Updating and Downdating Techniques for optimizing Network Communicability.

Francesca Arrigo and Michele Benzi

Università degli Studi dell’Insubria (Como, Italy)
Emory University (Atlanta, USA)

Two days on Applied Mathematics in Cagliari. April 9–10, 2015
Motivation

There is no easy way to describe complex networks. Several indices and measures have been introduced to understand how they behave.

Total communicability (TC):
quantifies the ease of spreading information across the network and how well connected a network is.

We want to manipulate the edges in the network in order to tune the TC.
Background

Let $G = (V, E)$ be a complex network with $n = |V|$ nodes and $m = |E|$ edges. Suppose that:

- $G$ is unweighted;
- $G$ has no multiple edges nor self loops;
- $G$ is undirected;
- $G$ is connected;
- $m = O(n)$.

The corresponding adjacency matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ will be:

- binary with $a_{ii} = 0$ for all $i \in V$;
- symmetric;
- irreducible;
- sparse.
Let $G = (V, E)$ be a complex network with $n = |V|$ nodes and $m = |E|$ edges. Suppose that:

- $G$ is unweighted;
- $G$ has no multiple edges nor self loops;
- $G$ is undirected;
- $G$ is connected;
- $m = O(n)$.

The corresponding adjacency matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ will be:

- binary with $a_{ii} = 0$ for all $i \in V$;
- symmetric;
- irreducible;
- sparse.
A few useful definitions

- **walk** of length $k$:
  \[ \{i_1, i_2, \ldots, i_{k+1} \in V | (i_l, i_{l+1}) \in E \text{ for all } 1 \leq l \leq k \} \]

- **closed walk** of length $k$: a walk for which $i_1 = i_{k+1}$.

**Remark:**
The quantities $(A^k)_{ii}$, $(A^k)_{ij}$ count (closed) walks of length $k$.

- **degree** of node $i$:

  \[ d_i = |\{ j \in V : (i, j) \in E \}| = (A^2)_{ii}. \]
The total communicability

(Benzi & Klymko, 2013)

**Total communicability:** quantifies the ease of spreading information across the network and how well connected a network is.

\[
TC(A) = 1^T e^A 1 = \sum_{k=0}^{\infty} \frac{1^T A^k 1}{k!}
\]

This index can be possibly normalized by the number of nodes \( n \) (or of edges \( m \)).

Bounds for the *normalized* TC:

\[
\frac{1}{n} \sum_{i=1}^{n} (e^A)_{ii} \leq \frac{TC(A)}{n} \leq e^{\lambda_1}.
\]
Our goal

We want to modify the edges in the graph in order to tune the TC. We need to develop methods to select which modifications we need to perform to achieve our goals.

For practical purposes: we do not want to change too much our network!!
Problem setting

We develop techniques aimed at tackling the following problems.

- **Downdate**: select $K$ edges that can be downdated from the network that cause the smallest drop in $TC(A)$;
- **Update**: select $K$ edges to be added to the network so as to increase as much as possible the total communicability of the graph;
- **Rewire**: select $K$ edges to be rewired in the network so as to increase as much as possible the value of $TC(A)$.

We need to “rank” the edges/virtual edges (or some of them) in order to make the (as close as possible to) optimal choice.
Main idea

Edge connecting important nodes are themselves important!

- Update: update the virtual edge that connects two central nodes;
- Downdate: downdate the edge connecting two peripheral nodes.

All our definitions are of the form:

$$eC(i, j) = C(i) \cdot C(j), \quad \forall i, j \in V$$

where $C(\cdot) : V \to \mathbb{R}$ is a centrality measure for nodes.
Node centrality measures

The centrality measures we use are all walk–based:

1. *eigenvector centrality*:

   \[ EC(i) = q_1(i); \]

2. *subgraph centrality*:

   \[ SC(i) = (e^A)_{ii} = \sum_{k=0}^{\infty} \frac{(A^k)_{ii}}{k!}; \]

3. *total communicability*:

   \[ TC(i) = (e^A1)_i = \sum_{k=0}^{\infty} \frac{(A^k1)_i}{k!}. \]
The role of the spectral gap

Recall that we are interested in rankings rather than in the actual values of the centralities.

It can be shown that, when $\lambda_1 - \lambda_2$ is “large enough”, then:

$$SC(i) \approx e^{\lambda_1} q_1(i)^2$$

$$TC(i) \approx e^{\lambda_1} \| q_1 \|_1 q_1(i)$$

We expect agreement among the rankings derived from the three edge centrality measures when $\lambda_1 \gg \lambda_2$. 
Algorithms

(i) eigenvector(.no): selects the modification using $^e EC$;
(ii) subgraph(.no): uses the $^e SC$ to select the modification;
(iii) nodeTC(.no): uses the $^e TC$ to select the (virtual) edge;
(iv) degree: selects the modification using the values of $d_i + d_j$.

To tackle these problems our strategy is:

- Downdate: min;
- Update: max;
- Rewire: min $\rightarrow$ max.
(i) eigenvector(.no): selects the modification using $^eEC$;
(ii) subgraph(.no): uses the $^eSC$ to select the modification;
(iii) nodeTC(.no): uses the $^eTC$ to select the (virtual) edge;
(iv) degree: selects the modification using the values of $d_i + d_j$.

To tackle these problems our strategy is:

- Downdate: min;
- Update: max;
- Rewire: min $\rightarrow$ max.
Why degree?

This heuristic selects the modifications using $d_i + d_j$.

It has two main justifications:

- it is exact when we approximate $e^{A \pm (e_i e_j^T + e_j e_i^T)}$ with its second order Maclaurin expansion in the expression for TC;
- “$d_i + d_j$” appears in the bounds via quadrature rules for the normalized TC when $A$ is replaced by $A \pm (e_i e_j^T + e_j e_i^T)$. 
Bounds via quadrature rules – general

We can derive bounds for $u^T f(A)u$ via the Gauss-Radau quadrature rule

$$
\Phi \left( b, \omega_0 + \frac{\gamma_0^2}{\omega_0 - b} \right) \leq x_0^T f(A)x_0 \leq \Phi \left( a, \omega_0 + \frac{\gamma_0^2}{\omega_0 - a} \right) \quad (1)
$$

where $f$ is such that $f^{(2j+1)}(x) < 0$ for all $x \in [a, b]$, with $[a, b]$ interval containing the spectrum of $A$, and where

$$
J_2 = \begin{pmatrix}
\omega_0 & \gamma_0 \\
\gamma_0 & * \\
\end{pmatrix}
$$

is derived via one step of the Lanczos algorithm with starting vectors $x_{-1} = \mathbf{0}$ and $x_0 = \frac{u}{\|u\|}$.

Bounds for the normalized TC

In this framework we need to use $f(x) = e^{-x}$ (and therefore to work on $-A$) and

$$\Phi(x, y) = \frac{c(e^{-x} - e^{-y}) + xe^{-y} - ye^{-x}}{x - y}, \quad c = \omega_0.$$  

The values of the variables are

$$\begin{cases} a \approx -\lambda_1 \\ b \approx -\lambda_n \\ \omega_0 = -\mu = -\frac{1}{n} \sum_{k=1}^{n} d_k \\ \gamma_0 = \sigma = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (d_k - \mu)^2} \end{cases}$$
How do the bounds change?

Downdate $A \rightarrow A - (e_i e_j^T + e_j e_i^T)$:

$$
\begin{aligned}
    a_- &= a; \\
    b_- &= b - 1; \\
    \omega_- &= \omega_0 + \frac{2}{n}; \\
    \gamma_- &= \left[ \gamma_0^2 - \frac{2}{n} (d_i + d_j - 1 + 2\omega_0 + \frac{2}{n}) \right]^{\frac{1}{2}}; \\
    c &= \omega_-. \\
\end{aligned}
$$

Update $A \rightarrow A + (e_i e_j^T + e_j e_i^T)$:

$$
\begin{aligned}
    a_+ &= a - 1; \\
    b_+ &= b; \\
    \omega_+ &= \omega_0 - \frac{2}{n}; \\
    \gamma_+ &= \left[ \gamma_0^2 + \frac{2}{n} (d_i + d_j + 1 + 2\omega_0 - \frac{2}{n}) \right]^{\frac{1}{2}}; \\
    c &= \omega_. \\
\end{aligned}
$$
About the algorithms..

**Update:** too many virtual edges ($O(n^2)$). We select the modification among the virtual edges in a suitable subgraph of $G$.

**Downdate:** for large networks, $m$ may be very large. Restriction to a suitable subgraph is advantageous (timings).

We DO NOT NEED to compute the total communicability!!!
About the algorithms..

Update: too many virtual edges \(O(n^2)\). We select the modification among the virtual edges in a suitable subgraph of \(G\).

Downdate: for large networks, \(m\) may be very large. Restriction to a suitable subgraph is advantageous (timings).

We DO NOT NEED to compute the total communicability!!!
## Dataset

<table>
<thead>
<tr>
<th>NAME</th>
<th>$n$</th>
<th>$m$</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_1 - \lambda_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca–HepTh</td>
<td>8638</td>
<td>24806</td>
<td>31.034</td>
<td>23.004</td>
<td>8.031</td>
</tr>
<tr>
<td>as–22july06</td>
<td>22963</td>
<td>48436</td>
<td>71.613</td>
<td>53.166</td>
<td>18.447</td>
</tr>
<tr>
<td>usroad–48</td>
<td>126146</td>
<td>161950</td>
<td>3.911</td>
<td>3.840</td>
<td>0.071</td>
</tr>
</tbody>
</table>

nz = 49612 nz = 96872 nz = 323900
## Dataset

<table>
<thead>
<tr>
<th>NAME</th>
<th>$n$</th>
<th>$m$</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_1 - \lambda_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca–HepTh</td>
<td>8638</td>
<td>24806</td>
<td>31.034</td>
<td>23.004</td>
<td>8.031</td>
</tr>
<tr>
<td>as–22july06</td>
<td>22963</td>
<td>48436</td>
<td>71.613</td>
<td>53.166</td>
<td>18.447</td>
</tr>
<tr>
<td>usroad–48</td>
<td>126146</td>
<td>161950</td>
<td>3.911</td>
<td>3.840</td>
<td>0.071</td>
</tr>
</tbody>
</table>

![Graphs of datasets](image)
DOWN: Results & Timings (in sec.)

<table>
<thead>
<tr>
<th></th>
<th>ca–HepTh</th>
<th>as–22july06</th>
<th>usroad–48</th>
</tr>
</thead>
<tbody>
<tr>
<td>eigenvector</td>
<td>278.13</td>
<td>599.83</td>
<td>11207.39</td>
</tr>
<tr>
<td>nodeTC</td>
<td>553.04</td>
<td>1234.49</td>
<td>2634.27</td>
</tr>
<tr>
<td>degree</td>
<td>29.67</td>
<td>53.42</td>
<td>153.52</td>
</tr>
<tr>
<td>eigenvector.no</td>
<td>0.07</td>
<td>1.79</td>
<td>4.08</td>
</tr>
<tr>
<td>nodeTC.no</td>
<td>0.34</td>
<td>0.83</td>
<td>1.34</td>
</tr>
<tr>
<td>subgraph.no</td>
<td>246.44</td>
<td>808.24</td>
<td>10322.41</td>
</tr>
<tr>
<td>subgraph.no*</td>
<td>107.36</td>
<td>383.34</td>
<td>1774.07</td>
</tr>
</tbody>
</table>

subgraph.no* works on a subset of $E$. 
DOWN: Results & Timings (in sec.)

<table>
<thead>
<tr>
<th></th>
<th>ca–HepTh</th>
<th>as–22july06</th>
<th>usroad–48</th>
</tr>
</thead>
<tbody>
<tr>
<td>eigenvector</td>
<td>278.13</td>
<td>599.83</td>
<td>11207.39</td>
</tr>
<tr>
<td>nodeTC</td>
<td>553.04</td>
<td>1234.49</td>
<td>2634.27</td>
</tr>
<tr>
<td>degree</td>
<td>29.67</td>
<td>53.42</td>
<td>153.52</td>
</tr>
<tr>
<td>eigenvector.no</td>
<td>0.07</td>
<td>1.79</td>
<td>4.08</td>
</tr>
<tr>
<td>nodeTC.no</td>
<td>0.34</td>
<td>0.83</td>
<td>1.34</td>
</tr>
<tr>
<td>subgraph.no</td>
<td>246.44</td>
<td>808.24</td>
<td>10322.41</td>
</tr>
<tr>
<td>subgraph.no*</td>
<td>107.36</td>
<td>383.34</td>
<td>1774.07</td>
</tr>
</tbody>
</table>

*subgraph.no* works on a subset of $E$. 
UP: Results & Timings (in sec.)

<table>
<thead>
<tr>
<th></th>
<th>ca–HepTh</th>
<th>as–22july06</th>
<th>usroad–48</th>
</tr>
</thead>
<tbody>
<tr>
<td>eigenvector</td>
<td>192.8</td>
<td>436.9</td>
<td>1599.5</td>
</tr>
<tr>
<td>nodeTC</td>
<td>561.94</td>
<td>1218.77</td>
<td>2932.01</td>
</tr>
<tr>
<td>degree</td>
<td>11.1</td>
<td>12.4</td>
<td>175.8</td>
</tr>
<tr>
<td>eigenvector.no</td>
<td>0.19</td>
<td>0.33</td>
<td>5.85</td>
</tr>
<tr>
<td>nodeTC.no</td>
<td>0.30</td>
<td>0.55</td>
<td>1.59</td>
</tr>
<tr>
<td>subgraph.no</td>
<td>10.6</td>
<td>74.1</td>
<td>651.6</td>
</tr>
</tbody>
</table>
UP: Results & Timings (in sec.)

<table>
<thead>
<tr>
<th></th>
<th>ca–HepTh</th>
<th>as–22july06</th>
<th>usroad–48</th>
</tr>
</thead>
<tbody>
<tr>
<td>eigenvector</td>
<td>192.8</td>
<td>436.9</td>
<td>1599.5</td>
</tr>
<tr>
<td>nodeTC</td>
<td>561.94</td>
<td>1218.77</td>
<td>2932.01</td>
</tr>
<tr>
<td>degree</td>
<td>11.1</td>
<td>12.4</td>
<td>175.8</td>
</tr>
<tr>
<td>eigenvector.no</td>
<td>0.19</td>
<td>0.33</td>
<td>5.85</td>
</tr>
<tr>
<td>nodeTC.no</td>
<td>0.30</td>
<td>0.55</td>
<td>1.59</td>
</tr>
<tr>
<td>subgraph.no</td>
<td>10.6</td>
<td>74.1</td>
<td>651.6</td>
</tr>
</tbody>
</table>
We test seven synthetic networks of increasing size (from 1000 to 7000) corresponding to Barabási–Albert scale-free graphs. The average degree of nodes is fixed: 5; Fixed number of modifications: 500.

- dominant eigenpair: `eigs` (Matlab built-in function);
- diagonal entries of $e^A$: `mmq` by G. Meurant (5 steps of Lanczos iteration);
- entries of $e^A\mathbf{1}$: `funm_kryl` by S. Güttel (Krylov method).
Timings in scale-free graphs $n = 1000, 2000, \ldots, 7000$
The natural connectivity

(Jun et al., 2010)

The *natural connectivity* is a measure of connectivity based on an intuitive notion of robustness: the existence of “redundant” routes makes a network more robust. Equivalently, a network is more robust if each of its nodes is involved in a lot of closed walks.

**Definition**

*The natural connectivity of the graph is defined as*

\[
\bar{\lambda} = \ln \left( \frac{EE}{n} \right) = \ln(EE) - \ln(n)
\]

where \( EE = EE(G, 1) := \sum_{j=1}^{n} e^{\lambda_j} \) is the Estrada Index of the network.
A statistical-mechanical approach

(Estrada & Hatano, 2007)

Let $\beta > 1$ be the “strength” of the connections between nodes. The adjacency matrix of the new network is $\beta A$.

From the standpoint of quantum statistical mechanics, $\mathcal{H} = -A$ is the Hamiltonian and $\beta = \frac{1}{k_B T}$ is the inverse temperature. The Estrada index

$$Z(G, \beta) = EE(G, \beta) = \text{Tr}(e^{\beta A})$$

can be seen as the partition function of the corresponding complex network.
A statistical-mechanical approach

(Estrada & Hatano, 2007)

The eigenvalues give the energy levels, each corresponding to a different state of the system. The Maxwell–Boltzmann distribution gives the probability that the system is found in a particular state $i$:

$$p_i = \frac{e^{\beta \lambda_i}}{EE(G, \beta)} = \frac{e^{\beta \lambda_i}}{\text{Tr}(e^{\beta A})}.$$  

The Gibbs entropy of the network is then defined as

$$S(G, \beta) = -k_B \sum_{i=1}^n p_i \ln(p_i) = -k_B \beta \sum_i \lambda_i p_i + k_B \ln(EE(G, \beta)).$$
The free energy (aka: the natural connectivity)

Using the equation that relates the *Helmholtz free energy* $F(G, \beta)$ with the *Gibbs entropy* $S(G, \beta)$ and the *total energy* $H(G, \beta)$:

$$ F(G, \beta) = H(G, \beta) - TS(G, \beta), $$

one gets

$$ \begin{cases} 
  H(G, \beta) = - \sum_i \lambda_i p_i \\
  F(G, \beta) = -\beta^{-1} \ln(EE(G, \beta)) 
\end{cases} . $$

By taking $\beta = 1$ in the second equation one finds that

$$ F := F(G, 1) = -\bar{\lambda} + \ln(n). $$
In 2014, Chan et al. proposed an algorithm to controllably modify the natural connectivity. This algorithm selects $k$ edges to be updated by maximizing at each step the quantity

$$c = e^{\lambda_1} \left( e^{2q_1(i)q_1(j)} + \sum_{h=2}^{t} e^{\lambda_h - \lambda_1} e^{2q_h(i)q_h(j)} \right)$$

and then updating the top $t = 50$ eigenpairs using the following rules:

$$\begin{cases} 
\lambda_k = \lambda_k + 2q_k(i)q_k(j) \\
q_k = q_k + \sum_{h=1, h\neq k}^{n} \left( \frac{q_h(i)q_k(j) - q_h(j)q_k(i)}{\lambda_k - \lambda_h} \right) q_h 
\end{cases} \quad k = 1, 2, \ldots, t$$

where $(i, j)$ is the selected update.
Figure: Evolution of the natural connectivity and of the normalized total communicability (in a semi–logarithmic scale plot) when 500 updates are performed on four large networks.
Conclusions

What I did show you:

- All the methods we have introduced work effectively and return excellent results (w/ or w/o the recomputation of rankings);
- the TC is a meaningful measure of network connectivity, as it evolves as the natural connectivity (i.e., the Helmholtz free energy).

What I did NOT show you:

- Our techniques perform as well as the locally optimal method;
- our heuristics are efficient and effective for the Rewire problem as well;
- it is fundamental to use edge centrality measures.

-Thank you!
Conclusions

What I did show you:

• All the methods we have introduced work effectively and return excellent results (w/ or w/o the recomputation of rankings);
• the TC is a meaningful measure of network connectivity, as it evolves as the natural connectivity (i.e., the Helmholtz free energy).

What I did NOT show you:

• Our techniques perform as well as the locally optimal method;
• our heuristics are efficient and effective for the Rewire problem as well;
• it is fundamental to use edge centrality measures.

- -Thank you!
Conclusions

What I did show you:

• All the methods we have introduced work effectively and return excellent results (w/ or w/o the recomputation of rankings);

• the TC is a meaningful measure of network connectivity, as it evolves as the natural connectivity (i.e., the Helmholtz free energy).

What I did NOT show you:

• Our techniques perform as well as the locally optimal method;
• our heuristics are efficient and effective for the Rewire problem as well;
• it is fundamental to use edge centrality measures.

- -Thank you!
Bibliography

F. Arrigo & M. Benzi, “Updating and downdating techniques for optimizing network robustness”, submitted;

M. Benzi & P. Boito, “Quadrature rule-based bounds for functions of adjacency matrices”, 2010;

M. Benzi & C. Klymko, “Total communicability as a centrality measure”, 2013;

E. Estrada & N. Hatano, “Communicability in complex networks”, 2008;


C. Fenu et al., “Network analysis via partial spectral factorization and Gauss Quadrature”, 2012;
Bibliography (cont.)

S. Güttel, *funm_kryl* toolbox for Matlab;

G. Meurant, *MMQ* toolbox for Matlab;

A. Taylor & D. J. Higham, *CONTEST: Toolbox files and documentation*;

T. Davis & Y. Hu, *The University of Florida Sparse Matrix Collection*;

W. Jun et al., “*Natural connectivity of complex networks*”, 2010;

H. Chan et al., “*Make it or Break it: manipulating robustness in large networks*”, 2014;