Modeling Large-Scale Networks

Tamara G. Kolda, Sinan Aksoy, Ali Pinar, Todd Plantenga, C. Seshadhri, Dylan Stark

40th Numerical Analysis Conference Woudschoten
Past, Present and Future of Scientific Computing
Zeist, The Netherlands
October 8, 2015
Networks are Increasingly Prevalent in Data Analysis...

- Computer traffic
- Social networks
- Biological signaling
- Communications
- Financial analysis
- Physical proximity
- Recommendation systems
- Publishing
- Etc.

One-Way Network

Two-Way Network
Network Models yield Understanding

- Discover underlying principals
  - “Physics”
  - Global vs. local properties
- Determine key metrics
  - Degree distribution
  - Motif (triangles, etc.) distribution
  - Community structure
  - Diameter
  - Eigenvalues
  - Etc.
- Generate artificial data
  - Scale up or down in size
  - Surrogate for real data, protecting privacy and security
  - Easy to share and reproduce
  - Compressed representations
- Desired model properties
  - Calibrates to real data
  - Scalable to billions of edges
A Good Network Model should have a Heavy-Tailed Degree Distribution

The **degree distribution** is one way to characterize a graph.

Barabasi & Albert, Science, 1999: “A common property of many large networks is that the vertex connectivities follow a scale-free power-law distribution”

\[ d_B = 4 \]
\[ d_F = 4 \]
\[ d_G = 3 \]
A Good Network Model should have High Clustering Coefficients

\[ d_i = \text{degree of node } i \]
\[ t_i = \text{triangles involving node } i \]

Node clustering coefficient:
\[ c_i = \frac{t_i}{\binom{d_i}{2}} \]

- \[ c_B = 1 \]
- \[ c_F = \frac{1}{6} \]
- \[ c_G = 0 \]

Degree-\(d\) clustering coefficient:
\[ c_d = \text{mean} \{ c_i | d_i = d \} \]

Global clustering coefficient:
\[ c = \frac{\sum_i t_i}{\binom{\sum_i d_i}{2}} \]

In social networks, the clustering coefficients decrease smoothly as the degree increases. High degree nodes generally have little social cohesion.
Current Network Models Cannot Match Both Degree & Triangle Dist.

Focus on One-Way Models

- Erdős-Rényi (1960)
  - All edges have equal probability
  - Con: Poisson degree distribution

- Preferential Attachment (Barabási-Albert 1999)
  - Nodes join the graph sequentially
  - Prefer nodes of higher degree
  - Pro: Power-law degree distribution
  - Con: Too few triangles

- Stochastic Blockmodel (Holland et al. 1983)
  - Each node belongs to a block
  - Edge probability between blocks
    \[
    \begin{bmatrix}
    0.6 & 0.1 \\
    0.1 & 0.4 \\
    \end{bmatrix}
    \]
  - Pro: Explicit community structure
  - Con: Wrong degree distribution
  - Con: Too few triangles

- Stochastic Kronecker, aka R-MAT (Chakrabarti et al. 2004)
  - Edge probabilities defined by Kronecker products of generator matrices
  - Pro: Scalable
  - Con: Wrong degree distribution
  - Con: Too few triangles

- Chung-Lu (2002), aka Configuration Model
  - Edge probabilities defined by desired degree of endpoints
  - Pro: Scalable
  - Pro: Matches many degree distributions
  - Con: Too few triangles
Fast Chung-Lu: Scalable Generator that Matches Degree Distribution

- Given degree distribution
- ⇒ Know desired degree of each node, \( d_i \)
- Total edges \( E = \frac{1}{2} \sum d_i \)
- Choose 2 endpoints at random per edge, proportional to \( d_i \)

\[
\text{Prob}(i \text{ selected}) = \frac{d_i}{2E}
\]

CL Matches Degree Distribution but not Clustering Coefficients
Non-negligible Clustering Coefficients Requires Dense Subgraphs!

- For sparse graphs, very small chance that a node’s neighbors are connected
- But, high clustering coefficient $\Rightarrow$ neighbors heavily connected
- Theorem: There must be dense Erdős-Rényi subgraphs!
- We create “affinity blocks” of heavily connected nodes
  - Each affinity block is an Erdős-Rényi graph
Block Two-Level Erdös-Rényi (BTER) creates Affinity Blocks

Preprocessing
- Create affinity blocks of nodes with (nearly) same degree, determined by degree distribution

Phase 1
- Erdös-Rényi graphs in each block
- Essentially all triangles occur in these blocks
- Connectivity per block based on clustering coefficient

Phase 2
- CL model on excess degree
- Creates connections across blocks

Seshadhri, Kolda, Pinar (Phys. Rev. E 2012)
Kolda, Plantenga, Pinar, Seshadhri (SISC 2014)
Affinity Blocks of ER Subgraphs with a Specified Clustering Coefficient

Some edges are dedicated to the ER subgraph. The remainder are “excess degree.”

ER subgraph
\( d + 1 \) nodes with \( d_i = d \)

\( \rho = \text{connection probability} \)
\( \Rightarrow \mathbb{E}(t_i) = \binom{d}{2} \rho^3 \)

\( c_i = \text{node clustering coefficient} \)
\( \Rightarrow t_i = \binom{d}{2} c_i \)

\( \Rightarrow \rho = \sqrt[3]{c_d} \)

Random ER subgraph
6 nodes of degree 5
Labeled by # triangles
Excess degree denoted by red stubs.

\( c_d = 0.3 \) for \( d = 5 \)
\( \Rightarrow \rho = \sqrt[3]{c_d} = 0.67 \)
\( \mathbb{E}(t_i) = 3 \)

Actual: \( \bar{t}_i = 3.5 \)
BTER has Many Affinity Blocks; Blocks are Relatively Small

Adjacency Matrix

Red = Phase 1
Blue = Phase 2

Adjacency Matrix - Lower Right Corner
BTER has better Clustering Coefficients than CL or SKG
BTER has better Clustering Coefficients than CL or SKG (again)
BTER has better Eigenvalues too!
Open Question: Can BTER Capture Higher-order (4-Vertex) Patterns?

<table>
<thead>
<tr>
<th></th>
<th>3-star</th>
<th>3-path</th>
<th>K₃+tail</th>
<th>C₄</th>
<th>C₄+chord</th>
<th>K₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>1.74e10</td>
<td>8.35e09</td>
<td>1.41e09</td>
<td>7.21e07</td>
<td>7.85e07</td>
<td>5.89e06</td>
</tr>
<tr>
<td>BTER</td>
<td>9.88e09</td>
<td>7.40e09</td>
<td>1.49e09</td>
<td>8.23e07</td>
<td>1.11e08</td>
<td>1.43e07</td>
</tr>
</tbody>
</table>
Edge Independence is Key to Scalability for BTER

- **Phase 1**
  - Edge independence:
    - Choose random block proportional to its “weight”
    - Choose uniform random edge within block
  - Single block $b$
    - Block size $= n_b$
    - Connectivity $= \rho_b$
    - Expected # edges $= \rho_b n_b (n_b - 1)/2$
    - Weight $= \#$ edges to be inserted
      \[
      w_b = \binom{n_b}{2} \ln \left( \frac{1}{1-\rho_b} \right)
      \]
    - Total edge insertions $= \sum_b w_b$

- **Phase 2 edges:**
  - Edge independence: Fast CL based on **excess degree**
  - To run simultaneously with phase 1, compute **expected** excess degree:
    \[
    e_i = d_i - (\rho_b \cdot d_b)
    \]
  - Total edge insertions $= \frac{1}{2} \sum_i e_i$
Scalable BTER is Based on a Series of Random Decisions, Cost = O(M log N)

Choose phase 1 or 2?

Choose block proportional to number of “samples” per block

Create Block 1 edge per ER model with connectivity \( \rho_1 \)
Choose 1st endpoint
Choose 2nd endpoint

Create Block K edge per ER model with connectivity \( \rho_K \)
Choose 1st endpoint
Choose 2nd endpoint

Create Phase 2 edge using CL model on expected “excess degree”

Choose 1st endpoint
Choose 2nd endpoint

Requires O(n) data to determine the various probabilities, and can be compressed to O(\( d_{\text{max}} \)).
BTER Phases 1 & 2 Simultaneous

Preprocessing
- Create affinity blocks of nodes with (nearly) same degree, determined by degree distribution
- Connectivity per block based on clustering coefficient
- For each node, compute desired
  - within-block degree
  - excess degree

Phase 1
- Erdős-Rényi graphs in each block
- Need to insert extra links to insure enough unique links per block

\[ w_b = \frac{n_b}{2} \ln \left( \frac{1}{1-\rho_b} \right) \]

Phase 2
- CL model on excess degree (a sort of weighted Erdős-Rényi)
- Creates connections across blocks

Occurring independently

Seshadhri, Kolda, Pinar (Phys. Rev. E 2012)
Kolda, Plantenga, Pinar, Seshadhri (SISC 2014)
uk-union: 122M nodes, 4.7B undirected edges, $d_{\text{avg}} = 76$. Clustering coeff. = 0.007
BTER model: 120M nodes, 4.4B undirected edges, $d_{\text{avg}} = 73$, clustering coeff. = 0.111

Data Source: Laboratory for Web Algorithms [http://law.di.unimi.it/datasets.php](http://law.di.unimi.it/datasets.php)
BTER Scales in MapReduce: 15 min for 4B-edge network

32-node x 4-procs-per-node Hadoop Cluster

- Total edges
- Unique edges

# mappers = # vertices / 1M
# reducers = min{ 128, # mappers }

Ran out of independent processors

Kolda, Plantenga, Pinar, Seshadhri (SISC, to appear)
BTER Scales in MPI:
3 min for 18B-edge network

- Setup
  - 32 nodes
  - 32 GB RAM per node
- Scaling
  - 32 to 1,024M vertices
  - Up to 18B edges
- BTER set up and edge generation scale nicely, as expected
- Total time = 3 minutes for 18B unique edges

- Thanks to Dylan Stark (Sandia) for MPI version and compiling these results
- Future work: Remove need for edge deduplication
BTER Benchmark Degree Distribution: Recommend Generalized Log Normal

- **Power Law (PL)**
  \[ n_d \propto d^{-\gamma} \]

- **Generalized Log-Normal**
  \[ n_d \propto \exp\left[-\left(\frac{\log d}{\alpha}\right)^\delta\right] \]

- **Discrete versions**
  \[ Pr(D = d) = f(d)/\left(\sum_{d'=1}^{d^*} f(d')\right) \]

- User specifies **desired average degree** and **absolute max degree**. Also require tolerance so that \( n \cdot \epsilon_{tol} \ll 1 \).
  \[ d = \sum_{d=1}^{d^*} d \cdot f(d) \text{ and } Pr(D = d^*) < \epsilon_{tol} \]

- Also have method for picking clustering coefficients that requires **desired global clustering coefficient** and **absolute max clustering coefficient**

\[ n = 10^7, \bar{d} = 16, d^* = 10^6 \]
Proposed Benchmark for BTER
Requires only 5 Parameters

1. # vertices = 10^6
2. Avg. degree = 16
3. Max degree = 10^4
4. Max CC = 0.50
5. Global CC = 0.10

BTER Realization
8M edges
Max degree = 2,594
Avg degree = 17
Global CC = 0.104
Time = 26 sec.

Kolda, Pinar, Plantenga, Seshadhri (SISC, 2014)
Bipartite Graphs: aka Hypergraphs, Two-way Graphs, Affiliation Networks

- Vertices separated into two partitions
- Edges only allowed between partitions
- Many networks have natural bipartite structure:
  - Author-Paper
  - Actor-Movie
  - Person-Group
  - Protein-Function
  - P2P Exchange (User-File)
  - Company Board-Member
  - Word-Sentence
  - User-Rating
A Good Bipartite Model should have Heavy-Tailed Degree Distributions

Condensed Matter Papers (Newman 2001)

IMDB (Latapy, Magnien, Del Vecchio 2008)

P2P (Latapy, Magnien, Del Vecchio 2008)

Flickr (Mislove et al. 2007)
Fast Bipartite Chung-Lu: Generator that Matches Degree Distributions

- Given degree distributions for red and blue
- Know *desired* degree of each node, $d_i$ for red and $d_j$ for blue
- Total edges $E = \sum d_i = \sum d_j$
- Choose one red and blue endpoints at random per edge

\[
\text{Prob}(i \text{ selected}) = \frac{d_i}{E} \quad \text{Prob}(j \text{ selected}) = \frac{d_j}{E}
\]
Fast Bipartite Chung-Lu Matches Degree Distributions for Real Data

Author-Paper Network (Newman, 2001)

Degree Distribution

P2P File Exchange (Latapy, 2008)

Degree Distribution

Actor-Movie Network (Latapy, 2008)

Degree Distribution

Flickr User Group (Mislove, 2007)

Degree Distribution
Fast Bipartite Chung-Lu Matches Degree Distributions for Real Data

Condensed Matter Papers (Newman 2001)

IMDB (Latapy, Magnien, Del Vecchio 2008)

P2P (Latapy, Magnien, Del Vecchio 2008)

Flickr (Mislove et al. 2007)
Metamorphosis is the Bipartite Analogue of Clustering Coefficient

Caterpillars:
\[ c_{(i,j)} = 3\text{-paths with center } (i, j) = (d_i - 1)(d_j - 1) \]

Butterflies:
\[ b_{(i,j)} = 4\text{-cycles with edge } (i, j) \]

Metamorphosis for Edge:
\[ m_{(i,j)} = b_{(i,j)}/c_{(i,j)} \]

Metamorphosis per Vertex:
\[ m_i = \text{mean}\{m_{(i,j)} | (i, j) \in E\} \]

Metamorphosis per Degree:
\[ m_d = \text{mean}\{m_i | d_i = d\} \]

Global Metamorphosis:
\[ m = \sum_{(i,j) \in E} b_{(i,j)} / \sum_{(i,j) \in E} c_{(i,j)} \]

Global Metamorphosis: Robins & Alexander, 2004
Metamorphosis is not a Consequence of Degree Distribution!

Chung-Lu creates caterpillars, but not enough butterflies

<table>
<thead>
<tr>
<th></th>
<th>Condensed Matter Papers</th>
<th>IMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
<td>Chung-Lu</td>
</tr>
<tr>
<td>Caterpillars</td>
<td>1,236,527</td>
<td>2,187,676</td>
</tr>
<tr>
<td>Butterflies</td>
<td>70,549</td>
<td>339</td>
</tr>
<tr>
<td>Metamorphosis</td>
<td>$2.28 \times 10^{-1}$</td>
<td>$6.20 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Thus, Chung-Lu can’t match per-degree metamorphosis coefficients.
Modeling Bipartite Metamorphosis is Hard

- Balancing act: avoid satisfying properties for one node type at expense of other
  - # nodes, degree range may be different for one node type
  - per-degree metamorphosis may be skewed

- Our goal: develop generative bipartite model matching deg. dists. & per-degree metamorphosis coeff.

“Another [new] direction is the development of models of 2-mode networks capturing properties met in practice. Just as is the case for 1-mode networks, much can be done concerning degrees, but very little is known concerning the modeling of clustering…”

-Latapy, Magnien, & Del Vecchio, Social Networks, 2008
Bipartite BTER creates Bipartite Affinity Blocks

**Preprocessing**
- Create affinity blocks of with combinations of nodes from each partition
- Need to balance different metamorphosis coefficients for each partition and degree

**Phase 1**
- Determine affinity block structure so that an appropriate number of butterflies are created
- Ideally, treat each partition as bipartite Erdős-Rényi graph

**Phase 2**
- Bipartite CL model on excess degree
- Creates connections across blocks
BTER Matches Degree Distributions and Metamorphosis for Real Data

Condensed Matter Papers (Newman 2001)

IMDB (Latapy, Magnien, Del Vecchio 2008)

Condensed Matter Papers (Newman 2001)

IMDB (Latapy, Magnien, Del Vecchio 2008)
BTER and Bipartite BTER are Useful Tools for Graph Generation

- **Generative Graph Models**
  - Key metrics include degree distribution and measures of social cohesion
  - Clustering coefficient (by degree) measures cohesion in one-way graphs
  - Metamorphosis coefficient (by degree and partition) measures cohesion in two-way graphs
  - Useful as data surrogates, benchmarks, etc.

- **BTER Generative Graph Model**
  - Identifies core structures in sparse networks with frequent triangles
  - Matches degree distribution and clustering coefficients
  - Scalable MPI & Hadoop implementations
  - Proposed benchmark using only five parameters

- **BTER Bipartite Generative Graph Model**
  - Matches dual degree distributions
  - Harder to define affinity block structure, but reasonable match to metamorphosis

**Team**
- Sinan Aksoy (UC San Diego)
- Ali Pinar (Sandia)
- Todd Plantenga (FireEye)
- Sesh Comandur (UC Santa Cruz)

**More Information**
Tamara G. Kolda
tgkolda@sandia.gov