker (Southampton), Damien Fay (Bournemouth)

Talk to UCL stats 2014

(Prepared using LaTeX and beamer.)

Introduction

- Evolving networks (graphs/topologies) are an important topic for research.
- Want to describe and understand processes which govern evolution.

Problem statement (vague)

- Want to grow networks with the same properties as real networks.
- Want to be able to describe the evolution of the real network.
- Want to be able to compare rival theories about the evolution.

Topology modelling – the 1 minute history

Scale free networks

A scale free network is one where the degree distribution follows a power law – $\mathbb{P}[\deg = i] \sim i^{-\alpha}$.

Scale free networks said to include:

- Internet Autonomous System (AS) graph [Faloutsos x 3 INFCOM 1999],
- hyperlinks in web pages / wikipedia,
- co-authorship/citation networks, and other social networks,
- biological networks (protein networks).

Preferential attachment

Probability of attach to node prop to node degree. Leads to scale free network (Barabási–Albert [Science 1999]).

Other models – mainly Internet focused

- Waxman model [Waxman IEEE Selected Areas in Communication 1988] – predates scale-free discovery.
- Generalised Linear Preference (GLP model) [Bu–Towsley, INFOCOM 2004] – uses non-linear connection probabilities.
- Positive Feedback Preference (PFP model) [Zhou–Mondragón Phys Rev E 2004]
  - Prob. of connecting to i is $p_i \sim d_i^{(1-\delta \log_{10} d_i)}$ where δ is a tunable parameter.
  - Combined with *interactive growth* model (how internal links connect).
  - δ tuned "by hand" to reproduce a number of statistics of interest.
  - Accounts for the fact that the fact that the internet is not pure power law.

The "basket of statistics" approach

- Current approach – call it the "basket of statistics" method.
  1. Select several statistics which can be measured on net snapshot.
  2. Use test model to grow test network (same size as real network).
  3. Compare the "basket of statistics" on real and test.
- New statistics motivate new models – but what if not all stats match?

Topology modelling appears to be progressing in the following manner:

1. Analyse snapshot of graph (topology) of interest.
2. Find some statistic the current model does not replicate (add this to "basket").
3. Create a new model which replicates the new statistic without affecting old ones.
4. Test using the above procedure.

Refined problem statement

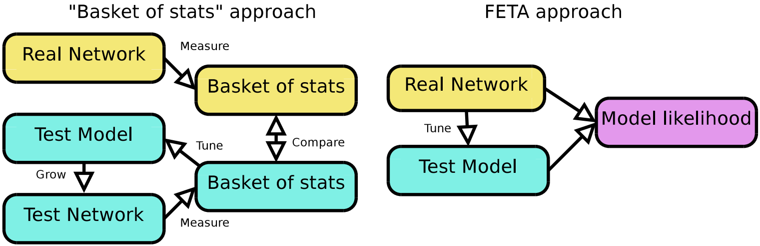
- Let $G(t)$ be a time evolving graph which evolves according to some probabilistic process.
- Let $\mathbf{G} = (G_i, G_{i+1}, \ldots, G_{i+n})$ be random variables representing this process observed at discrete times.
- Let $\mathbf{g} = (g_i, g_{i+1}, \ldots, g_{i+n})$ be a set of observations of $\mathbf{G}$.

Problem statement — more precise

Given observations of a graph $\mathbf{g}$ want to:

- Create models which formally specifies $\mathbb{P}\left[G_{t+1} = g_{t+1} | G_t = g_t, \ldots\right]$.
- Measure the likelihood of such a model producing $\mathbf{g}$.
- Automatically test many such models.

FETA approach



A probabilistic model of graph evolution

- Creating a parameterised model $M(\theta)$ of $\mathbb{P}[G_{t+1} = g_{t+1} | G_t = g_t, \ldots]$. is not straightforward.
- This is not like normal stochastic process. The dimensionality of $G(t)$ changes over time.
- Could transform to some multi-dimensional process with dimension highest dimension graph will achieve (nasty solution).
- Also want a solution which is compatible with existing research in field (can test existing research methods).

The FETA model structure

Operation model

- Process to select an operation on the network.
- Could be: add node, add edge, remove node and so on.

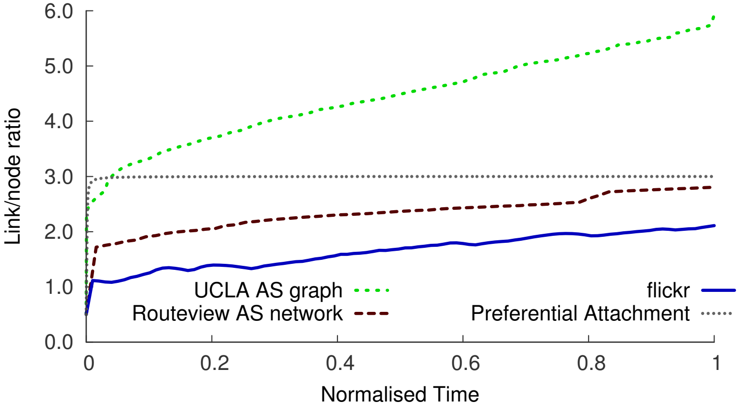
Object model

- Process selects which nodes/edges are involved in operation selected by operation model.
- Probabilities are assigned to nodes and potential edges for random selection.
- Edges selected by assigning probabilities to node pairs.
- Object model is main focus of this presentation.

FETA Model – operations model example

- Results reported here operation model can select from:
  1. NewNodes(n, m) Create a new node and connect it to n new nodes and m existing nodes.
  2. NewLinks(n) Select an existing node and connect it to n existing nodes.
  3. NewClique(n, m) Create a clique between n new nodes and m existing nodes.
- Example: Original preferential attachment model is: NewNodes$(0, 3)$.
- Graph evolution is broken down into the addition of cliques, new nodes and links between existing nodes. (There is some ambiguity here).
- The full operations model gives the probability of each operation (with parameters) at each time step.
- More focus needed on the operations model. Here it is just "copied" for real data.

Importance of operations model



Object model examples

- For simplicity consider graphs which evolve using only the NewNode$(0, 1)$ operation – a new node is created and connects to one existing node.
- Some function which maps all possible choices (of node or link) to a probability.
- For example the Preferential Attachment model is $p_i = d_i/k$ where:
  - p_i is the probability of choosing node i.
  - d_i is the degree of node i.
  - k is a normalising constant such that $\sum_i p_i = 1$.
- The PFP model is $p_i = d_i^{1+\delta \log_{10}(d_i)}/k$ where δ is a parameter.

The likelihood of FETA model

- Let $M(\theta)$ be a parameterised FETA model which assigns probabilities to operations and object models with some parameters θ.
- Define $f_{i,M(\theta)}(g_i) = \mathbb{P}\left[G_i = g_i | M(\theta), G_{i-1} = g_{i-1}, G_{i-2} = g_{i-2}, \ldots\right]$
- For convenience just write $f_i(g_i)$
- Then the likelihood of the model $M(\theta)$ given the observations $\mathbf{g}$ (from i to $i+n$) is $L(M(\theta)|\mathbf{g}) = \prod_{k=i+1}^{k=i+n} f_k(g_k)$.
- This likelihood defines how likely the model is given the observations (or conversely, how probable the observations given the model).
- It is the ability to assign a true likelihood to the graph evolution which is key to the FETA process.

Usable likelihood

- Define $l(M(\theta)|\mathbf{g}) = \log(L(M(\theta)|\mathbf{g})$.
- Because of normalisation problems standard log-likelihood maximisation techniques do not work.
- Likelihood can be split into operation model and object model components.
- Let M_0 that be the null hypothesis – all choices are equally likely. Let m be the number of choices (* warning – details here).
- Human readable measure is c_0 the per choice likelihood ratio.

Per choice likelihood ratio c_0

$$c_0 = \left[\frac{L(C|F)}{L(C|M_0)}\right]^{1/m} = \exp\left[\frac{l(C|F) - l(C|M_0)}{m}\right].$$

Building object models from components

- Three possible object models have been introduced already.
  1. M_0 – all nodes equal.
  2. M_d – preferential attachment (nodes weighted by degree).
  3. $M_p(\delta)$ – PFP model δ is parameter.
- How about mixture models?
- $M = \beta_1 M_0 + \beta_2 M_d$ (nodes sometimes chosen randomly, sometimes by degree) – $0 < \beta_1 < 1$ and $\beta_1 + \beta_2 = 1$.
- On the positive site, a larger family of explanations, on the negative, more parameterisation.

Object model components

Throughout k is a normalising constant such that $\sum_i p_i = 1$ for all nodes considered. p_i is the probability of picking node i (at the stage being considered).

- Random model M_0 $\quad p_i = 1/k$.
- Preferential attachment M_d $\quad p_i = d_i/k$.
- PFP $M_p(\delta)$ $\quad p_i = d_i^{1+\delta \log_{10}(d_i)}/k$ where δ is a parameter.
- Degree power $M_d(\alpha)$ $\quad p_i = d_i^{\alpha}/k$ where α is a parameter.
- Triangle model M_t $\quad p_i = t_i/k$ where t_i is the triangle count of node i.
- Singleton model M_1 $\quad p_i = \begin{cases} 1/k & d_i = 1 \\ 0 & \text{otherwise} \end{cases}.$
- Doubleton model M_2 $\quad p_i = \begin{cases} 1/k & d_i = 2 \\ 0 & \text{otherwise} \end{cases}.$
- Hot model $M_h(n)$ $\quad p_i = \begin{cases} 1/k & \text{node chosen in last } n \text{ picks} \\ 0 & \text{otherwise} \end{cases}$

  where n is a parameter.

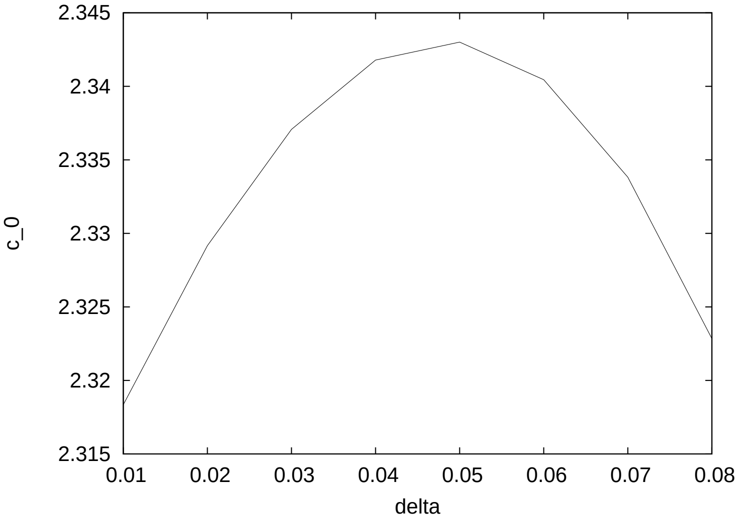
A GLM approach to optimise β parameters

- Want to automatically fit β_i in models of form $M = \beta_1 M_1 + \beta_2 M_2 + \cdots$.
- Functional form looks temptingly like a generalised linear model.
- Let $p_{i,j}$ be the probability model assigns to node i at step j.
- Cannot fit to $p_{i,j}$ at each stage because probability is not directly measureable.
- Instead all we know is whether node i was actually selected or not at stage t.
- Let $I_{i,j}$ be an indicator variable such that $I_{i,j}$ is one if node i was chosen for choice j and zero otherwise.
- By definition $\mathrm{E}\left[I_{i,j}\right] = p_{i,j}$.
- Therefore, we fit models of the form $I_{i,j} = \beta_1 M_1 + \beta_2 M_2 + \cdots$.
- Obviously many models of this form can be tried. Statistical significance will reject unnecessary variables.

Artificial tests

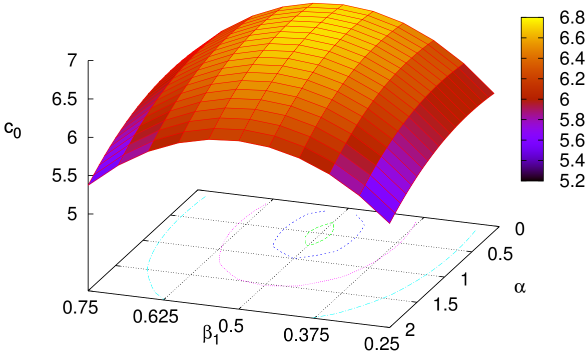
- Perhaps the most convincing test of such a model is its ability to recover parameters from a known model.
- Build a model with known $M(\theta)$. Assume a model structure, try to recover θ.

Sweep one parameter (10,000 link network)



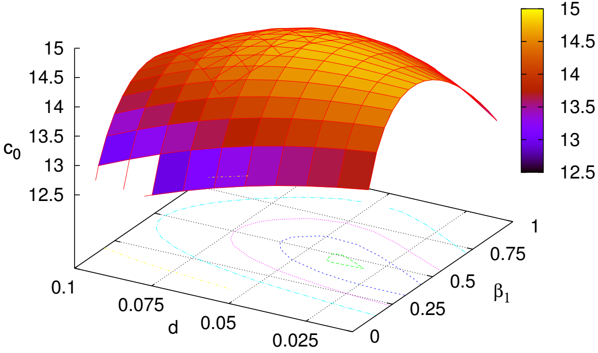
PFP model $M = M_d(0.05)$. Correct answer is $\delta = 0.05$.

Sweep two parameters (10,000 link network)



Correct model $M = 0.5M_2 + 0.5M_d(0.5)$ fitted
$M = \beta_1 M_2 + (1 - \beta_1)M_d(\alpha)$.

Sweep two parameters (10,000 link network)



Correct model $M = 0.5 M_p(0.05) + 0.5 M_t$ fitted

$$M = \beta_1 M_p(d) + (1 - \beta_1) M_t$$

Parameter recovery using GLM procedure

- Test model $M = 0.25M_0 + 0.25M_t + 0.25M_1 + 0.25M_2$.
- Random model + triangle model + singleton model + doubleton model.
- Generate 10,000 links and fit using GLM.

Parameter	Estimate	Significance
β_0	0.23 ± 0.021	0.1%
β_t	0.28 ± 0.017	0.1%
β_1	0.24 ± 0.016	0.1%
β_2	0.25 ± 0.020	0.1%

GLM procedure with incorrect model

- In reality we do not know which model components to use.
- Here the GLM is tested with an additional spurious model component M_d (preferential attachment).
- The M_d component is rejected.

Parameter	Estimate	Significance
β_0	0.33 ± 0.059	0.1%
β_t	0.29 ± 0.017	0.1%
β_1	0.24 ± 0.016	0.1%
β_2	0.23 ± 0.022	0.1%
β_d	-0.089 ± 0.059	5%

General comments on GLM procedure

- Works well to recover parameters to known model.
- Can have issues when model components express "similar" things (e.g. PFP and preferential attachment in same model).
- Acts as a guide to the user as to which model components to include and which to reject.
- Does not allow testing of non-linear parameters (e.g. δ) but can be combined with "parameter sweep".
- Occasionally fails badly – parameters always sum to 1 but can be negative.
- Sample point "explosion" each choice has as many samples as nodes in graph. Over specified model...
- Use train, cross-validate, test sampling methodology (not short of data).
- Ultimately though, the likelihood estimate c_0 is the arbiter of which model is correct.

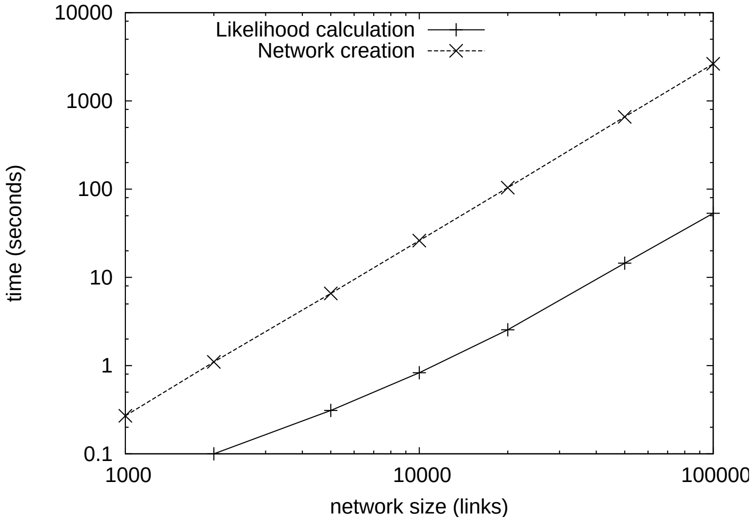
Real data tests

- Tests have been performed on seven real networks:
  1. Two views of Internet autonomous system graph.
  2. Two photo sharing websites.
  3. ArXiV linked publications.
  4. Facebook wall posts.
  5. Enron email database.
- Model sizes varied from 15,788 links to 200,000.
- Hypothetical models are created from components using FETA (and GLM) and their c_0 measured.

Real data test claims

- In order to make a comparison we use the operations model by "cloning" the real operations and test four object models:
  1. Preferential attachment M_d.
  2. Degree Power (tuning α) $M_d(\alpha)$.
  3. PFP model (tuning δ) $M_p(\delta)$.
  4. Best model found combining all elements and tuning parameters.
- Models are assessed by comparing c_0 – higher is "better fit".
- Graphs are then "grown" using the various models to compare their parameters with the real network.
- The dynamic behaviour of target statistics is plotted as deviation from the real data.

Runtime of likelihood estimate versus network creation



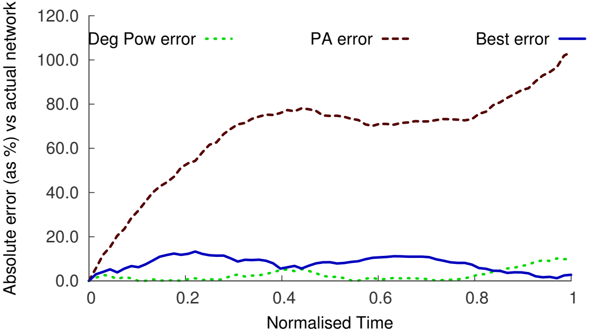
Data sets

- Facebook data:
  - Facebook data 200,000 public Facebook wall posts.
  - Time stamped so dynamic behaviour available.
- Enron data:
  - 250,000 emails from Enron – released as part of investigation into disgraced company.
  - Time stamped email with recipients form directed dynamic network.
- Treated as undirected here and duplicates (and self links) removed.

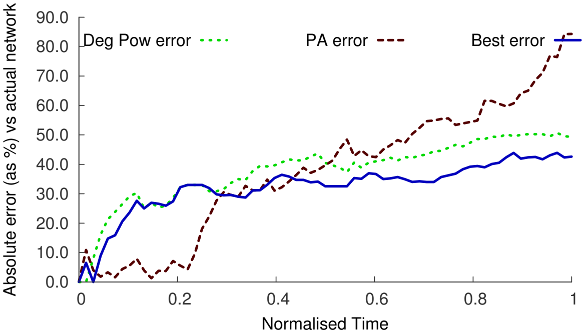
Facebook models

- The Preferential attachment model has $c_0 = 1.091$.
- Highest c_0 PFP model has $c_0 = 1.201$ at $\delta = -0.225$.
- Highest c_0 degree power model has $c_0 = 1.220$ with $\alpha = 0.575$.
- Best model is a mixture of random and degree power
- It has $c_0 = 1.221$ and is $0.3M_0 + 0.7M_d(0.8)$.
- Expect therefore that Best is only slightly better than Degree power and PFP.
- Expect both are better than Preferential attachment.
- Note that due to more degrees of freedom this will always be the ordering (PA special case of PFP and Degree-power).

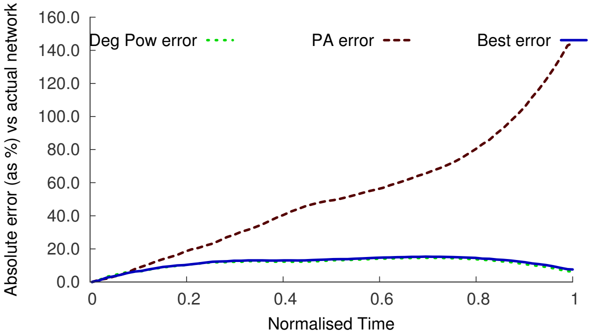
Facebook data – number of nodes of degree 1



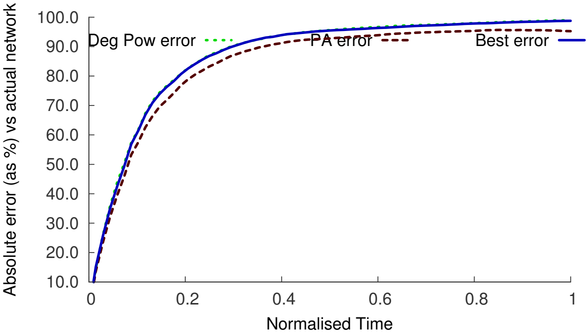
Facebook data – degree of maximal degree node



Facebook data – mean square node degree



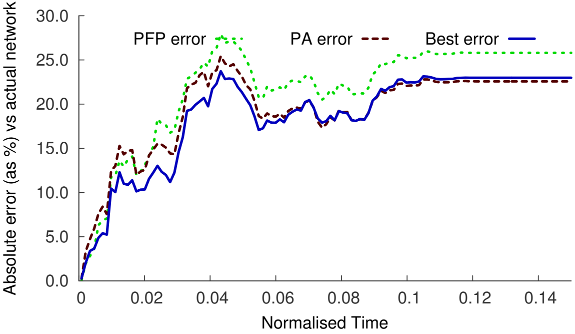
Facebook data – clustering coefficient



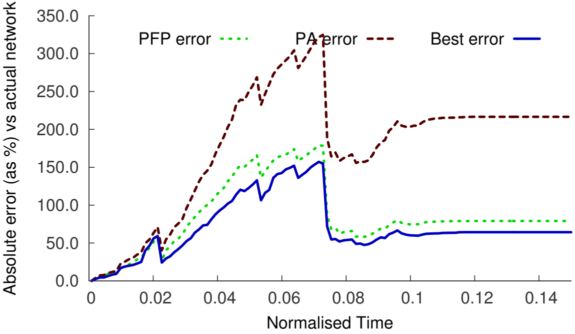
Enron data model

- The preferential attachment model gives $c_0 = 4.898$.
- PFP model has maximal $c_0 = 4.927$ when $\delta = -0.02$.
- The degree power model has its maximum $c_0 = 4.903$ with $\alpha = 0.98$.
- The "best" model has $c_0 = 21.35$ and combined PFP and the "hot" model.
- It is given by $M = 0.75M_p(-0.02) + 0.25M_h(1)$.
- Expect "best" is much better than PFP or Degree power.

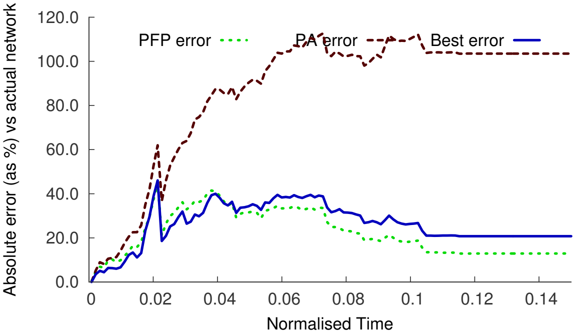
Enron data – number of nodes of degree 1



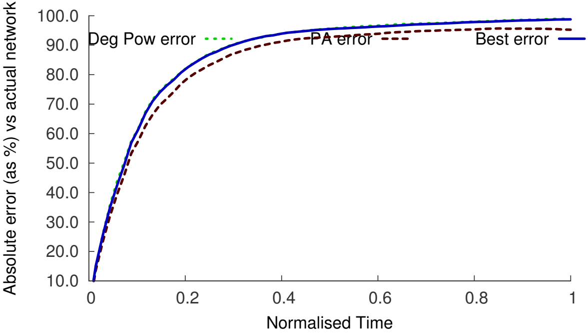
Enron data – degree of maximal degree node



Enron data – mean square node degree



Enron data – clustering coefficient



Results summary

- Clearly even the best models were not perfect on all data – this is true of all data analysed.
- It can also be seen that analysis of a snapshot might fool researcher that a model was excellent when it was poor.
- Roughly the models were in the order predicted by c_0.
- The main exception is that the "Best" model for Enron would be expected to be much better but is only a little better.
- No models capture the clustering coefficient well.
- However, this provides reasonable evidence that tuning models using c_0 produces a "better" fit to graphs.

Conclusions

- The likelihood parameters and the null model here provide a rigorous way to assess a potential dynamic model of network evolution.
- Known model parameters can be recovered using sweeps of likelihood or GLM for linear parameters.
- The likelihood is reflected in improved performance on replicating network statistics.
- The advantages of this framework are several:
  1. Assesses the dynamic history of the data not statistics of a snapshot.
  2. Single statistically rigorous estimate of model likelihood.
  3. Quicker than growing a network and testing statistics (using same codebase).
- An exciting new way to test theories about topologies if you have the data for it.

Further work

- What model components can be added (particularly for assortativity and clustering).
- More data must be found – currently tested on seven networks but need more.
- Further work must be done on the operations model.
- Multiplicative model combinations for the object model might have greater success: $M = KM_d^{\beta_d} M_T^{\beta_T} \cdots$.
- Software and data freely available – please email `richard@richardclegg.org`
- See also the website (needs updating –work on improved Java code very much underway) `http://www.richardclegg.org/software/FETA`
- I am very keen to collaborate – this idea is interesting but needs development.