To be connected, or not to be connected...
That is the **Minimum Inefficiency Subgraph** Problem

Natali Ruchansky
Francesco Bonchi
David Garcia-Soriano
Francesco Gullo
Nicolas Kourtellis
Biologists in Lab X have constructed a large protein-protein interaction network (PPI).
Biologists in Lab X have constructed a large protein-protein interaction network (PPI).

The PI has tasked them with making an amazing discovery about relationship among specific proteins P1, P2, and P3.
Given a set of subjects in a terrorist network suspected of organizing an attack. Which other subjects, likely to be involved, should we keep under control?
Given a set of users who clicked on an ad, who else should the ad be displayed to?
Given a set of patients infected with a viral disease, which other people should we monitor?
Community search / seed set expansion

• General class of problems of the form:

  Given a graph $G=(V,E)$ and a set of vertices $Q \subset V$, find a subgraph $H$ of $G$ that “explains” the connections among $Q$. ($H$ minimizes/maximizes some objective function)

• Several approaches in the literature
  – $H$ must be a connected subgraph
  – Mostly based on random-walks
  – Tend to return rather large solutions
  – Solutions get very large when query nodes belong to different communities
  – Have parameters
The Minimum Wiener Connector Problem
(SIGMOD 2015)

Our proposal: find the connected subgraph $H$ containing $Q$ and minimizing the Wiener Index (the sum of pairwise distances)

$$H^* = \arg \min_{G[S]: Q \subseteq S \subseteq V} \sum_{\{u, v\} \in S} d_{G[S]}(u, v)$$

- Parameter-free
- Returns smaller and denser subgraphs
  No matter whether the query nodes belong to the same community or not
- Add “important” nodes (high centrality)
- Efficient algorithm with approximation guarantees
Smaller, denser, and more central vertices

<table>
<thead>
<tr>
<th></th>
<th>email</th>
<th>yeast</th>
<th>oregon</th>
<th>astro</th>
<th>dblp</th>
<th>youtube</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>V(H)</td>
<td>$</td>
<td>671</td>
<td>819</td>
<td>9028</td>
<td>12758</td>
</tr>
<tr>
<td>$</td>
<td>V(H)</td>
<td>$</td>
<td>155</td>
<td>188</td>
<td>4556</td>
<td>1735</td>
</tr>
<tr>
<td>$</td>
<td>V(H)</td>
<td>$</td>
<td>137</td>
<td>100</td>
<td>1846</td>
<td>598</td>
</tr>
<tr>
<td>$</td>
<td>V(H)</td>
<td>$</td>
<td>26</td>
<td>24</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>$</td>
<td>V(H)</td>
<td>$</td>
<td>24</td>
<td>24</td>
<td>23</td>
<td>23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>email</th>
<th>yeast</th>
<th>oregon</th>
<th>astro</th>
<th>dblp</th>
<th>youtube</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta(H)$</td>
<td>0.016</td>
<td>0.016</td>
<td>0.01</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>$\delta(H)$</td>
<td>0.047</td>
<td>0.028</td>
<td>0.02</td>
<td>0.019</td>
<td>0.01</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td>$\delta(H)$</td>
<td>0.029</td>
<td>0.039</td>
<td>0.02</td>
<td>0.07</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>$\delta(H)$</td>
<td>0.080</td>
<td>0.088</td>
<td>0.090</td>
<td>0.09</td>
<td>0.08</td>
<td>0.1</td>
</tr>
<tr>
<td>$\delta(H)$</td>
<td>0.093</td>
<td>0.091</td>
<td>0.106</td>
<td>0.13</td>
<td>0.11</td>
<td>0.13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>email</th>
<th>yeast</th>
<th>oregon</th>
<th>astro</th>
<th>dblp</th>
<th>youtube</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_c(H)$</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td>$\beta_c(H)$</td>
<td>0.03</td>
<td>0.02</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td>$\beta_c(H)$</td>
<td>0.03</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
<td>0.02</td>
<td>&lt;0.01</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td>$\beta_c(H)$</td>
<td>0.09</td>
<td>0.07</td>
<td>0.10</td>
<td>0.11</td>
<td>0.10</td>
<td>0.13</td>
</tr>
<tr>
<td>$\beta_c(H)$</td>
<td>0.11</td>
<td>0.11</td>
<td>0.12</td>
<td>0.14</td>
<td>0.12</td>
<td>0.18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>email</th>
<th>yeast</th>
<th>oregon</th>
<th>astro</th>
<th>dblp</th>
<th>youtube</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W(H)$</td>
<td>$\approx 750k$</td>
<td>$\approx 2M$</td>
<td>$\approx 137M$</td>
<td>$\approx 292M$</td>
<td>$\approx 400M$</td>
<td>$\approx 1.5G$</td>
</tr>
<tr>
<td>$W(H)$</td>
<td>54,598</td>
<td>69,296</td>
<td>$\approx 50M$</td>
<td>$\approx 8.3M$</td>
<td>$\approx 12.6M$</td>
<td>$\approx 561M$</td>
</tr>
<tr>
<td>$W(H)$</td>
<td>52,222</td>
<td>15,838</td>
<td>$\approx 7.5M$</td>
<td>40,079</td>
<td>$\approx 1.2M$</td>
<td>$\approx 1.3M$</td>
</tr>
<tr>
<td>$W(H)$</td>
<td>1,200</td>
<td>1,259</td>
<td>1,164</td>
<td>1,318</td>
<td>3,371</td>
<td>1,324</td>
</tr>
<tr>
<td>$W(H)$</td>
<td>968</td>
<td>931</td>
<td>923</td>
<td>1,007</td>
<td>2,043</td>
<td>956</td>
</tr>
</tbody>
</table>
Relaxing connectivity instead of forcing connectivity relax the constraint
Desired Properties

Parsimonious vertex addition
• vertices should be added iff they help forming a more **cohesive** subgraph

Outlier Tolerance
• query vertices which are far from others should remain disconnected

Multi-community awareness
• if the query vertices span multiple communities, connectedness should not be imposed among them
Cohesiveness

• As with the Wiener Connector, we leverage shortest path distances; however, the distance between disconnected vertices is infinite.

• Idea: use the reciprocal of the shortest-path distance! This has the useful property of handling disconnection neatly ($\infty^{-1} = 0$)

Network Efficiency (Latora and Marchiori):

$$E(G) = \frac{1}{|V|(|V|-1)} \sum_{u,v \in V, u \neq v} \frac{1}{d_{G}(u, v)}$$

Harmonic Centrality (Boldi and Vigna):

$$c(u) = \sum_{v \in V} \frac{1}{d_{G}(v, u)}$$
What about these problem statements?

Given a graph $G=(V,E)$ and a set of vertices $Q \subseteq V$, find a (not-necessarily connected) subgraph $H$ of $G$, with $Q \subseteq V(H)$ that maximizes network efficiency $E(H)$

Given a graph $G=(V,E)$ and a set of vertices $Q \subseteq V$, find a (not-necessarily connected) subgraph $H$ of $G$, with $Q \subseteq V(H)$ that maximizes the total harmonic centrality $C(H)$
These do not work...

C(G[Q]) = 0
E(G[Q]) = 0

C(H) = 9900
E(H) = 0.942

a clique of size 100
Minimize Network Inefficiency

Given a graph $G=(V,E)$, we define its inefficiency as:

$$
\mathcal{I}(G) = \sum_{u,v \in V, u \neq v} 1 - \frac{1}{d_G(v,u)}
$$

Note:

$$
\mathcal{E}(G) = \frac{C(G)}{n(n-1)}
$$

$$
\mathcal{I}(G) = n(n-1) - C(G)
$$
... and this works

A clique of size 100

C(G[Q])=0
E(G[Q])=0
I(G[Q])=6

C(G[Q])=0
E(G[Q])=0
I(G[Q])=12

C(G[Q])=9900
E(G[Q])=0.942
I(G[Q])=606
Problem statement and hardness

**Problem 1 (Min-Inefficiency-Subgraph).** Given an undirected graph $G = (V, E)$ and a query set $Q \subseteq V$, find

$$H^* = \arg\min_{G[S]:Q \subseteq S \subseteq V} I(G[S]).$$

**Theorem 4.1.** *Min-Inefficiency-Subgraph* is NP-hard, and it remains hard even on undirected graphs with diameter 3.
Greedy Algorithm

**Connect**
Start with the Minimum Wiener Connector for Q

**Remove**
Remove one vertex at a time until Q is disconnected

**Choose**
Choose the intermediate solution S that minimizes I(S)
Figure 1: Comparison on the Dolphins social network: query vertices are in blue, added vertices are in green.
The data is a graph where each vertex is an area of the brain and edges are added according to co-activation in experiments. (The graph is one connected component)

The 3 components in the solution end up corresponding to different functions: motor, visual, and emotional.

relaxing connectivity highlights three different functional relationships and gives a smaller, more interpretable solution
Brain Co-activation Network: competitors
Experimental Results

Parsimonious vertex addition
  • vertices should be added iff they help forming a more cohesive subgraph

Outlier Tolerance
  • query vertices which are far from others should remain disconnected

Multi-community awareness
  • if the query vertices span multiple communities, connectedness should not be imposed among them
Experimental Results

- Solution size vs. # query vertices
- # disconnected singletons in solution vs. # outliers selected
- # connected component in solution vs. # of communities spanned by Q
Cohesive meal creation

Minimum Inefficiency
- lemongrass
- scallop
- onion
- bell pepper
- mushroom
- beef
- black bean
- honey

MDL-based
- lemongrass
- scallop
- onion
- bell pepper
- mushroom
- soybean
- honey
- black bean

Bump Hunting
- lemongrass
- scallop
- onion
- bell pepper
- mushroom
- coffee
- peanut butter
- soybean
- honey
- black bean
Takeaway

how are we related?

but I don’t...

you love cats!

Selective Connector
Thanks!

@FrancescoBonchi
www.francescobonchi.com
francesco.bonchi@isi.it