Community Structure
Communities

• So far:
  – Properties of nodes: degree, centralities, triangles
  – Statistics of local properties: degree distribution

• How about large-scale organization?
  – Core-periphery, hierarchy,…
  – Communities
Introduction – Zachary’s karate club

• Social scientist Wayne W. Zachary collected friendship data at a karate club from 1970 to 72

• Edges represent friendship (activities outside the club)

• Conflict between instructor (node 1) and administrator (node 2), group broke up into two

• Q: Can we predict the groups based on network structure?

Zachary’s solution: network flow, source=node 1, sink=node 34

Cut: flow bottleneck

All but node 9 correct

Introduction – Zachary’s karate club

Introduction – Zachary’s karate club

• Zachary’s solution: network flow, source=node 1, sink=node 34
• Cut: flow bottleneck

All but node 9 correct
History

Early Communities
George Homans recorded the communication of bank tellers (top), identifying their communities (bottom) [2].

Graph Partitioning
Predecessors to community finding, graph partitioning algorithms optimize the layout of integrated circuits.

1927
The sociologist Stuart Rice uses voting patterns to identify communities in political bodies [4].

1940
Duncan R Luce and Albert D Peddy define communities as cliques [3].

1949
Robert Weiss and Eugene Jacobson identify communities by removing individuals linked to multiple groups [4].

1950
Brian Wilson Kerningam and Shen Lin develop a graph partitioning algorithm [18], widely used in chip design (BOX 9.1).

1955
Mark Granovetter explores the interplay between communities and weak ties [62].

1960
George Miller and his colleagues develop an algorithm for partitioning graphs into connected components [25].

1970
Wayne W. Zachary maps out the karate club, that a quarter of a century later becomes a test bed for community identification [7].

1973
Michelle Girvan and Mark Newman propose the hierarchical divisive algorithm, igniting an explosive interest in community identification [9]. They also introduce modularity in 2004 [23].

1977
Gary Flake, Steve Lawrence, and Lee Giles define a WWW community as documents that have more links to each other than to documents outside their community [15].

1980
Tamás Vicsek introduces the CFinder algorithm to identify overlapping communities [36].

2000
Erzsébet Ravasz proposes a hierarchical agglomerative algorithm, unleashing an explosion of research within systems biology [11].

Barabási, Network Science
Introduction – Zachary’s karate club

- Communities = locally dense subgraphs
- Modern network community detection from 2000s

First presenter to mention the ZKC at a conference gets the trophy.

http://networkkarate.tumblr.com/
Scientific collaboration at SFI

- Link (A – B) : A and B coauthored a paper
- Node classification by role according to position in community

Rat protein-protein interaction network

- Link (A – B) : A and B proteins physically interact
- Modules correspond to functions

Basics

• What is a community?
  - Intuitively: densely connected subnetworks

• Why is it interesting?
  - Nodes that participate in same function/nodes with similar attributes form groups and these groups are represented in network structure.

• Challenges:
  - No single clear definition. → Many competing options.
  - Large networks, different features. → Even more algorithms.
  - Which one to choose?
  - How to evaluate performance of method?

Outline

- How to identify communities?
- How to assess the quality of a community division?
- How assess the quality of a method?

- A lot of competing methods and measures
  - Here: selection that shows the development of the field
- And some guidelines to navigate the field
Method 1: Hierarchical Clustering
Hierarchical clustering

- Traditional method used by social scientists.
- 0) Define a distance metric $\sigma_{ij}$ between nodes based on network
  1) Each node in its own community.
  2) Calculate a distance between pairs of communities according to some rule.
  3) Join closest pair.
  4) Go to step 2.
Hierarchical clustering

- Traditional method used by social scientists.
- 0) Define a **distance metric** $\sigma_{ij}$ between nodes based on network
  1) Each node in its own community.
  2) Calculate a distance between pairs of communities according to **some rule**.
  3) Join closest pair.
  4) Go to step 2.
Hierarchical clustering: distance

- Node distance should be low if nodes are in a community.
- Popular choice: 
  \[ \sigma_{ij} = \frac{n_{ij}}{\sqrt{k_i k_j}} \]

- Other distances possible, e.g. number of independent paths connecting i and j

### Example

- \( k_i = 4 \)
- \( k_j = 5 \)
- \( n_{ij} = 3 \)

\[ \sigma_{ij} = \frac{3}{\sqrt{4 \cdot 20}} \approx 0.67 \]
Hierarchical clustering: distance

• Distance between communities with more than one node:

\[
x_{ij} = r_{ij} = \begin{bmatrix}
A & D & E & F & G \\
A & 2.75 & 2.22 & 3.46 & 3.08 \\
B & 3.38 & 2.68 & 3.97 & 3.40 \\
C & 2.31 & 1.59 & 2.88 & 2.34 \\
\end{bmatrix}
\]

- Single Linkage: \( x_{12} = 1.59 \)
- Complete Linkage: \( x_{12} = 3.97 \)
- Average Linkage: \( x_{12} = 2.84 \)
Hierarchical clustering: result

- Example artificial network
  (slightly different definition of distance)

Cut corresponds to one partition

dendrogram

Hierarchical clustering: result

- Dendrogram

A.-L. Barabási, *Network Science: Communities*.
Hierarchical clustering: application

- E. coli metabolism

- The color of each node, capturing the predominant biochemical class to which it belongs, indicates that different functional classes are segregated in distinct network neighborhoods.

- The highlighted region selects the nodes that belong to the pyrimidine metabolism, one of the predicted communities.

A.-L. Barabási, *Network Science: Communities*. 
Hierarchical clustering: issues

- Advantages:
  - Easy to understand
  - Easy to implement

- Disadvantages:
  - Slow(ish), number of steps to evaluate: $\sim N^2$ or $\sim N^3$, depending on linkage
  - “Tends to group together those nodes with the strongest connections but leave out those with weaker connections → divisions consist of a few dense cores surrounded by a periphery of unattached nodes”
  - (Results depend on distance and linkage)

- Open question: where to cut the dendrogram?

Method 2: Betweenness based division
Betweenness based division

- Alternative method: instead of agglomerating communities, breaking one large into smaller ones

1) Start from one large community
2) Calculate a centrality measure for each link
3) Remove link with highest centrality
4) Go to step 2

- Centralities: betweenness for edge $i$

$$B(i) = \sum_{s \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

- Other: random walk, ...

Betweenness based division

- Our little example:

(a) [Graph Image]
(b) [Graph Image]
(c) [Graph Image]
(d) [Graph Image]

A.-L. Barabási, *Network Science: Communities.*
Betweenness based division: application

- Zachary’s Karate Club
Betweenness based division

• Advantages:
  – Easy to understand
  – Easy to implement
  – Perhaps less decisions have to be made

• Disadvantages:
  – Slow, number of steps to evaluate: \( \sim N L^2 \) or faster if we don't recalculate the betweenness in each step
  – (Results depend on centrality)

• Still open question: where to cut the dendrogram?
Quality of community division
Modularity

- Again: many existing measures.
- Naïve: fraction of links that are inside communities

\[ Q = \frac{1}{2m} \sum_{ij} A_{ij} \delta_{s_i,s_j} \]

- \( m \): #links in the network
- \( A_{ij} \): adjacency matrix, 1 if \( i \) and \( j \) are connected, 0 if not
- \( \delta_{s_1s_2} \): 1 if in the same community, 0 if not

Modularity

\[ Q = \frac{1}{2m} \sum_{ij} A_{ij} \delta_{s_i,s_j} \]

- Entire network one community: even if links are randomly placed, all links are inside the community
- Instead: fraction of links inside communities compared to what you would expect by chance
- What is by chance?

What is by chance?

- Degree preserved randomization

Original  \[\xrightarrow{\text{}}\]  \[\xrightarrow{\text{}}\]  \[\xrightarrow{\text{}}\]  What we compare to
Modularity

- Modularity:

\[ M = Q = \frac{1}{2m} \sum_{ij} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \delta_{s_i, s_j} \]

- \( m \): #links in the network
- \( A_{ij} \): adjacency matrix, 1 if \( i \) and \( j \) are connected, 0 if not
- \( k_i \): degree of node \( i \)
- \( \delta_{s_1s_2} \): 1 if in the same community, 0 if not
- High \( M \) → good division

Modularity

(a) Optimal Partition
\[ M = 0.41 \]

(b) Suboptimal Partition
\[ M = 0.22 \]

(c) Single Community
\[ M = 0 \]

(d) Negative Modularity
\[ M = -0.12 \]
Modularity

- Where to cut dendrogram? At maximum Q!

Method 3: Direct optimization of $Q$
Direct optimization of modularity

- Exact maximum of $Q \to$ NP-complete (exponentially increasingly difficult with $N$)
- Approximation methods: a lot to choose from
- Louvain method
  - Fast, $\sim L$
  - Typically preforms well on tests
- Two steps applied iteratively:
  1) Find local maximum.
  2) Coarse grain network.

Louvain method: Step 1

1) For node i, calculate $\Delta Q$ for each neighbor j if node i is removed from its community and placed in the community of j. Coarse grain network.

2) Move i to community that maximizes $\Delta Q$, if $\Delta Q > 0$.

3) Repeat while Q increases.

\[
\Delta M_{0,2} = 0.023 \\
\Delta M_{0,3} = 0.032 \\
\Delta M_{0,4} = 0.026 \\
\Delta M_{0,5} = 0.026
\]
Louvain method

1\textsuperscript{st} Pass

\begin{align*}
\Delta M_{0,2} &= 0.023 \\
\Delta M_{0,3} &= 0.032 \\
\Delta M_{0,4} &= 0.026 \\
\Delta M_{0,5} &= 0.026
\end{align*}

2\textsuperscript{nd} Pass

\[]

Application: Belgium phone call network

- Link (A – B) : A and B talk frequently on the phone
- Phone calls of ~2 million customers

Belgium phone call network

- Link (A – B) : A and B talk frequently on the phone
- Phone calls of ~2 million customers


French-speaking Walloons

Dutch-speaking Flemish
Comparing methods
Comparing methods

- We need a network where we know the true community.
- Option I: Real networks with known ground truth
- Option II: Model networks with built-in communities
Comparing methods

- We need a network where we know the true community.
- Option I: Real networks
- Zachary Karate Club

Artificial benchmarks

- Girvan-Newman benchmark
- $N=128$ node divided into 4 groups, $<k>=16$
- $p_{in}$ = prob. that two nodes in the same group are connected
- $p_{out}$ = prob. that two nodes in different groups are connected (not independent)
- $\mu = \text{fraction of external links} = \frac{3p_{out}}{p_{in} + 3p_{out}}$
- No communities: $p_{in}=p_{out}$ or $\mu =0.75$

Artificial benchmarks

- Is the Girvan-Newman benchmark realistic?
- Community size distribution
Artificial benchmarks

- Lancichinetti–Fortunato–Radicchi (LFR) benchmark
- $N$ nodes, $N_c$ communities
- $\mu =$ fraction of external links
- Power-law degree distribution
- Power-law community size distribution

Comparing community divisions

- We know what we should get.
- How to systematically compare what we found?
- Again, a lot of possibilities
- Our choice now: Normalized mutual information

\[ \mu = 0.50 \]
Normalized mutual information

Information theory approach: if two partitions are similar, one needs very little information to infer one partition given the other. We can use the mutual information

Shannon entropy:

\[
H(\{C_1\}) = - \sum_{C_1} p(C_1) \log p(C_1)
\]

Measures the amount of information in a string of random variables drawn from \(p(C_1)\)

Probability that a randomly chosen node belongs to community \(C_1\)

\[
p(C_1) = \frac{\text{how many nodes belong to } C_1}{\text{sum over all partitions}} = \frac{N_{C_1}}{\sum_C N_C}
\]

Information theory approach: if two partitions are similar, one needs very little information to infer one partition given the other. We can use the **mutual information**

\[
I (\{C_1\}, \{C_2\}) = \sum_{C_1} \sum_{C_2} p(C_1, C_2) \log \frac{p(C_1, C_2)}{p(C_1) p(C_2)}
\]

Joint probability that a randomly chosen node belongs to community \(C_i\) in the first partition and \(C_j\) in the second

\[
p(C_i, C_j) = \frac{\text{how many nodes that are in } C_i \text{ are also in } C_j}{\text{sum over all possible pairs } C_i \text{ and } C_j} = \frac{N_{C_i C_j}}{\sum_{C_1, C_2} N_{C_1 C_2}}
\]

Probability that a randomly chosen node belongs to community \(C_i\)

\[
p(C_i) = \frac{\text{how many nodes belong to } C_i}{\text{sum over all partitions}} = \frac{N_{C_i}}{\sum_{C} N_C}
\]

Normalization by average Shannon entropy:

\[
I_n (\{C_1\}, \{C_2\}) = \frac{2I (\{C_1\}, \{C_2\})}{H (\{C_1\}) + H (\{C_2\})}
\]

Normalized mutual information

In summary:

\[ I_n (\{C_1\}, \{C_2\}) = \frac{2I (\{C_1\}, \{C_2\})}{H (\{C_1\}) + H (\{C_2\})} \]

- it quantifies the "amount of information" (in units such as bits) obtained about one random variable, through the other random variable (wiki)
- \( I_n = 1 \rightarrow \) same division
- \( I_n = 0 \rightarrow \) two divisions independent from each other

Benchmarks and NMI in action

- NG benchmark, hierarchical clustering

A.-L. Barabási, *Network Science: Communities*. 
Benchmarks and NMI in action

- NG benchmark, hierarchical clustering

A.-L. Barabási, *Network Science: Communities*. 

\[ \mu = 0.40 \quad I_n = 0.95, n_c = 4 \]

\[ \mu = 0.50 \quad I_n = 0.56, n_c = 6 \]
Benchmarks and NMI in action

- Purple: Hierarchical; Orange: Louvain; Gray: Betweenness

LFR parameters: N = 1000; degree exp. = 2; max degree = 50; comm. size exponent = 1, min comm. size = 20, max = 100

A.-L. Barabási, *Network Science: Communities.*
• Many other methods to find communities:
  – Local: instead of finding global division, find the community a given node belongs to
  – Spectral: based on spectrum of graph Laplacian
  – Dynamical: Potts-model, oscillators, random walks
  – Stochastic block models: find best fit using maximum likelihood fit of benchmark-like model → mathematically principled results
How to choose method?

• What is the best method?
  → No clear answer.

• Better question: What is the method that fits my needs?
  – What do we expect to find? Overlapping communities? Size of the groups?
How to choose method?

• What is the best method?
  → No clear answer.

• Better question: What is the method that fits my needs?
  – What do we expect to find? Overlapping communities? Size of the groups?
## How to choose method?

<table>
<thead>
<tr>
<th>Name</th>
<th>Overlap</th>
<th>Dir</th>
<th>Weight</th>
<th>Dyn</th>
<th>NoPar</th>
<th>MDim</th>
<th>Incr</th>
<th>Multip</th>
<th>Complexity</th>
<th>BESn</th>
<th>BESm</th>
<th>Year</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolutionary*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(O(n^2))</td>
<td>5k</td>
<td>?</td>
<td>2006</td>
<td>21</td>
</tr>
<tr>
<td>MSN-BD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(O(n^2ck))</td>
<td>6k</td>
<td>3M</td>
<td>2006</td>
<td>22</td>
</tr>
<tr>
<td>SocDim</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(O(n^2 \log n))*</td>
<td>80k</td>
<td>6M</td>
<td>2009</td>
<td>23</td>
</tr>
<tr>
<td>PMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(O(mn^2))</td>
<td>15k</td>
<td>27M</td>
<td>2009</td>
<td>24</td>
</tr>
<tr>
<td>MRGC</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(O(mD))</td>
<td>40k</td>
<td>?</td>
<td>2007</td>
<td>25</td>
</tr>
<tr>
<td>Infinite Relational</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(O(n^2cD))</td>
<td>160</td>
<td>?</td>
<td>2006</td>
<td>26</td>
</tr>
<tr>
<td>Find-Tribes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(mnK^2))</td>
<td>26k</td>
<td>100k</td>
<td>2007</td>
<td>27</td>
</tr>
<tr>
<td>AutoPart</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(mk^2))</td>
<td>75k</td>
<td>500k</td>
<td>2004</td>
<td>28</td>
</tr>
<tr>
<td>Timefall</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(O(mk))</td>
<td>7.5M</td>
<td>53M</td>
<td>2008</td>
<td>29</td>
</tr>
<tr>
<td>Context-specific Cluster Tree</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(mk))</td>
<td>37k</td>
<td>367k</td>
<td>2008</td>
<td>30</td>
</tr>
<tr>
<td>Modularity</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(mk \log n))</td>
<td>118M</td>
<td>1B</td>
<td>2004</td>
<td>31</td>
</tr>
<tr>
<td>MetaFac</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(mnD))</td>
<td>115k</td>
<td>613</td>
<td>2008</td>
<td>32</td>
</tr>
<tr>
<td>Variational Bayes</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(mk + n))</td>
<td>16k</td>
<td>?</td>
<td>2005</td>
<td>33</td>
</tr>
<tr>
<td>LA → IS^2*</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(nK \log n))</td>
<td>108k</td>
<td>330k</td>
<td>2005</td>
<td>34</td>
</tr>
<tr>
<td>L-Shell</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(n^2 \log n))</td>
<td>77k</td>
<td>254</td>
<td>2005</td>
<td>35</td>
</tr>
<tr>
<td>Internal-External Degree</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(n^2 \log n))</td>
<td>775k</td>
<td>4.7M</td>
<td>2009</td>
<td>36</td>
</tr>
<tr>
<td>Edge Betweenness</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(m^2n))</td>
<td>271k</td>
<td>1k</td>
<td>2002</td>
<td>37</td>
</tr>
<tr>
<td>CONGO*</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(n\log n))</td>
<td>30k</td>
<td>116k</td>
<td>2005</td>
<td>38</td>
</tr>
<tr>
<td>Label Propagation</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(m + n))</td>
<td>374k</td>
<td>30M</td>
<td>2007</td>
<td>39</td>
</tr>
<tr>
<td>Node Colouring</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(nk^2))</td>
<td>2k</td>
<td>?</td>
<td>2007</td>
<td>40</td>
</tr>
<tr>
<td>Kirchhoff</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(m + n))</td>
<td>115k</td>
<td>613</td>
<td>2004</td>
<td>41</td>
</tr>
<tr>
<td>Communication Dynamic</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(mnt))</td>
<td>160k</td>
<td>530k</td>
<td>2008</td>
<td>42</td>
</tr>
<tr>
<td>GuruMine</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(T.An^2))</td>
<td>217k</td>
<td>212k</td>
<td>2008</td>
<td>43</td>
</tr>
<tr>
<td>DegreeDiscountIC</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(k \log n + m))</td>
<td>37k</td>
<td>230k</td>
<td>2009</td>
<td>44</td>
</tr>
<tr>
<td>MMSB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(nk))</td>
<td>871</td>
<td>2k</td>
<td>2007</td>
<td>45</td>
</tr>
<tr>
<td>Walktrap</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(m^2n))</td>
<td>160k</td>
<td>1.8M</td>
<td>2006</td>
<td>46</td>
</tr>
<tr>
<td>DOC5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(m \log^2 n))</td>
<td>6k</td>
<td>6M</td>
<td>2008</td>
<td>47</td>
</tr>
<tr>
<td>Infomap</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(m^2n))</td>
<td>20k</td>
<td>127k</td>
<td>2005</td>
<td>48</td>
</tr>
<tr>
<td>K-Clique</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(kmn))</td>
<td>200k</td>
<td>500k</td>
<td>2008</td>
<td>49</td>
</tr>
<tr>
<td>S-Plexes Enumeration</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(m^2))</td>
<td>16k</td>
<td>31k</td>
<td>2009</td>
<td>50</td>
</tr>
<tr>
<td>Bi-Clique</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(3^k))</td>
<td>20k</td>
<td>127k</td>
<td>2009</td>
<td>51</td>
</tr>
<tr>
<td>EAGLE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(2nk \log n))</td>
<td>885k</td>
<td>5.5M</td>
<td>2010</td>
<td>52</td>
</tr>
<tr>
<td>Link modularity</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(nK^2))</td>
<td>4.8M</td>
<td>42M</td>
<td>2011</td>
<td>53</td>
</tr>
<tr>
<td>HLC*</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(nk))</td>
<td>885k</td>
<td>5.5M</td>
<td>2010</td>
<td>54</td>
</tr>
<tr>
<td>Link Maximum Likelihood</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(2nk \log n))</td>
<td>20k</td>
<td>127k</td>
<td>2009</td>
<td>55</td>
</tr>
<tr>
<td>Hybrid*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(nK))</td>
<td>325k</td>
<td>1.5M</td>
<td>2010</td>
<td>56</td>
</tr>
<tr>
<td>Multi-relational Regression</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(n^2))</td>
<td>1k</td>
<td>4k</td>
<td>2008</td>
<td>57</td>
</tr>
<tr>
<td>Hierarchical Bayes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>(O(n^2))</td>
<td>112</td>
<td>?</td>
<td>2007</td>
<td>58</td>
</tr>
<tr>
<td>Expectation Maximization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>(O(nK))</td>
<td>325k</td>
<td>1.5M</td>
<td>2010</td>
<td>59</td>
</tr>
</tbody>
</table>

Where to start reading?

   → Short, big picture

   → >100 pages, complete at the time, good for looking up methods

   → shorter, compares a lot of methods

4) A.-L. Barabási, Network Science, Chapter 9
   http://barabasi.com/networksciencebook/
   Appears in print in May.
   → Lot of figures of the lecture are from here. Easy to read, tells a detailed story, but does not cover everything.
Extra time:
Problems with modularity
Is the maximum unique?

- Should we merge two communities?
- Intuitive:

\[
\begin{array}{c}
\text{Global maximum:} \\
M = 0.871
\end{array}
\]

- Random:

\[
\begin{array}{c}
\text{Random:} \\
M = 0.80
\end{array}
\]

Good et al., *PRE*, 81:046106 (2010)
A.-L. Barabási, *Network Science: Communities.*
Resolution limit

- Should we merge two communities?

\[
\Delta M_A = \frac{1}{2L} \sum_{i,j \in A} A_{i,j} - \frac{k_i k_j}{2L} = \frac{2k_A^{\text{int}}}{2L} - \frac{1}{2L} \sum_{i,j \in A} k_i k_j
\]

\[
\Delta M_{AB} = -M_A - M_B + M_{AB} =
\]

\[
= - \left( \frac{k_A^{\text{int}}}{L} - \sum_{i,j \in A} \frac{k_i k_j}{(2L)^2} \right) - \left( \frac{k_B^{\text{int}}}{L} - \sum_{i,j \in B} \frac{k_i k_j}{(2L)^2} \right) + 
\]

\[
+ \left( \frac{k_A^{\text{int}} + k_B^{\text{int}} + L_{AB}}{L} - \sum_{i,j \in A} \frac{k_i k_j}{(2L)^2} - \sum_{i,j \in B} \frac{k_i k_j}{(2L)^2} - 2 \sum_{i \in A, j \in B} \frac{k_i k_j}{(2L)^2} \right) = 
\]

\[
= \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2}
\]


A.-L. Barabási, *Network Science: Communities.*
Resolution limit

- Should we merge two communities?

$$\Delta M_{AB} = \frac{L_{AB}}{L} - \frac{k_A k_B}{2L^2},$$

where $k_A$ and $k_B$ are the total degree in $A$ and $B$.

If

$$\frac{k_A k_B}{2L} < 1 \quad \text{and} \quad L_{AB} \geq 1 \quad \Rightarrow \quad \Delta M_{AB} > 0$$

We merge $A$ and $B$ to maximize modularity.

Assuming

$$k_A \sim k_B = k \quad \Rightarrow \quad k \leq \sqrt{2L}$$

Modularity has a resolution limit, as it cannot detect communities smaller than this size.


A.-L. Barabási, *Network Science: Communities.*
Even more time:
Link communities
Link communities

Social networks, a link may indicate:
- they are in the same family; they work together; they share a hobby.

Biological networks:
each interaction of a protein is responsible for a different function, uniquely defining the protein’s role in the cell.

Define a hierarchical algorithm based on similarity of links

A.-L. Barabási, Network Science: Communities.
1. Define link similarity

\[
S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}
\]

\(n_+(i)\): the list of the neighbors of node \(i\), including itself.

\(S\) measures the relative number of common neighbors \(i\) and \(j\) have.

\[
|n_+(i) \cap n_+(j)| = 4
\]

\[
|n_+(i) \cup n_+(j)| = 12
\]

\[
S(e_{ik}, e_{jk}) = \frac{1}{3}
\]
2. Apply hierarchical clustering (agglomerative, single linkage)
The network of characters in Victor Hugo’s 1862 novel *Les Miserables*. Two characters are connected if they interact directly with each other in the story. The link colors indicate the clusters, grey nodes corresponding to single-link clusters. Each node is depicted as a pie-chart, illustrating its membership in multiple communities. Not surprisingly, the main character, Jean Valjean, has the most diverse community membership.

A.-L. Barabási, *Network Science: Communities*. 