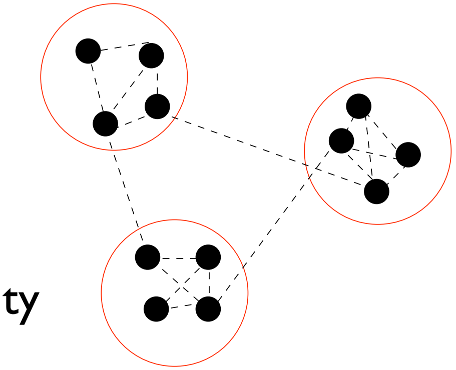
Community Detection as an Inference Problem

M. B. Hastings, T-13, LANL

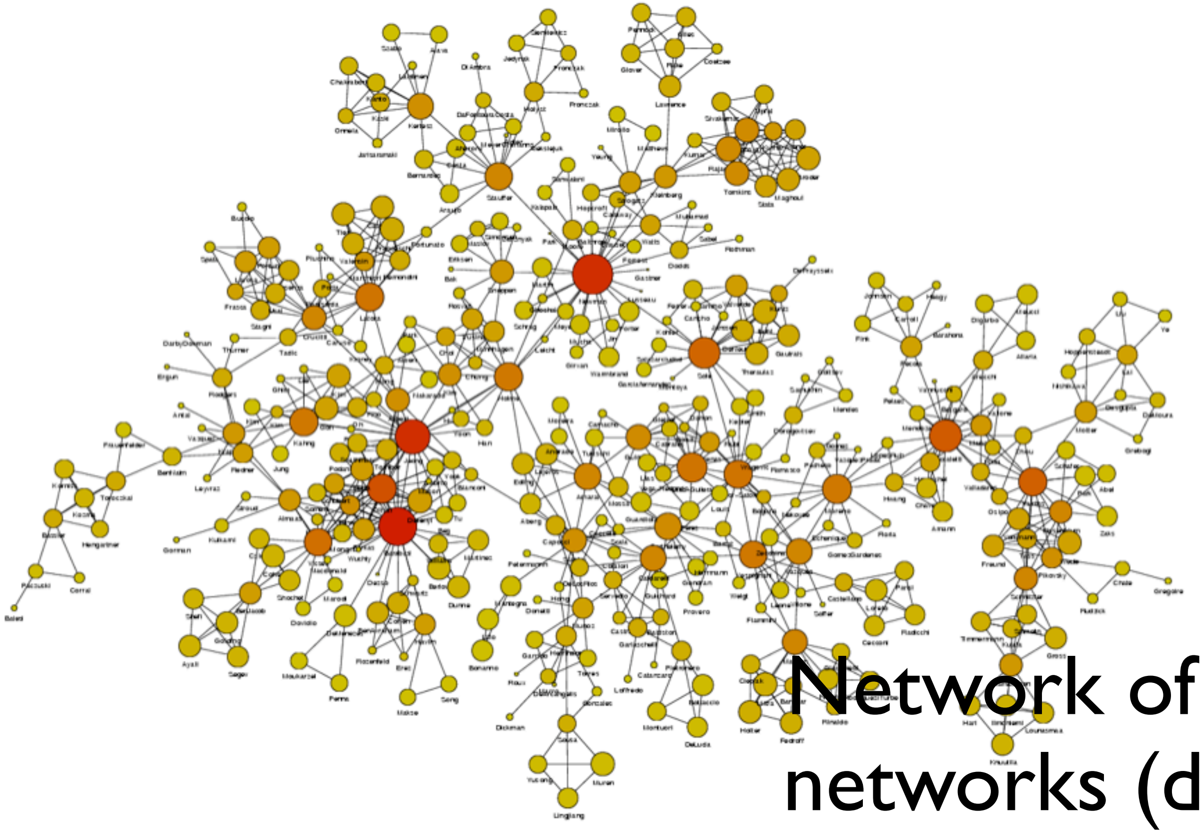
Thanks: E. Ben-Naim, M. Chertkov, Z. Toroczkai

What is a community?

Network: vertices connected by edges



Community: subset of nodes on the network such that nodes in same community are more likely to be connected than nodes in different communities



Examples: division of social networks in groups, division of biological networks, routing in communication networks, etc...

Network of scientists working on networks (due to Mark Newman)

M.E.J. Newman, physics/0605087

Really, what is a community?

"I know it when I see it"?

Many algorithms proposed, such as Newman-Girvan, 2004. See L. Danon et. al. J Stat. P09008 (2005) for a review of methods and performance.

How to quantify performance? Clearly, we want the "best" algorithm.

One precise problem: graph partitioning

Divide a graph into two equal communities so as to minimize the number of links between communities.

Outline:

- The “four groups” test for performance: given a set of communities generate a random network to test an algorithm
- Bayesian inference problems: given a network, deduce the most likely community assignment
- Error-correcting codes, communication on a noisy channel, and belief propagation
- The BP and MFT algorithms for community detection and their performance
- Outlook

The "Four Groups" Test

Newman, Girvan (2004)

- Start with N nodes divided into q different communities (typically, N=128, q=4)
- Connect nodes in the same community independently at random with probability p_{in}
- Connect nodes in different communities with probability p_{out}

Define average number of links to same or different community $z_{in,out}$

Run a community detection algorithm on the given network. See if its assignment of communities matches the initial assignment of communities!

This tests the accuracy of the algorithm. Higher accuracy is better. The problem is easier when z_{in} is large and z_{out} is small as the communities are better defined

Note that there is an arbitrariness in this definition: the communities can be "re-labeled" arbitrarily. See Newman (2004) for precise definition of accuracy.

Inference problem: given graph, what is the probability that a given community assignment is correct?

Bayesian method: probability that a given community assignment is correct is proportional to the a priori expectation for that assignment multiplied by the probability that the given community assignment produces the given graph

Let q_i be the community assignment for node i where $1 \leq q_i \leq q$

Then, it may be shown that the probability of producing a given graph is equal to $p(\{q_i\}) \propto \exp[\sum_{<ij>} J\delta_{q_i,q_j}] \exp[\sum_{i \neq j} J'\delta_{q_i,q_j}/2]$

$$J = \log[(p_{in}(1-p_{out}))/((1-p_{in})p_{out})],$$
$$J' = \log[(1-p_{in})/(1-p_{out})]$$

J>0, J'<0, ferro short-range couplings, AF long-range couplings

For simplicity we ignore the a priori knowledge of exactly 128/4=32 nodes in each community and just use $p(\{q_i\})$

This gives a Potts model. Compare to previous phenomenological derivation of Potts model (Reichardt and Bornholdt). Freedom to re-label communities becomes Potts symmetry

Maximum likelihood (most likely assignment of communities) is ground state. We do something slightly different.

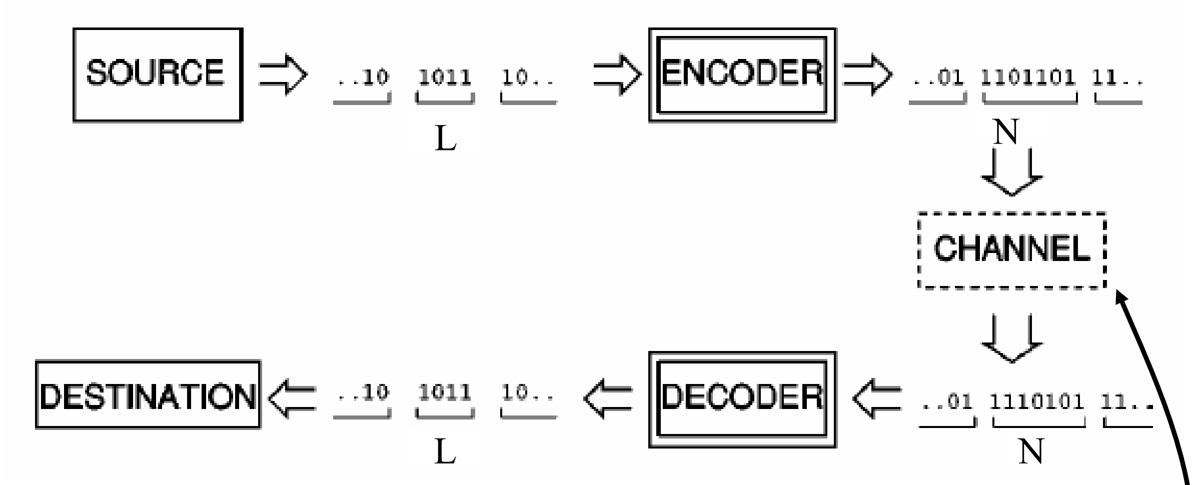
Error-correcting codes:

Digital Error-Correction

N > L R=L/N - code rate

Coding

Decoding



noise

$$\vec{x}^{(in)} \to \vec{x}^{(out)} = \vec{x}^{(in)} - \vec{\varphi} \qquad \vec{x} = \left(x_1, \cdots, x_N\right)$$

example

white $$P(\vec{x}^{(out)} \mid \vec{x}^{(in)}) = \prod_{i=1}^{N} p(x_i^{(out)} \mid x_i^{(in)})$$

Gaussian symmetric $$p(x \mid y) = \exp\left(-\frac{s^2}{2}(x-y)^2\right)\sqrt{s^2/2\pi}$$

channel

menu

Receiver can compute probability that any given message was sent. Want to find most likely message. This is a tough problem! (exponentially many possible messages)

Low-density parity check codes (LDPC)

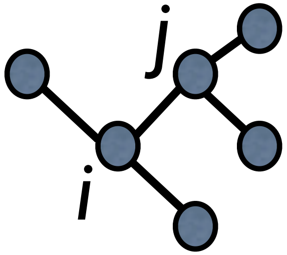
Gallagher, 1963

Finding most likely message corresponds to finding the ground state of an 2-state Potts model on a graph with interactions between neighbors and magnetic fields. Spin up=1, spin down=0.

The graph in LDPC has few loops ("low-density")

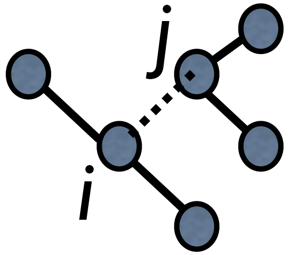
Exact solution at non-zero temperature (noise) on a tree (no loops) with Bethe-Peierls technique:

Define $p_i(q_i)$ to be the probability the node i is in state $q_i, 1 \leq q_i \leq 2$



$$p_i(r) \propto \exp[h_i(r)] \times \prod_{j}^{i,j n.n.} [\exp(J) p_{ji}(r) + (1 - p_{ji}(r))]$$

Define $p_{ij}(q_i)$ to be the probability the node i is in state q_i ***on the graph with the link from i to j removed***



$$p_{ij}(r) \propto \exp[h_i(r)] \times \prod_{k \neq j}^{i,k n.n.} [\exp(J) p_{ki}(r) + (1 - p_{ki}(r))]$$

Belief Propagation Algorithm: use the same equations on a graph with a low density of loops. Not exact but a good approximation.

Difficulties using BP for community detection:

- Graph is highly connected (long-range links between all nodes). Too many variables to use BP for these long-range links.

- Need a spontaneous symmetry breaking to get a group assignment.

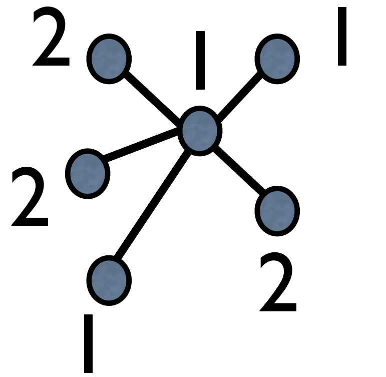
Solution: treat long-range interactions using mean-field theory. Treat short-range using BP.

$$h_i(r) = J' \sum_{j \neq i} p_i(r)$$

Extracting information from the BP algorithm

- Rather than maximum-likelilhood (most probable assignment of communities for all nodes), we assign each node to the most likely community for that node. (requires spontaneous symmetry breaking, alternative is to use correlation functions?)

- Maximum accuracy. If by chance a given node connects to more nodes of a different community then it will be mis-classified. Maximum about 92% for $z_{in} = z_{out} = 8, q = 3$



Can carry to further levels of a tree-like graph to bound the accuracy

Implementing the BP algorithm for community detection:

- Solve BP equations iteratively. We chose the following: pick a random node and update belief. Pick a random edge and update belief. Repeat. Replace belief with weighted average of old belief and solution of BP equations. $p_i(r) \to 0.75 * p_i(r) + 0.25 * \exp[h_i(r)] \prod_{j}^{i,j n.n.} [\exp(J) p_{ji}(r) + (1 - p_{ji}(r))]$

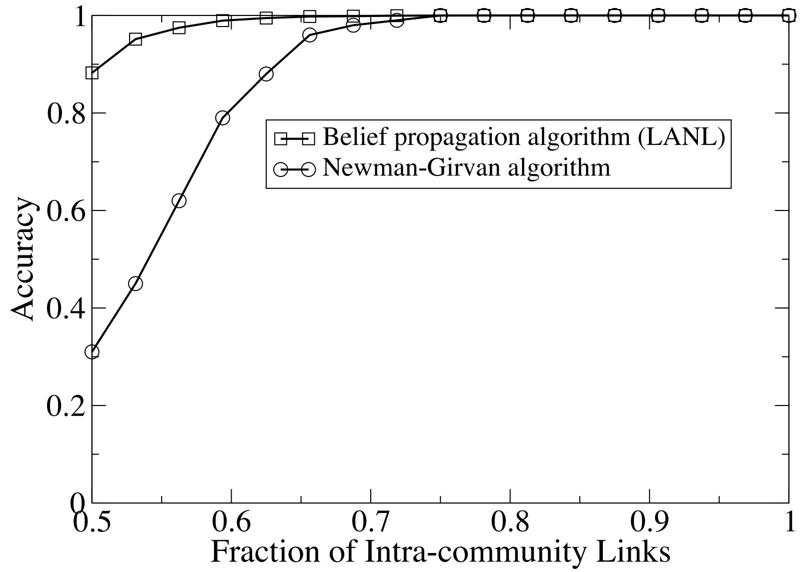
- Possible failure mode: BP equations do not converge. Diagnostic: beliefs oscillate. Solution: more iterations, less change on iteration. Note that for error-correction various other methods are used to solve BP equations.

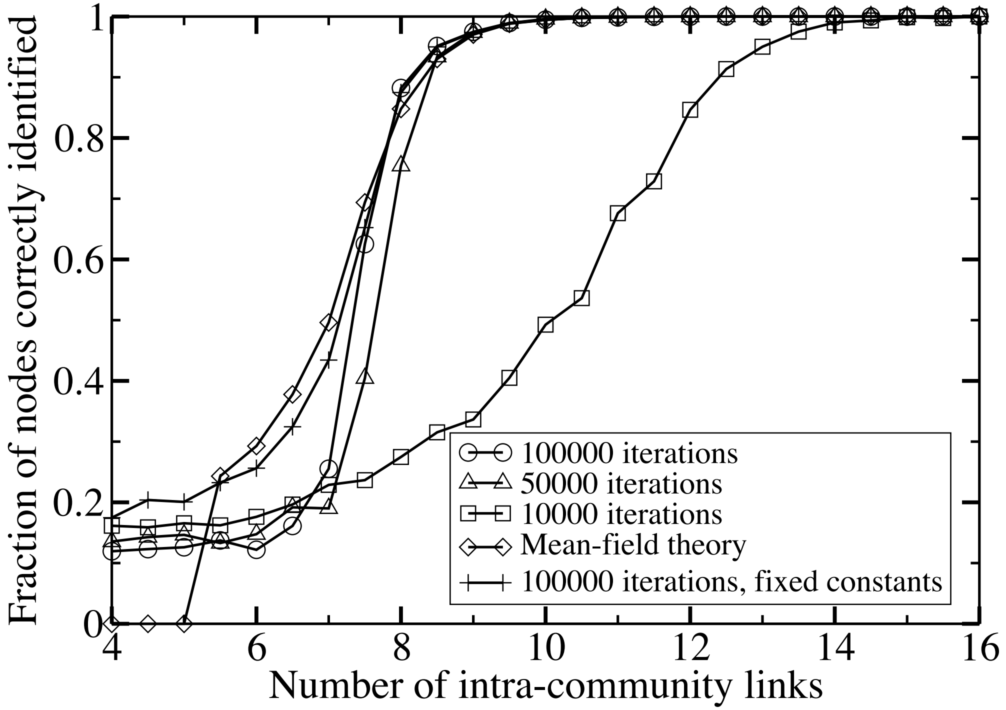
- Possible failure mode: BP equations do not break symmetry. Diagnostic: beliefs converge to symmetric solution. Solution: lower temperature, larger J,J’. Both failure modes can be detected by looking at beliefs

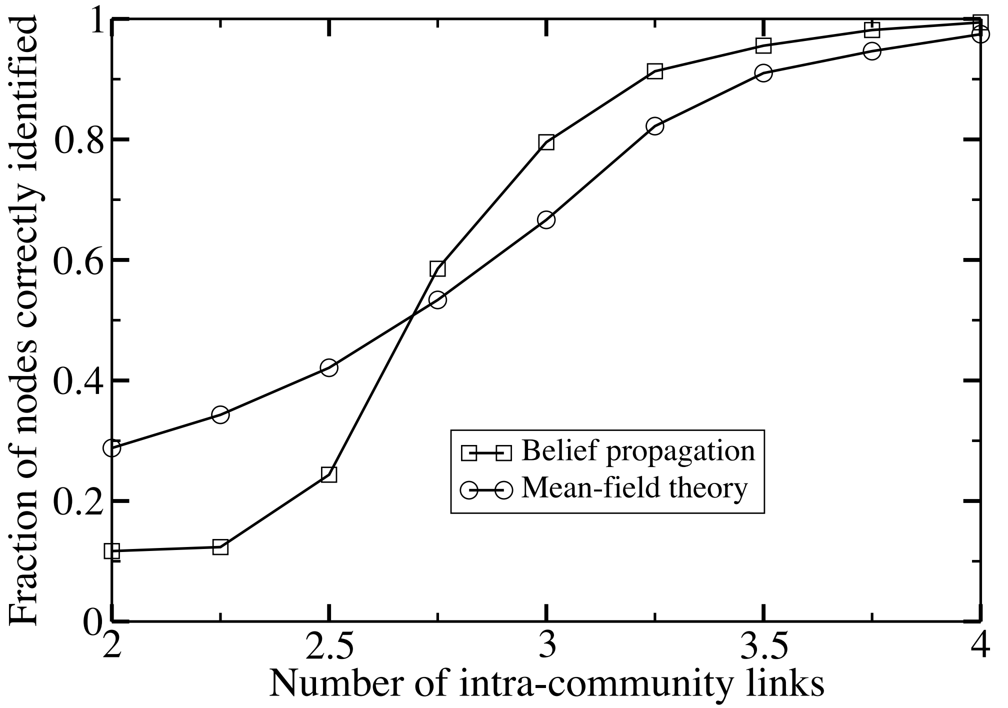
- Note: in practice, a range of values of J,J’ work, not just those from Bayesian estimate.

- Alternative algorithm: use MFT for all interactions. Simpler, faster.

Results:







Results for graphs with average coordination number 16 and 4. Outperforms Newman-Girvan. Seems to outperform all others too (see L. Danon). Only algorithm with comparable performance is annealing (slow).

Step 1: inference

More general inference problem: can use any starting distribution of links. Example:

$$z^i_{in} + z^i_{out} = k^i;\ z_{in}/z_{out} = \text{const.}$$

$$\rightarrow \exp[J_{ij}\delta_{q_i,q_j}]$$

Coupling constants depend on degrees

Step 2: optimization

Solving Potts problem: belief propagation (or survey or mean-field or loop equations (Chertkov 2006))

Magnitude of beliefs carries information

Conclusion

- Inference formulation of community detection
- Belief propagation is very accurate
- Time required: number of iterations=(number of nodes)*(iterations/node). The required (iterations/node) scales as a phase ordering time. We guess that networks without spatial structure take time $T \propto N \log^{\alpha}(N)$
- Application to clustering problems (M. B. Hastings, in preparation)