Sparse Linear Algebra Issues Arising in the Analysis of Complex Networks

Michele Benzi

Department of Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
Acknowledgments

- Christine Klymko (Emory)
- Ernesto Estrada (Strathclyde, UK)
- Support: NSF DMS-1115692
- Papers and preprints available on web page
Outline

- Complex networks: motivation and background
- Quantitative analysis of networks
- Subgraph centrality and communicability
- Algorithms and numerical experiments
- Some open challenges
Complex networks: motivation and background

- **Complex networks** provide models for physical, biological, engineered or social systems (e.g., molecular structure, gene and protein interaction, food webs, transportation networks, power grids, social networks, ...).
- Graph analysis provides **quantitative tools** for the study of complex networks.
- Techniques from **spectral graph theory**, **linear and multilinear algebra**, **probability**, **approximation theory**, etc. play a major role.

**Network science** today is a **vast** multidisciplinary field. Important early work was done by **social scientists**: sociologists, anthropologists, experimental psychologists, economists and even bibliometrists. More recently, **physicists**, **computer scientists** and **applied mathematicians** have made major contributions.
But what exactly is a complex network?

Unfortunately, no precise definition exists.

It is easier to say which graphs are not complex networks. Regular lattices are not considered complex networks, and neither are completely random graphs such as the Gilbert or Erdös–Rényi models.

Random graphs are, however, considered by many to be useful as toy models for complex networks.
Regular lattice: not a complex network!
Star graph: also not a complex network!
Erdös–Rényi graph: also not a complex network!
Some of the attributes typical of many real-world complex networks are:

- Scale-free (power law degree distribution)
- Small-world (small graph diameter)
- Highly clustered (many triangles, hubs...)
- Hierarchical
- Rich in ‘motifs’
- Self-similar

**Caveat**: there are important examples of real-world complex networks lacking one or more of these attributes.
Example of complex network: B–A model
Example of complex network: Golub collaboration graph
Example of complex network: Erdös collaboration graph
Example of complex network: PPI network of *Saccharomyces cerevisiae* (beer yeast)
Example of complex network: the Internet
Example of (directed) complex network: a food web

![Food Web Diagram]

Marsh Hawk → Bald Eagle → Harbour Seal
Snow Geese → Herring → Salmon
Marsh Vegetation (includes many species of sedges, grasses, bull rushes, algae, etc.)

Picture credits: http://www.physicalgeography.net/fundamentals/9o.html
Notation:

- $G = (V, E)$ is a connected simple graph with $N = |V|$ nodes and $m = |E|$ edges (i.e., $G$ is unweighted, undirected and has no loops)
- $A \in \mathbb{R}^{N \times N}$ is the associated adjacency matrix:
  - $A_{ij} = 1$ if nodes $i$ and $j$ are adjacent, $A_{ij} = 0$ otherwise
  - $A$ is symmetric
  - $A_{ii} = 0$, $1 \leq i \leq N$
  - $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ eigenvalues of $A$
- For a node $i$, define its degree $d_i := \sum_{k=1}^{N} A_{ik}$ (number of immediate neighbors in $G$)
- Graph Laplacian: $L := D - A$ where $D = \text{diag}(d_1, \ldots, d_N)$
Example: PPI network of *Saccharomyces cerevisiae*

Adjacency matrix, $N = 2224$, $m = 6609$. 
Example: PPI network of *Saccharomyces cerevisiae*

Same, reordered with Reverse Cuthill–McKee
Example: PPI network of *Saccharomyces cerevisiae*

Degree distribution ($d_{\text{min}} = 1$, $d_{\text{max}} = 64$)
Example: PPI network of *Saccharomyces cerevisiae*

![Graph showing eigenvalues](image-url)
Example: Intravenous drug users network

Adjacency matrix, $N = 616$, $m = 2012$. 
Example: Intravenous drug users network

Same, reordered with Reverse Cuthill–McKee
Example: Intravenous drug users network

Degree distribution \((d_{\text{min}} = 1, \ d_{\text{max}} = 58)\)
Example: Intravenous drug users network

Eigenvalues
Basic questions about networks include **centrality**, **communicability** and **community detection** issues:

- Which are the most “important” nodes?
  - Network connectivity (vulnerability)
  - Lethality in PPI networks
  - Author centrality in collaboration networks

- How do “disturbances” spread in a network?
  - Spreading of epidemics, rumors, fads, ...
  - Routing of messages; returnability

- How to identify “community structures” in a network?
  - Clustering, transitivity
  - Partitioning
Subgraph centrality

There are dozens of different definitions of centrality for nodes in a graph. The simplest is degree centrality, which is just the degree $d_i$ of node $i$. This does not take into account the “importance” of the nodes a given nodes is connected to—only their number.

Subgraph centrality (Estrada & Rodríguez-Velásquez, 2005) measures the centrality of a node by taking into account the number of subgraphs the node “participates” in.

This is done by counting, for all $k = 1, 2, \ldots$ the number of closed walks in $G$ starting and ending at node $i$, with longer walks being penalized (given a smaller weight).
Subgraph centrality

Recall that

- \((A^k)_{ii}\) = \# of closed walks of length \(k\) based at node \(i\),
- \((A^k)_{ij}\) = \# of walks of length \(k\) that connect nodes \(i\) and \(j\).

Using \(1/k!\) as weights leads to the notion of subgraph centrality:

\[
SC(i) = \left[ I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots \right]_{ii} = [e^A]_{ii}
\]

Note: the additional term of order \(k = 0\) does not alter the ranking of nodes in terms of subgraph centrality.
Subgraph centrality

Of course different weights can be used, leading to different matrix functions, such as the resolvent (Katz, 1953):

\[(I - cA)^{-1} = I + cA + c^2 A^2 + \cdots , \quad 0 < c < 1/\lambda_N.\]

Subgraph centrality has been used successfully in various settings, especially proteomics and neuroscience.

Also, in the case of a directed network one can use the solution vectors of the linear systems

\[(I - cA)x = 1 \quad \text{and} \quad (I - cA^T)y = 1\]

to rank hubs and authorities.
Other matrix functions of interest are \( \cosh(A) \) and \( \sinh(A) \), which correspond to considering only walks of even and odd length, respectively.

Because in a bipartite graph \( \sinh(A) = 0 \), the quantity

\[
\langle B(G) \rangle := \frac{\text{Tr}(\cosh(A))}{\text{Tr}(e^A)}
\]

provides a measure of how “close” a graph is to being bipartite.

Hyperbolic matrix functions are also used to define the notion of \textit{returnability} in digraphs (Estrada & Hatano, 2009).
Communicability measures how “easy” it is to send a message from node $i$ to node $j$ in a graph. It is defined as (Estrada & Hatano, 2008):

$$C(i,j) = [e^A]_{ij}$$

$$= \left[ I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots \right]_{ij}$$

$$\approx \text{weighted sum of walks joining nodes } i \text{ and } j.$$

As before, other power series expansions (weights) can be used, leading to different functions of $A$.

For a large graph, computing all communicabilities $C(i,j)$ ($i \neq j$) is prohibitively expensive. Instead, averages are often used—as in statistical physics.
Communicability

The average communicability of a node is defined as

$$\langle C(i) \rangle := \frac{1}{N-1} \sum_{j \neq i} C(i,j) = \frac{1}{N-1} \sum_{j \neq i} [e^A]_{ij}. $$

Communicability functions can be used to study the spreading of diseases (or rumors) and to identify bottlenecks in networks.

Recently, community detection algorithms based on various communicability measures have been developed (Fortunato, 2010; Ma, Gao & Yong, 2010; Estrada, 2011).
We saw that many problems in graph analysis lead to the computation of selected entries of $f(A)$, where $A$ is the adjacency matrix and $f(x)$ is an analytic function.

Typically, the diagonal entries of $f(A)$ are wanted, and for large graphs some global averages over subsets of entries of $f(A)$, often expressed in terms of traces.

High accuracy is not always required. For example, in the case of centralities it is important to be able to identify just the top-ranked nodes. For the communicabilities, bounds are often sufficient.
All of these computations can be formulated as evaluation of bilinear forms $u^T f(A) v$ for suitably chosen vectors $u, v \in \mathbb{R}^N$ and for suitable functions $f(x)$:

- **Subgraph centrality:** $SC(i) = e_i^T \exp(A) e_i$
- **Communicability:** $C(i,j) = e_i^T \exp(A) e_j$
- **Average communicability:**

$$
\langle C(i) \rangle = \frac{1}{N-1} \left[ e_i^T \exp(A) e_i - e_i^T \exp(A) e_i \right]
$$

where $e$ is the vector of all ones.

How do we compute these quantities?
Consider the spectral decompositions

$$A = Q\Lambda Q^T, \quad f(A) = Qf(\Lambda)Q^T.$$  

For $u, v \in \mathbb{R}^N$ we have

$$u^T f(A)v = u^T Qf(\Lambda)Q^Tv = p^T f(\Lambda)q = \sum_{i=1}^{N} f(\lambda_i)p_iq_i,$$

where $p = Q^Tu$ and $q = Q^Tv$. Rewrite this as a Riemann-Stieltjes integral:

$$u^T f(A)v = \int_a^b f(\lambda) d\mu(\lambda), \quad \mu(\lambda) = \begin{cases} 
0 & \lambda < a = \lambda_1 \\
\sum_{j=1}^{i} p_j q_j & \lambda_i \leq \lambda < \lambda_{i+1} \\
\sum_{j=1}^{N} p_j q_j & b = \lambda_N \leq \lambda.
\end{cases}$$
The general Gauss-type quadrature rule gives in this case:

\[
\int_a^b f(\lambda)d\mu(\lambda) = \sum_{j=1}^{\frac{n}{2}} w_j f(t_j) + \sum_{k=1}^{M} v_k f(z_k) + R[f],
\]

where the nodes \( \{z_k\} \) are prescribed.

- Gauss: \( M = 0 \),
- Gauss–Radau: \( M = 1 \), \( z_1 = a \) or \( z_2 = b \),
- Gauss–Lobatto: \( M = 2 \), \( z_1 = a \) and \( z_2 = b \).

The evaluation of these quadrature rules is reduced to

- computation of orthogonal polynomials via three-term recurrence,
- or, equivalently, computation of entries and spectral information of the corresponding tridiagonal matrix (Lanczos).
The tridiagonal matrix $J_n$ corresponds to the three-term recurrence relationship satisfied by the set of polynomials orthonormal with respect to $d\mu$.

\[
J_n = \begin{pmatrix}
\omega_1 & \gamma_1 \\
\gamma_1 & \omega_2 & \gamma_2 \\
& \ddots & \ddots & \ddots \\
& & \gamma_{n-2} & \omega_{n-1} & \gamma_{n-1} \\
& & & \gamma_{n-1} & \omega_n
\end{pmatrix}
\]

The eigenvalues of $J_n$ are the Gauss nodes, whereas the Gauss weights are given by the squares of the first entries of the normalized eigenvectors of $J_n$.

The quadrature rule is computed with the Golub–Welsch QR algorithm.
Consider the case $u = v = e_i$ (corresp. to the $(i, i)$ entry of $f(A)$).

The entries of $J_n$ are computed using the symmetric Lanczos algorithm:

$$\gamma_j x_j = r_j = (A - \omega_j I)x_{j-1} - \gamma_{j-1}x_{j-2}, \quad j = 1, 2, \ldots$$

$$\omega_j = x_{j-1}^T A x_{j-1},$$

$$\gamma_j = \|r_j\|$$

with initial vectors $x_{-1} = 0$ and $x_0 = e_i$.

Each additional Lanczos step amounts to adding another node to the Gauss-type quadrature rule, resulting in tighter and tighter bounds (implementation based on MMQ Matlab toolbox by G. Meurant).
For $f(A) = e^A$ and $f(A) = (I - cA)^{-1}$ we obtain:

- bounds on $[f(A)]_{ii}$ from symmetric Lanczos,
- bounds on $[f(A)]_{ii} + [f(A)]_{ij}$ from unsymmetric Lanczos,
- lower bounds from the Gauss and the Gauss-Radau rules,
- upper bounds from the Gauss-Radau and Gauss-Lobatto rules.

In computations we often use the simple (Gerschgorin) estimates

$$
\lambda_{\text{min}} \approx - \max_{1 \leq i \leq N} \{\deg(i)\}, \quad \lambda_{\text{max}} \approx \max_{1 \leq i \leq N} \{\deg(i)\}.
$$

Using more accurate estimates of the extreme eigenvalues of $A$ generally leads to improved results, especially in scale-free case.
Numerical experiments were performed on a number of networks, both synthetic (generated with the CONTEST Toolbox for Matlab, Taylor & Higham 2010) and from real-world applications, including PPI and transportation networks.

- A few (3-5) Lanczos steps per estimate are usually enough
- Independent of $N$ for graphs with bounded max degree
- Almost as good for power-law degree distributions
- Very efficient approach for computing global averages
Example: Range dependent network from CONTEST, \( N = 100 \).

- MMQ approximations of the Estrada index \( EE(G) = 425.0661 \)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gauss</td>
<td>348.9706</td>
<td>416.3091</td>
<td>424.4671</td>
<td>425.0413</td>
<td>425.0655</td>
</tr>
<tr>
<td>Radau (lower)</td>
<td>378.9460</td>
<td>420.6532</td>
<td>424.8102</td>
<td>425.0570</td>
<td>425.0659</td>
</tr>
<tr>
<td>Radau (upper)</td>
<td>652.8555</td>
<td>437.7018</td>
<td>425.6054</td>
<td>425.0828</td>
<td>425.0664</td>
</tr>
<tr>
<td>Lobatto</td>
<td>2117.9233</td>
<td>531.1509</td>
<td>430.3970</td>
<td>425.2707</td>
<td>425.0718</td>
</tr>
</tbody>
</table>

- MMQ approximations of \([e^A]_{1,5} = 0.396425\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radau (lower)</td>
<td>(-2.37728)</td>
<td>0.213316</td>
<td>0.388791</td>
<td>0.396141</td>
<td>0.396431</td>
</tr>
<tr>
<td>Radau (upper)</td>
<td>4.35461</td>
<td>0.595155</td>
<td>0.404905</td>
<td>0.396626</td>
<td>0.396420</td>
</tr>
</tbody>
</table>
Bounds for the Estrada index of small world matrices of increasing size. Watts–Strogatz model with parameters \((4, 0.1)\); five Lanczos iterations.

<table>
<thead>
<tr>
<th>(N)</th>
<th>Gauss</th>
<th>Gauss-Radau (l)</th>
<th>Gauss-Radau (u)</th>
<th>Gauss-Lobatto</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.64e5</td>
<td>1.63e5</td>
<td>1.64e5</td>
<td>1.66e5</td>
</tr>
<tr>
<td>2500</td>
<td>4.09e5</td>
<td>4.10e5</td>
<td>4.11e5</td>
<td>4.12e5</td>
</tr>
<tr>
<td>5000</td>
<td>8.18e5</td>
<td>8.18e5</td>
<td>8.20e5</td>
<td>8.33e5</td>
</tr>
<tr>
<td>7500</td>
<td>1.23e6</td>
<td>1.22e6</td>
<td>1.22e6</td>
<td>1.25e6</td>
</tr>
<tr>
<td>10000</td>
<td>1.63e6</td>
<td>1.63e6</td>
<td>1.64e6</td>
<td>1.66e6</td>
</tr>
</tbody>
</table>

Five iterations suffice to reach a small relative error, independent of \(N\).
Experiments were performed on scale-free networks (Barabási–Albert model) of increasing size, to see the effect of increasing maximum degree.

In this case, the Gerschgorin estimate $\lambda_{\text{max}} \approx d_{\text{max}}$ gives poor results. Indeed, whereas the max degree increases with $N$, the spread of the eigenvalues tend to grow very slowly.

A better strategy is to estimate $\lambda_{\text{min}}$, $\lambda_{\text{max}}$ using Lanczos.

The number of Lanczos steps per estimate is then approximately independent of $N$, for a fixed accuracy. Typically, 9-11 Lanczos steps suffice to achieve relative errors around $10^{-6}$.
Some challenges

- Relate convergence rates of Gaussian quadrature rules to network properties
- Can one quickly locate subsets of nodes with the largest centralities \( SC(i) \)?
- For truly massive networks, current approaches are too expensive
  - Fast estimates for \( \text{Tr}(f(A)) \): randomized algorithms? (Avron & Toledo)
  - Low-rank approximations?
  - Exploitation of locality?
- Parallelization is tricky
- Major issue in this area: how to evaluate and compare measures?
Some classic early references:

The field exploded in the late 1990s due to several breakthroughs by physicists, applied mathematicians and computer scientists. This helped make the field more quantitative and mathematically sophisticated. Landmark papers include

Recent systematic accounts include