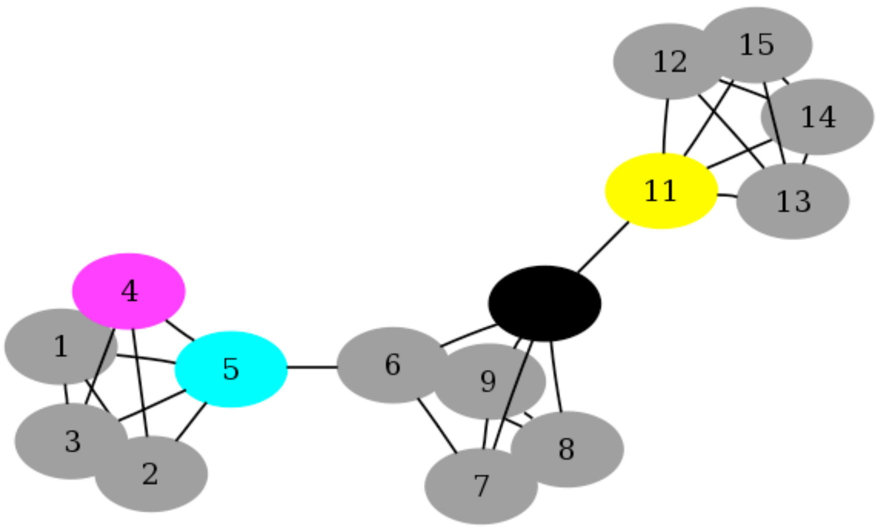
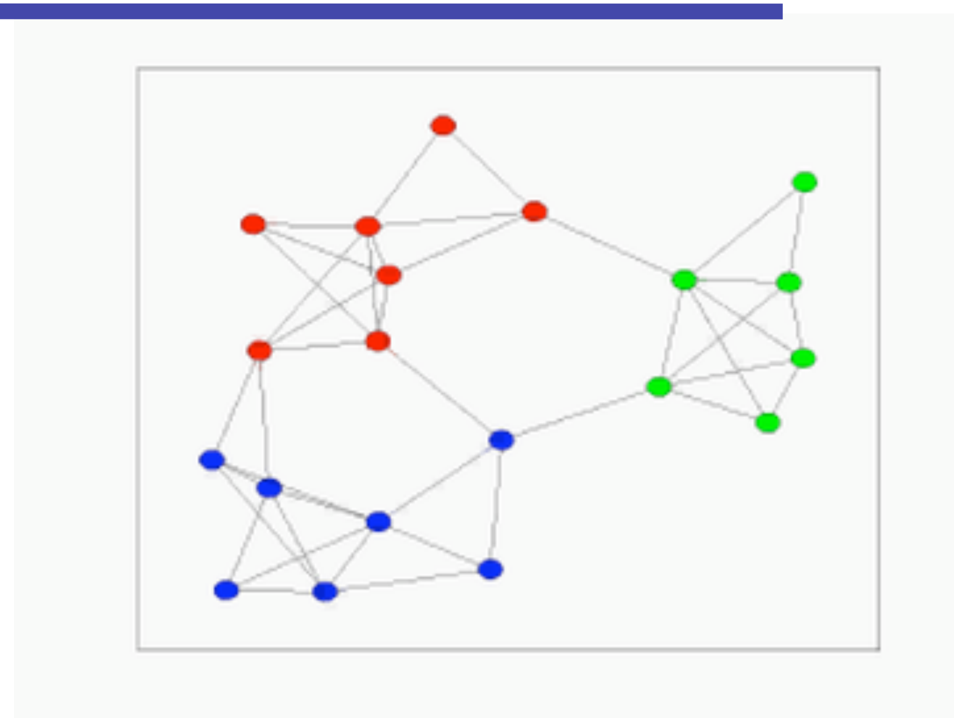
inferring & encoding graph partitions

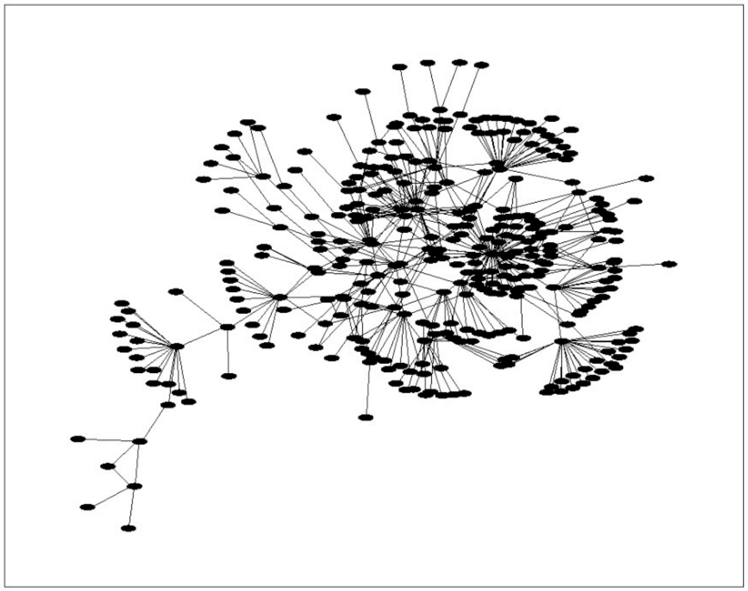


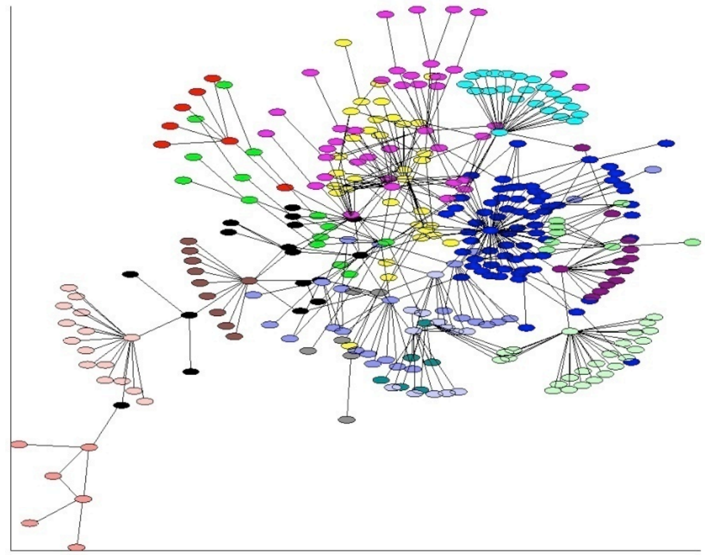


chris.wiggins@columbia.edu

* APAM: department of applied physics and applied mathematics
* C2B2: center for computational biology and bioinformatics

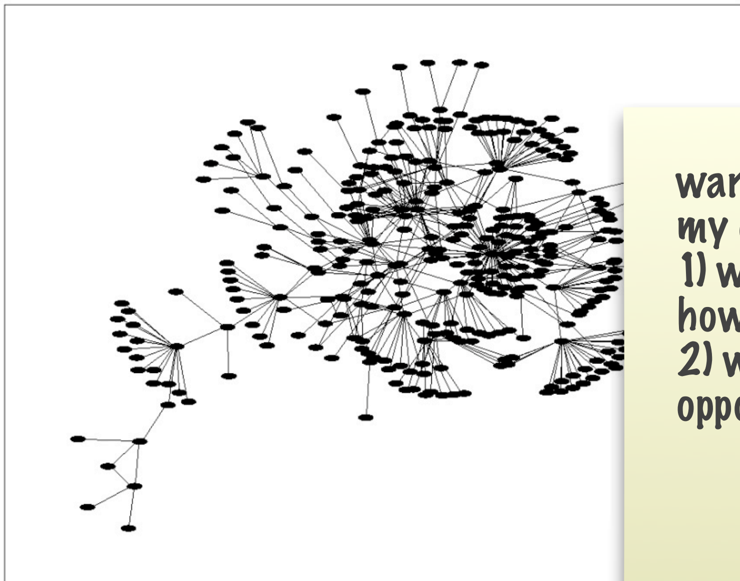
agenda





Part 0: context
Part 1: inferring modules
Part 2: encoding modules

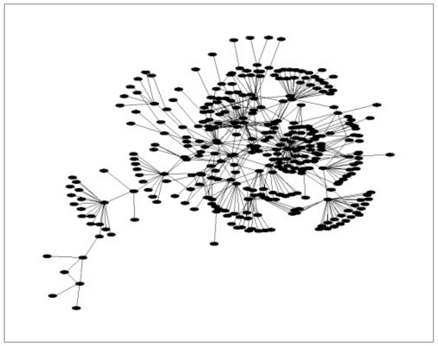
agenda

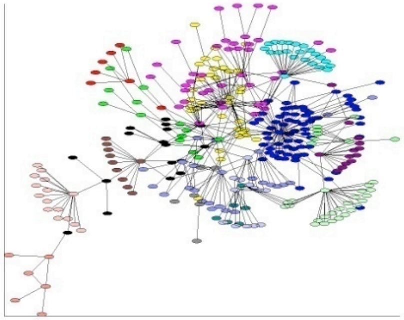


warning. not an algorithmicist, ergo much of my concern is about
1) what is to be optimized (as opposed to how to optimize it)
2) whether you reveal the "truth" (as opposed to how fast you find what you find)

Part 0: context
Part 1: inferring m
Part 2: encoding r ... sure)

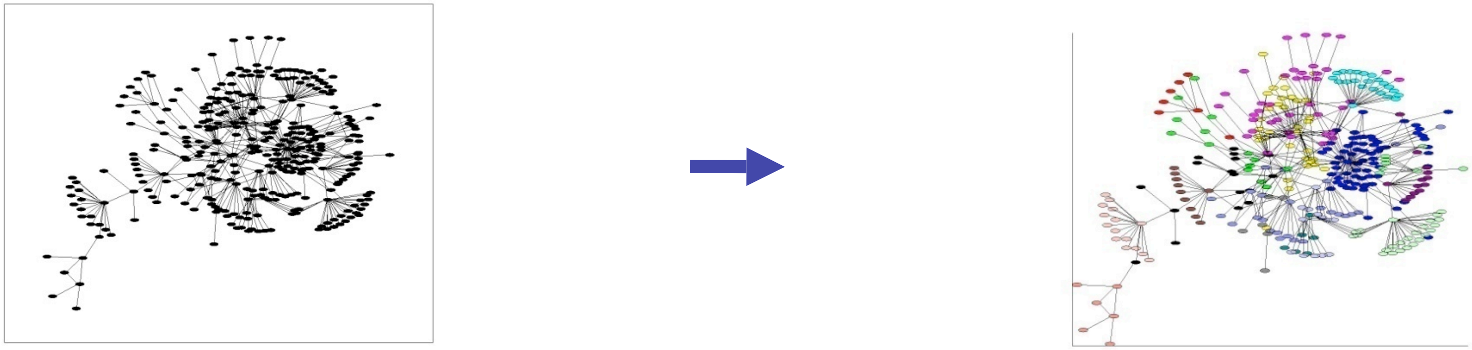
Part 1: inferring modules





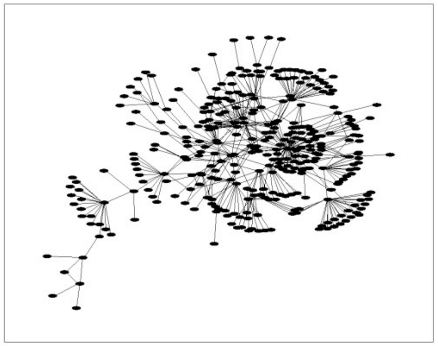
- pseudohistory
- problems and solutions
- algorithm
- results

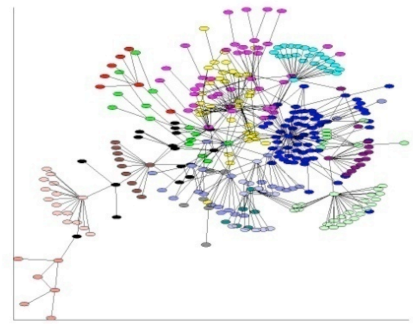
inferring modules:



- pseudohistory:
  1980s-2006: “community” models (1980s) as validation
  2006-now: use model to infer community directly

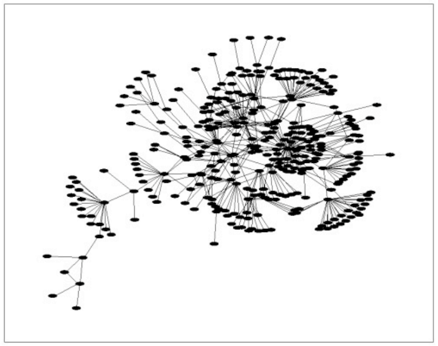
inferring modules: arxiv-history

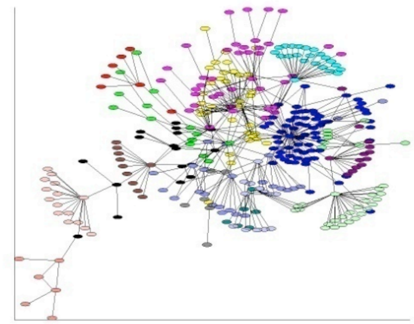




- April 2006: Hastings (arXiv:cond-mat/0604429)
  - need coupling constants
  - needs # modules
- May 2007: Newman+Liecht (arXiv:physics/0611158)
  - infer coupling constants
  - needs # modules
- June 2007: Xing + coworkers (arXiv:0706.0294)
  - infer coupling constants
  - Get # modules via BIC
- Sep 2007: Hofman+C.W. (arXiv:0709.3512)
  - infer coupling constants
  - infer # modules

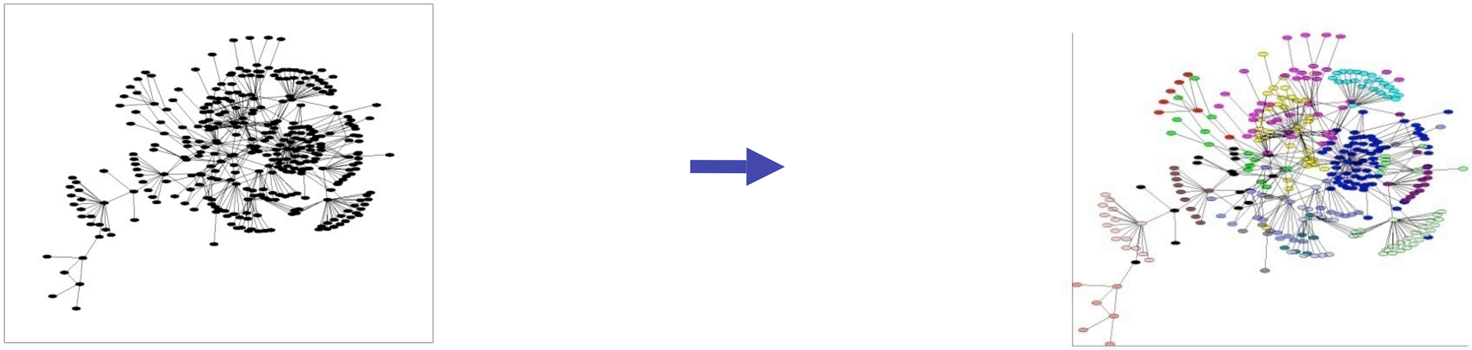
inferring modules:





- pseudohistory:
  1980s-2006: “community” models (1980s) as validation
  2006-now: use model to infer community directly
- key tools:
  - in statistical physics: “mean field”/”test hamiltonians” (Feynman/Bogoliubov)
  = in ML/statistics: “variational bayesian methods” (Jordan, Mackay)

inferring modules:

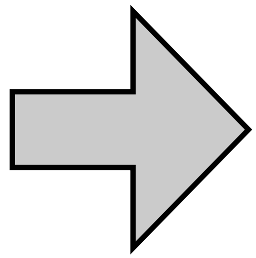


1. how would you do this?
   an example, some math

2. what could possibly go wrong?
   overfitting

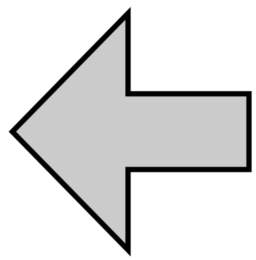
Community detection as inference

Know ensemble (parameters, assignment variables, complexity)



Sample microstates

Infer ensemble (parameters, latent variables, complexity)

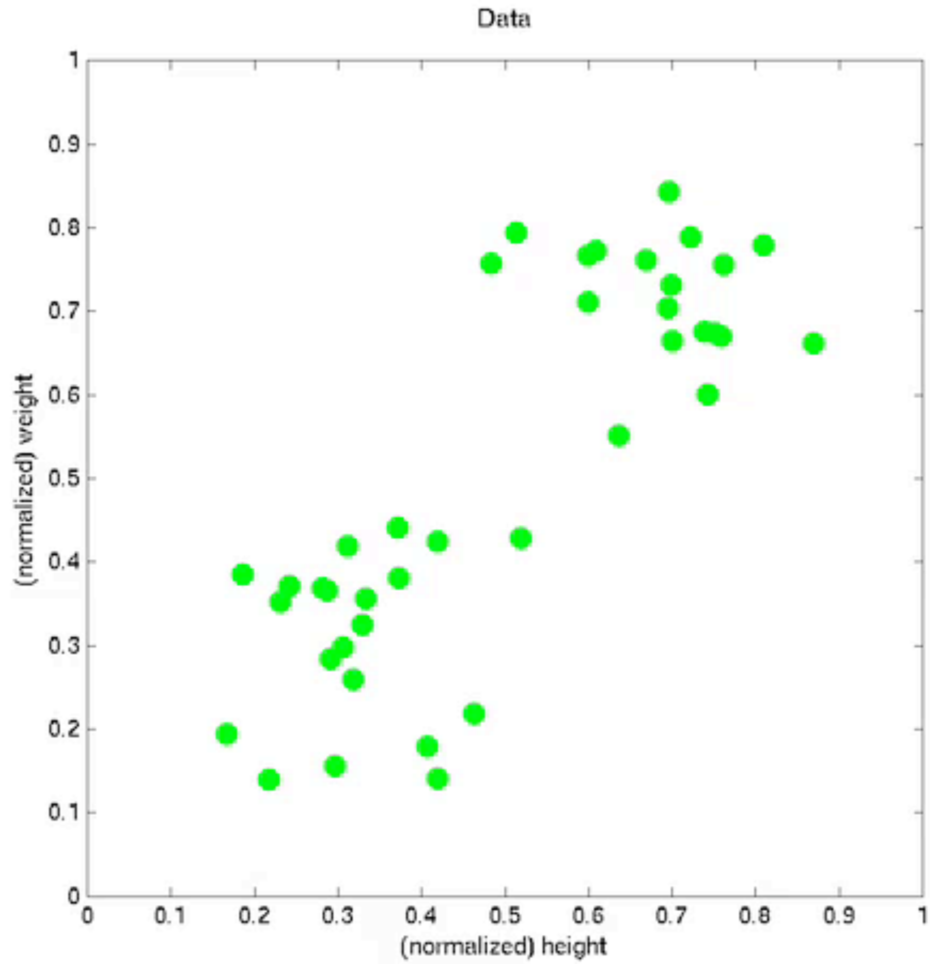


Observe microstate

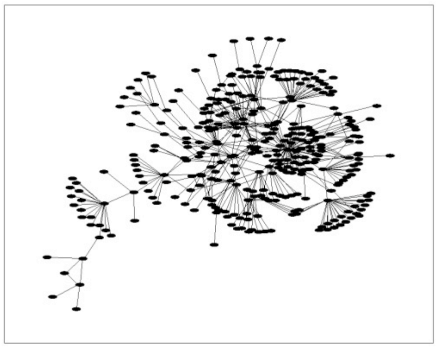
how would you do this?

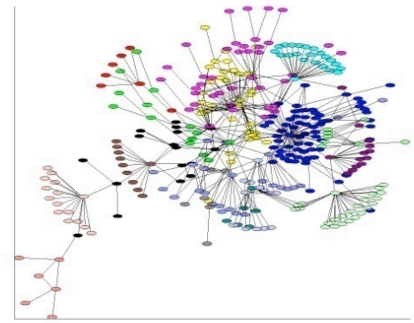
infer "modules" as latent variables in generative model, just like in mixture modeling

movie by J. Hofman using **Netlab** (Nabney+Bishop)



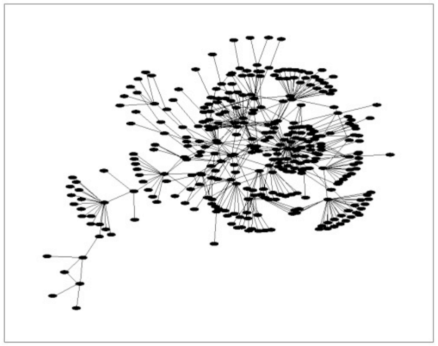
inference+maximum likelihood

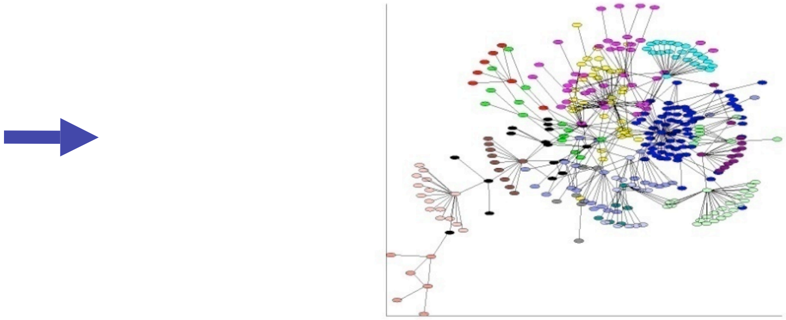




- why we fit: maximum likelihood (math)
- mixture modeling: maximum likelihood (movie)

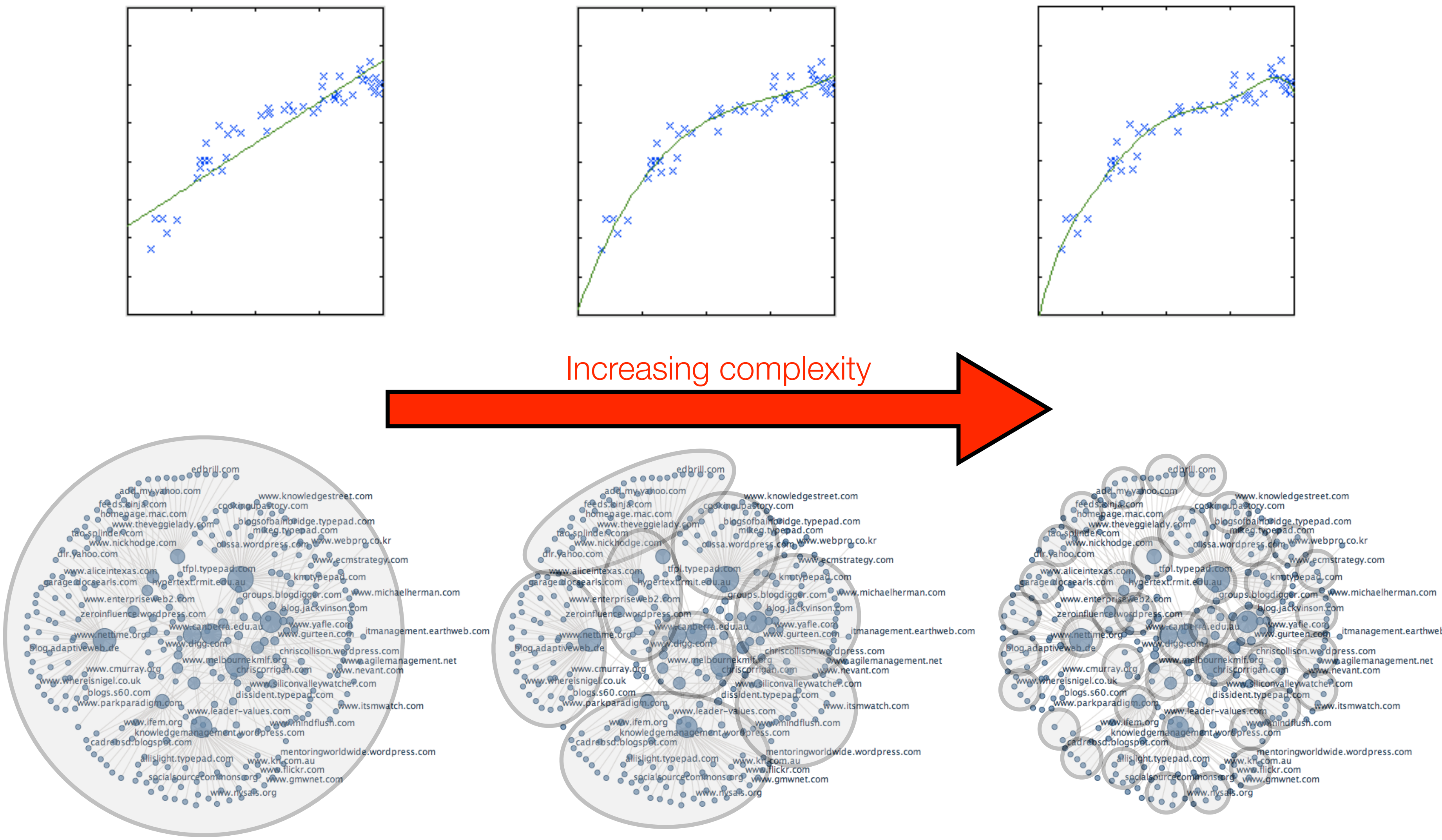
problem: complexity control





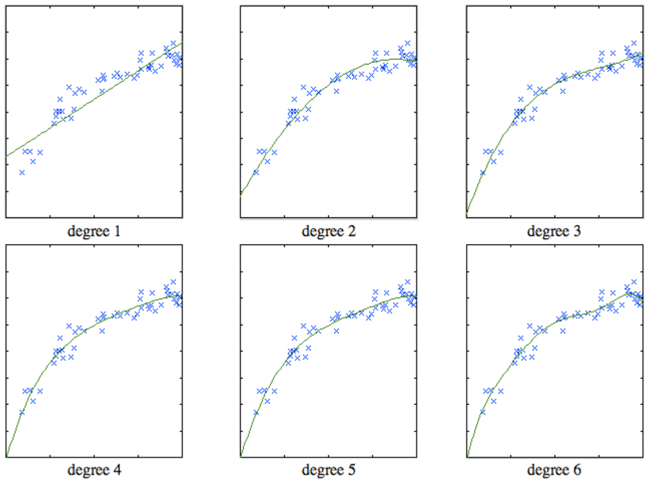
- "overfitting" (e.g., in regression)
- over complexifying (e.g., in mixture models)
  - cross validation would be nice, but...
  - bayesian approaches?

Complexity control in probabilistic models

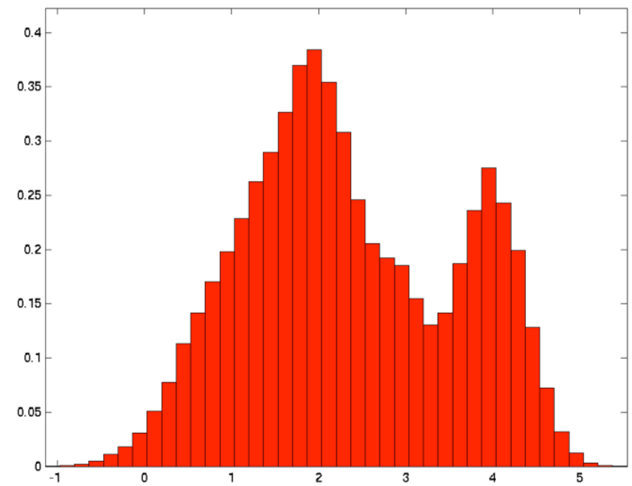


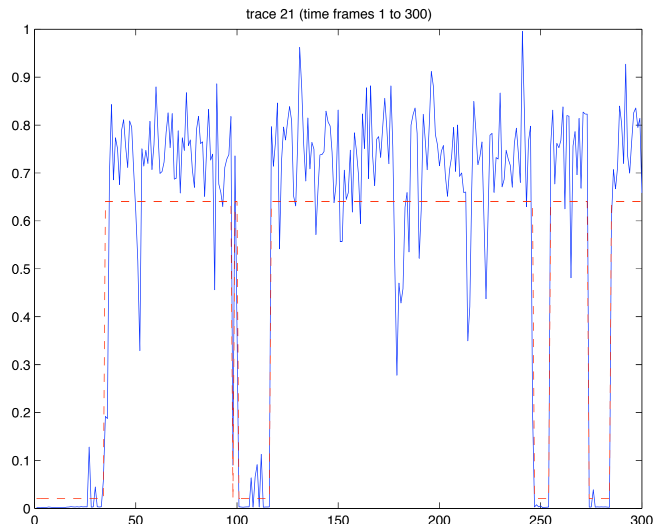
Complexity control in probabilistic models

What degree?

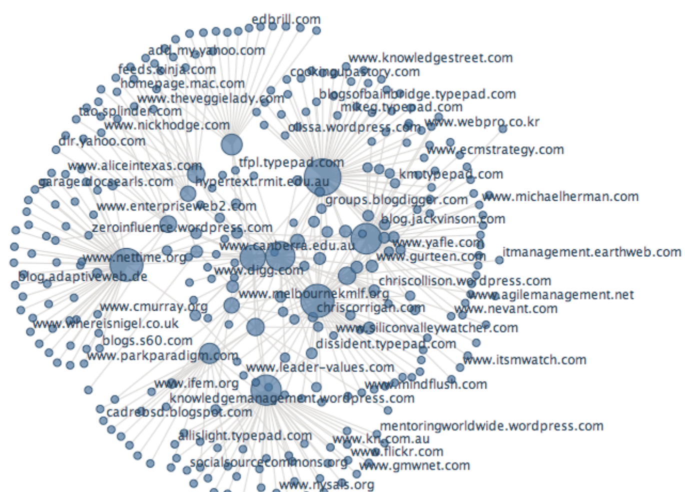


How many Gaussians?





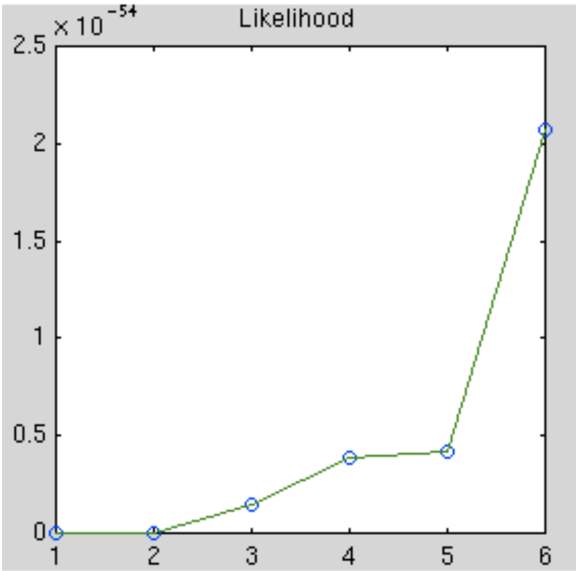
How many states?



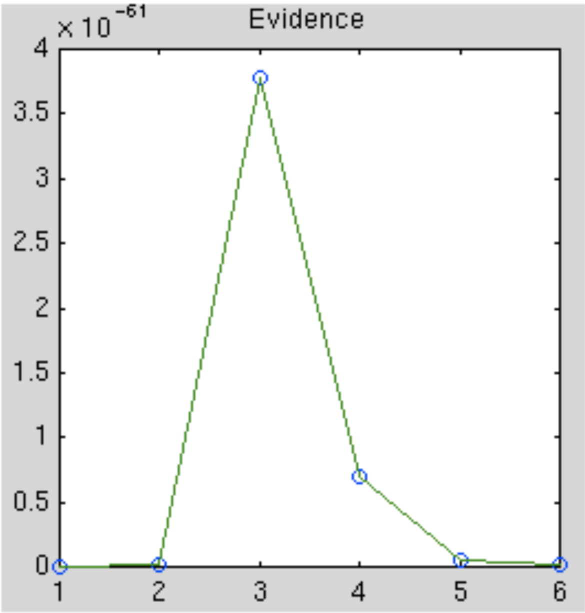
How many modules?

Complexity control in probabilistic models

- Maximize evidence (integrating over unknown parameters) to infer most probable model complexity



$$p(\mathcal{D}|\hat{\theta}, K)$$



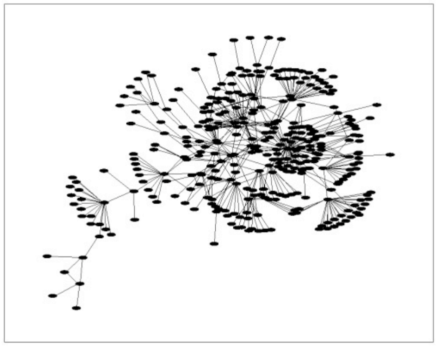
$$p(\mathcal{D}|K) = \int d\theta \; p(\mathcal{D}|\theta, K) p(\theta|K)$$

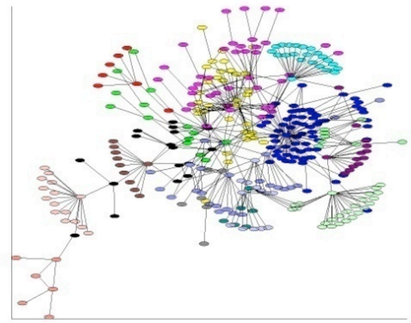
http://research.microsoft.com/~minka/statlearn/demo/

why evidence?
intuition (Schwartz, 1978):

$$
\begin{aligned}
p(\mathcal{D}|K) &= \int d\theta p(\mathcal{D}|\theta, K) p(\theta|K) \\
&= \int d\theta \mathrm{e}^{\ln p(\mathcal{D}|\theta, K)} p(\theta|K) \\
&\approx \mathrm{e}^{\ln p(\mathcal{D}|\hat{\theta}, K)} p(\hat{\theta}|K) \sqrt{\left|\frac{2\pi}{\nabla_\theta \nabla_\theta \ln p(\mathcal{D}|\hat{\theta}, K)}\right|} \\
&\sim C_1 \mathrm{e}^{\ln p(\mathcal{D}|\hat{\theta}, K)} \left(\frac{2\pi}{N}\right)^{K/2} \\
&\sim C_2 \mathrm{e}^{-\left(-\ln p(\mathcal{D}|\hat{\theta}, K) + \frac{1}{2} K \ln N\right)} \\
\text{e.g.,} &\sim C_2 \mathrm{e}^{-\boxed{\left(\chi^2 + \frac{1}{2} K \ln N\right)}}
\end{aligned}
$$

inferring modules:



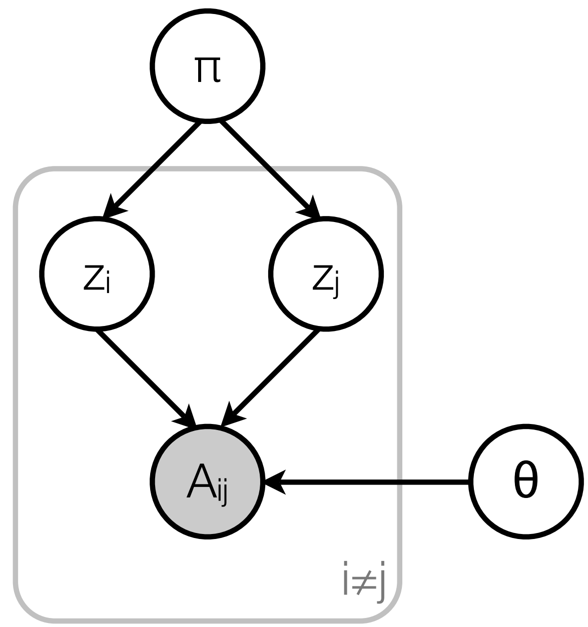


proposal (only important slide in Part I):

- compute p(D|K) for the “community model”
- use variational methods to infer modules+”scale”

the “community model” / “stochastic block model” (SBM)

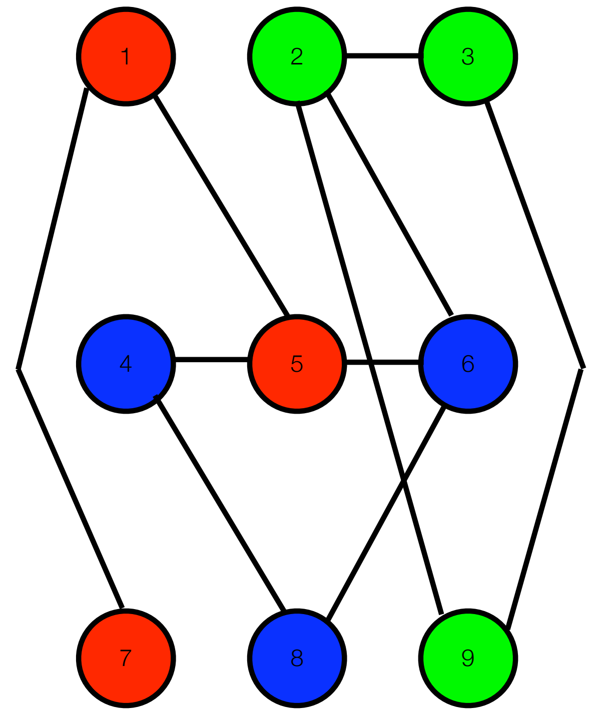
- For each node:
  - **Roll K-sided die** with bias π to determine z_i=1,...,K, the (unobserved) module assignment for i^{th} node
- For each pair of nodes (i,j):
  - If $z_i = z_j$, **flip “in community” coin** with bias θ_c to determine edge A_{ij}
  - If $z_i \neq z_j$, **flip “between communities” coin** with bias θ_d to determine edge A_{ij}



Stochastic block models (Holland, Laskey, Leinhardt 1983; Wang and Wong, 1987)

the "community model" / "stochastic block model" (SBM)

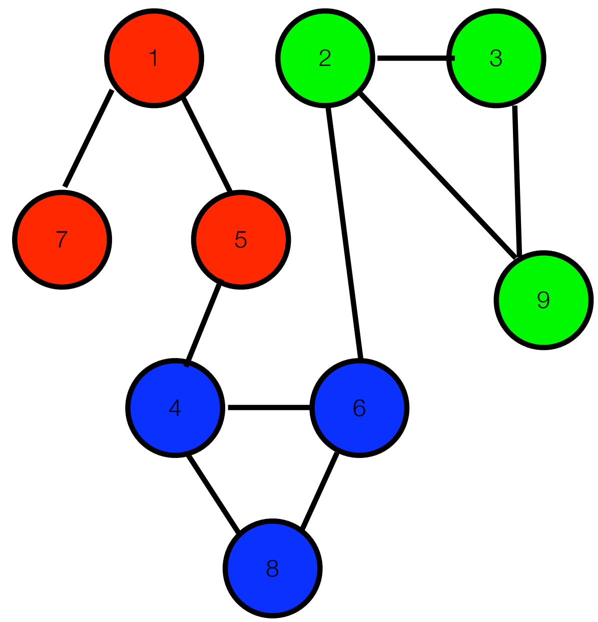
- For each node:
  - **Roll K-sided die** with bias π to determine $z_i=1,\ldots,K$, the (unobserved) module assignment for i^{th} node
- For each pair of nodes (i,j):
  - If $z_i=z_j$, **flip "in community" coin** with bias θ_c to determine edge A_{ij}
  - If $z_i \neq z_j$, **flip "between communities" coin** with bias θ_d to determine edge A_{ij}



Stochastic block models (Holland, Laskey, Leinhardt 1983; Wang and Wong, 1987)

the “community model” / “stochastic block model” (SBM)

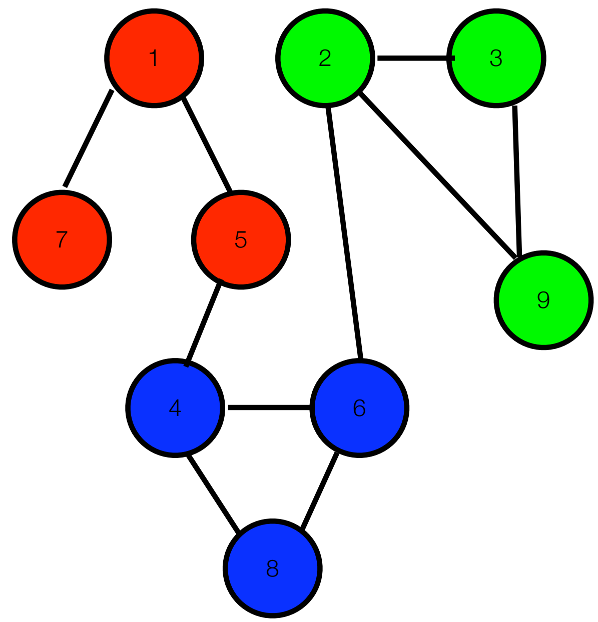
- For each node:
  - **Roll K-sided die** with bias π to determine $z_i=1,\ldots,K$, the (unobserved) module assignment for i^{th} node
- For each pair of nodes (i,j):
  - If $z_i=z_j$, **flip “in community” coin** with bias θ_c to determine edge A_{ij}
  - If $z_i \neq z_j$, **flip “between communities” coin** with bias θ_d to determine edge A_{ij}



Stochastic block models (Holland, Laskey, Leinhardt 1983; Wang and Wong, 1987)

the “community model” / “stochastic block model” (SBM)

- For each node:
  - **Roll K-sided die** with bias π to determine $z_i=1,\ldots,K$, the (unobserved) module assignment for i^{th} node
- For each pair of nodes (i,j):
  - If $z_i=z_j$, **flip “in community” coin** with bias θ_c to determine edge A_{ij}
  - If $z_i \neq z_j$, **flip “between communities” coin** with bias θ_d to determine edge A_{ij}



Stochastic block models (Holland, Laskey, Leinhardt 1983; Wang and Wong, 1987)

the “community model” (math)

Die rolling, coin flipping, and priors:

$$p(\vec{z}|\vec{\pi}) \equiv \prod_{\mu=1}^{K} \pi_\mu^{n_\mu}$$

$$p(\mathbf{A}|\vec{z},\vec{\pi},\vec{\theta}) \equiv \theta_c^{c_+}(1-\theta_c)^{c_-}\theta_d^{d_+}(1-\theta_d)^{d_-}$$

$$p(\vec{\theta}) \equiv \mathcal{B}(\theta_c;\tilde{c}_{+_0},\tilde{c}_{-_0})\mathcal{B}(\theta_d;\tilde{d}_{+_0},\tilde{d}_{-_0})$$

$$p(\vec{\pi}) \equiv \mathcal{D}(\vec{\pi};\tilde{\vec{n}})$$

where counts are:

edges within modules $$c_+ \equiv \sum_{i,j} A_{ij}\delta_{z_i,z_j}$$

non-edges within modules $$c_- \equiv \sum_{i,j}(1-A_{ij})\delta_{z_i,z_j}$$

edges between modules $$d_+ \equiv \sum_{i,j} A_{ij}(1-\delta_{z_i,z_j})$$

non-edges between modules $$d_- \equiv \sum_{i,j}(1-A_{ij})(1-\delta_{z_i,z_j})$$

nodes in each module $$n_\mu \equiv \sum_{i=1}^{N}\delta_{z_i,\mu}$$

stochastic block models (Holland, Laskey, Leinhardt 1983)

the "community model" (math)

Die rolling, coin flipping, and priors:

$$p(\vec{z}|\vec{\pi}) \equiv \prod_{\mu=1}^{K} \pi_\mu^{n_\mu}$$

$$p(\mathbf{A}|\vec{z},\vec{\pi},\vec{\theta}) \equiv \theta_c^{c_+}(1-\theta_c)^{c_-}\theta_d^{d_+}(1-\theta_d)^{d_-}$$

$$p(\vec{\theta}) \equiv \mathcal{B}(\theta_c;\tilde{c}_{+_0},\tilde{c}_{-_0})\mathcal{B}(\theta_d;\tilde{d}_{+_0},\tilde{d}_{-_0})$$

$$p(\vec{\pi}) \equiv \mathcal{D}(\vec{\pi};\tilde{\vec{n}})$$

where counts are:

edges within modules $$c_+ \equiv \sum_{i,j} A_{ij}\delta_{z_i,z_j}$$

non-edges within modules $$c_- \equiv \sum_{i,j} (1-A_{ij})\delta_{z_i,z_j}$$

edges between modules $$d_+ \equiv \sum_{i,j} A_{ij}(1-\delta_{z_i,z_j})$$

non-edges between modules $$d_- \equiv \sum_{i,j} (1-A_{ij})(1-\delta_{z_i,z_j})$$

nodes in each module $$n_\mu \equiv \sum_{i=1}^{N} \delta_{z_i,\mu}$$

stochastic block models (Holland, Laskey, Leinhardt 1983)

the “community model” (math)

- Die rolling, coin flipping <-> infinite-range spin-glass Potts model:

$$\mathcal{H} \equiv -\ln p(\mathbf{A}, \vec{z}|\vec{\pi}, \vec{\theta}) = -\sum_{i,j}(J_L A_{ij} - J_G)\delta_{z_i,z_j} + \sum_{\mu=1}^{K} h_\mu \sum_{i=1}^{N} \delta_{z_i,\mu}$$

$$\begin{aligned} J_G &\equiv \ln \vartheta_c/\vartheta_d \\ J_L &\equiv \ln(1-\vartheta_d)/(1-\vartheta_c) + J_G \\ h_\mu &\equiv -\ln \pi_\mu \end{aligned}$$

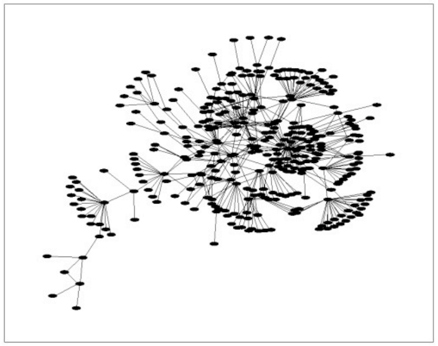
- Infer *distributions* over spin assignments, coupling constants, and chemical potentials and find number of occupied spin states

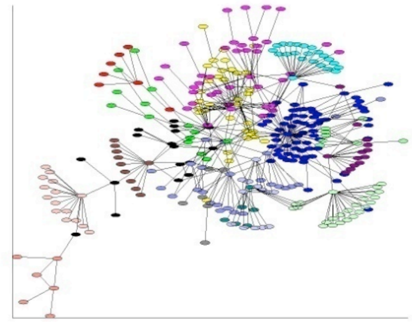
$$p(A|K) = \sum_{\vec{z}} \int d\vec{\theta} \int d\vec{\pi}\; p(\mathbf{A}, \vec{z}, \vec{\pi}, \vec{\theta}) = \sum_{\vec{z}} \int d\vec{\theta} \int d\vec{\pi}\; e^{-\mathcal{H}} p(\vec{\theta}) p(\vec{\pi})$$ ⟵

Can do integrals, but sum is intractable, $O(K^N)$; use mean-field variational technique

Extends Newman (2004, 2006), Hastings (2006), Bornholdt & Reichardt (2006)

inferring modules:

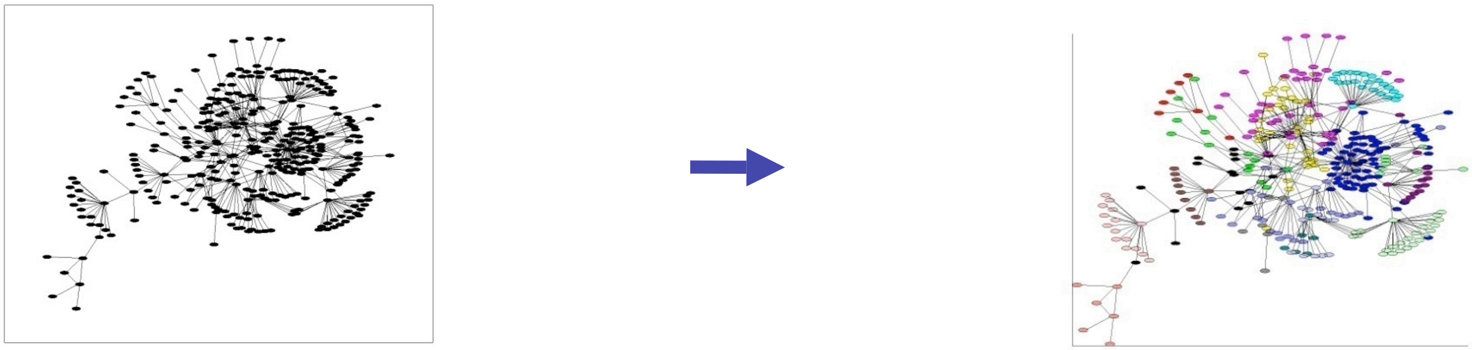




proposal (only important slide in Part I):

- compute p(D|K) for the "community model"
- use variational   inference to infer modules+"scale"

inferring modules:



proposal (only important slide in Part I):

- compute p(D|K)* for the “community model”**
- use variational*** inference to infer modules+”scale”

* a.k.a a disorder-averaged partition function

** a.k.a. a spin glass

*** a.k.a. MFT w/test hamiltonian
(Feynman, Bogliubov, Mackay/Jordan/Beal/Gharamani)

“variational methods”

- Gibbs’/Jensen’s inequality (log of expected value bounds expected value of log) for *any* distribution q

$$
\begin{aligned}
-\ln p(\mathbf{A}|K) &= -\ln \sum_{\vec{z}} \int d\vec{\theta} \int d\vec{\pi}\; p(\mathbf{A},\vec{z},\vec{\pi},\vec{\theta}|K) \\
&= -\ln \sum_{\vec{z}} \int d\vec{\theta} \int d\vec{\pi}\; q(\vec{z},\vec{\pi},\vec{\theta}) \frac{p(\mathbf{A},\vec{z},\vec{\pi},\vec{\theta}|K)}{q(\vec{z},\vec{\pi},\vec{\theta})} \\
&\le \underbrace{-\sum_{\vec{z}} \int d\vec{\theta} \int d\vec{\pi}\; q(\vec{z},\vec{\pi},\vec{\theta}) \ln \frac{p(\mathbf{A},\vec{z},\vec{\pi},\vec{\theta}|K)}{q(\vec{z},\vec{\pi},\vec{\theta})}}_{F\{q;A\}}
\end{aligned}
$$

Variational Bayes (MacKay, Jordan, Ghahramani, Jaakola, Saul 1999; cf. Feynman 1972)

“Mean field theory” = “factorized test distribution” = “additive test Hamiltonian”

- F is a functional of q; find approximation to posterior by optimizing approximation to evidence
- Take q(z, π, θ)=q(z)q(π)q(θ); $Q_{i\mu}$ is probability node i in module μ

$$
\begin{aligned}
F\{q;\mathbf{A}\} &= -\ln\frac{\mathcal{Z}_{\vec{\pi}}\mathcal{Z}_c\mathcal{Z}_d}{\tilde{\mathcal{Z}}_{\vec{\pi}}\tilde{\mathcal{Z}}_c\tilde{\mathcal{Z}}_d} + \sum_{\mu=1}^{K}\sum_{i=1}^{N} Q_{i\mu}\ln Q_{i\mu} \\
&\quad -(\tilde{c}_+ - (\langle c_+\rangle + \tilde{c}_{+_0}))\langle\ln\theta_c\rangle \\
&\quad -(\tilde{c}_- - (\langle c_-\rangle + \tilde{c}_{+_0}))\langle\ln(1-\theta_c)\rangle \\
&\quad -(\tilde{d}_+ - (\langle d_+\rangle + \tilde{d}_{+_0}))\langle\ln\theta_d\rangle \\
&\quad -(\tilde{d}_- - (\langle d_-\rangle + \tilde{d}_{-_0}))\langle\ln(1-\theta_d)\rangle \\
&\quad -\sum_{\mu=1}^{K}(\tilde{n}_\mu - (\langle n_\mu\rangle + \tilde{n}_{\mu_0}))\langle\ln\pi_\mu\rangle
\end{aligned}
$$

where expected counts are:

$$
\begin{aligned}
\langle c_+\rangle &= \frac{1}{2}Tr(\mathbf{Q}^T\mathbf{A}\mathbf{Q}) \\
\langle c_-\rangle &= \frac{1}{2}Tr(\mathbf{Q}^T\bar{\mathbf{A}}\mathbf{Q}) \\
\langle d_+\rangle &= M - \langle c_+\rangle \\
\langle d_-\rangle &= C - M - \langle c_-\rangle \\
\langle n_\mu\rangle &= \sum_{j=1}^{N} Q_{j\mu}
\end{aligned}
$$

Inferring modules to maximize *evidence* P(D|K)

Initialization: Initialize the N-by-K matrix $\mathbf{Q} = \mathbf{Q}_0$ and set pseudocounts $\tilde{c}_{+_0}, \tilde{c}_{-_0}, \tilde{d}_{+_0}, \tilde{d}_{-_0}$, and $\tilde{n}_{\mu_0}$.

Main loop: Until convergence in $F\{q;\mathbf{A}\}$,

1. update $\tilde{c}_\pm$'s, $\tilde{d}_\pm$'s and $\tilde{n}_\mu$'s from expected counts and pseudocounts

$$
\begin{aligned}
\tilde{c}_+ &= \frac{1}{2}Tr(\mathbf{Q}^T\mathbf{A}\mathbf{Q}) + \tilde{c}_{+_0} \\
\tilde{c}_- &= \frac{1}{2}Tr(\mathbf{Q}^T\bar{\mathbf{A}}\mathbf{Q}) + \tilde{c}_{-_0} \\
\tilde{d}_+ &= M - \langle c_+ \rangle + \tilde{d}_{+_0} \\
\tilde{d}_- &= C - M - \langle c_- \rangle + \tilde{d}_{-_0} \\
\tilde{n}_\mu &= \sum_{j=1}^{N} Q_{j\mu} + \tilde{n}_{\mu_0},
\end{aligned}
$$

where $\bar{\mathbf{A}}$ is the logical negation of $\mathbf{A}$, $C = N(N-1)/2$, and $M = \frac{1}{2}\sum_{i,j} A_{ij}$;

2. update expected value of coupling constants

$$
\begin{aligned}
\langle J_L \rangle &= \psi(\tilde{c}_+) - \psi(\tilde{c}_-) - \psi(\tilde{d}_+) + \psi(\tilde{d}_-) \\
\langle J_G \rangle &= \psi(\tilde{d}_-) - \psi(\tilde{d}_+ + \tilde{d}_-) - \psi(\tilde{c}_-) + \psi(\tilde{c}_+ + \tilde{c}_-),
\end{aligned}
$$

where $\psi(x)$ is the digamma function;

3. update $\mathbf{Q}$ as

$$
\mathbf{Q} \leftarrow \frac{1}{\mathcal{Z}} e^{\langle J_L \rangle \mathbf{A}\mathbf{Q} - \langle J_G \rangle \langle \vec{\mathbf{n}} \rangle + \langle \ln \vec{\pi} \rangle}
$$

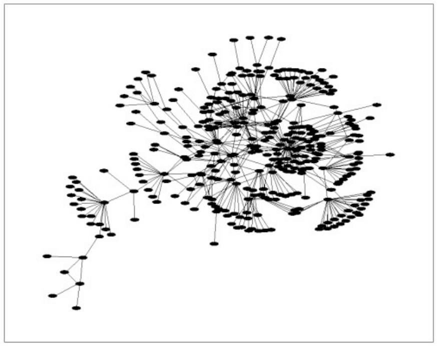
where $\langle \ln \pi_\mu \rangle = \psi(\tilde{n}_\mu) - \psi(\sum_\mu \tilde{n}_\mu)$ and $\mathcal{Z}$ is set by the normalization $\sum_\mu Q_{i\mu} = 1$;

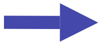
4. calculate the updated optimal free energy

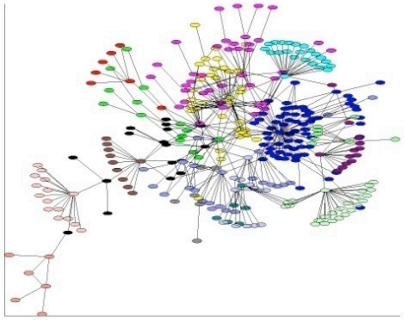
$$
F\{q;\mathbf{A}\} = -\ln \frac{\mathcal{Z}_c \mathcal{Z}_d \mathcal{Z}_{\vec{\pi}}}{\tilde{\mathcal{Z}}_c \tilde{\mathcal{Z}}_d \tilde{\mathcal{Z}}_{\vec{\pi}}} + \sum_{\mu=1}^{K}\sum_{i=1}^{N} Q_{i\mu} \ln Q_{i\mu}.
$$

Inferring modules to maximize *evidence* P(D|K)

Part 1: inferring modules

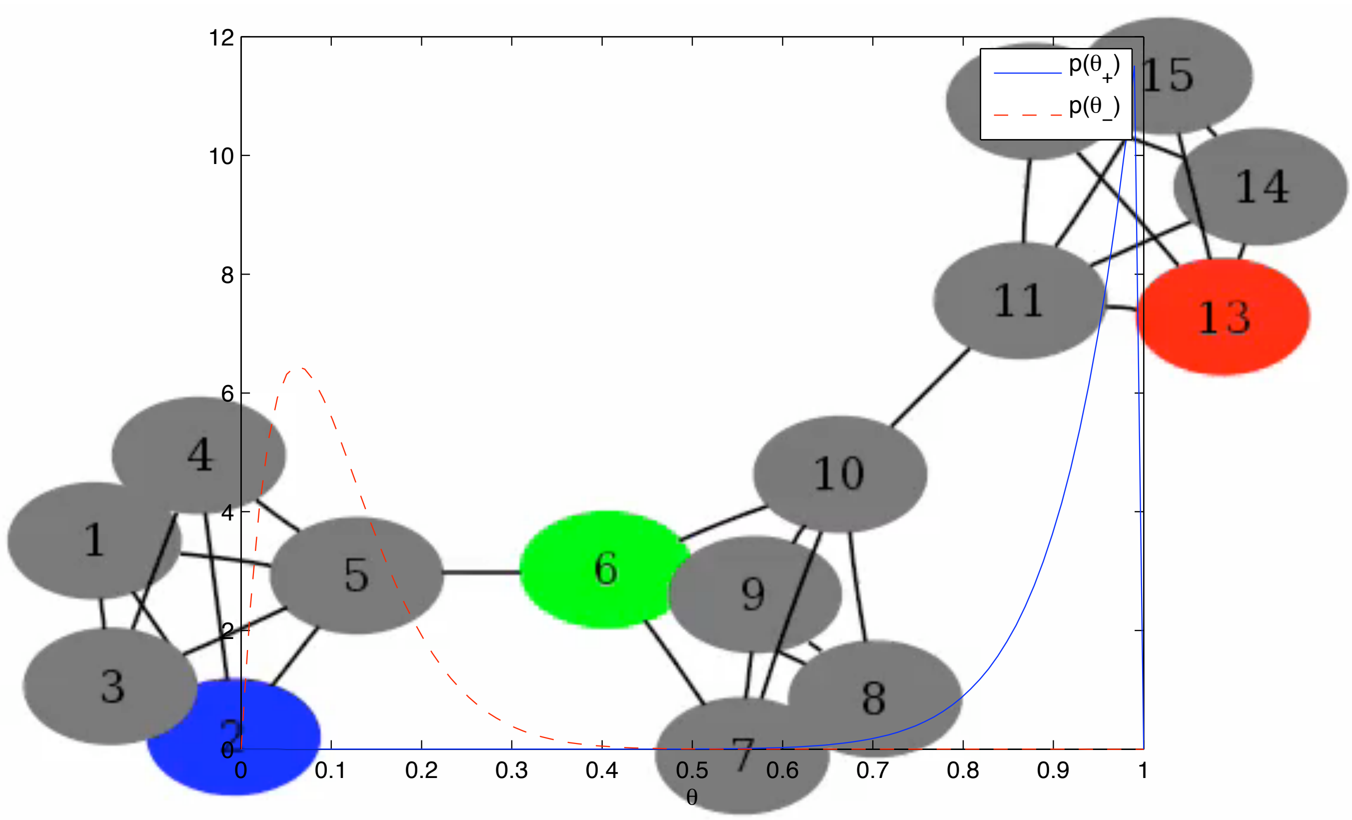






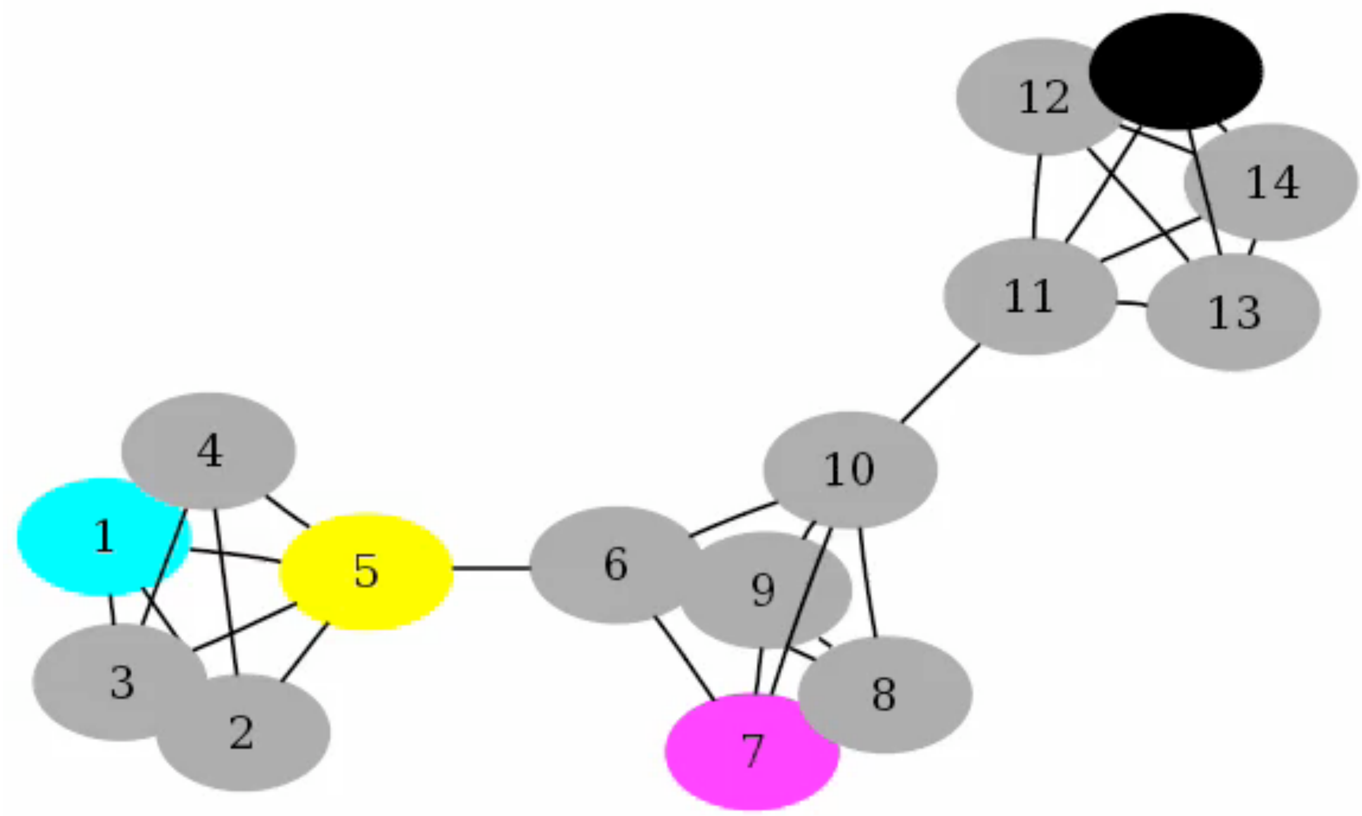
- pseudohistory
- problems and solutions
- algorithm
- results
  - fake data
  - real data
  - comparison w/other approaches

Validation: Toy Graph (K=3)



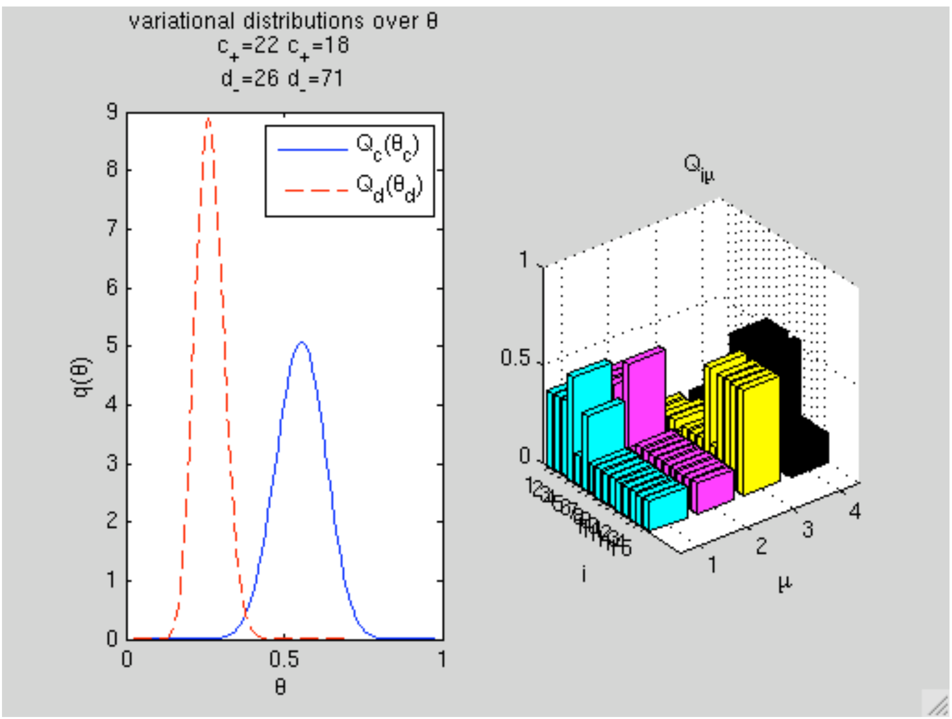
Validation: Toy Graph (K=4)

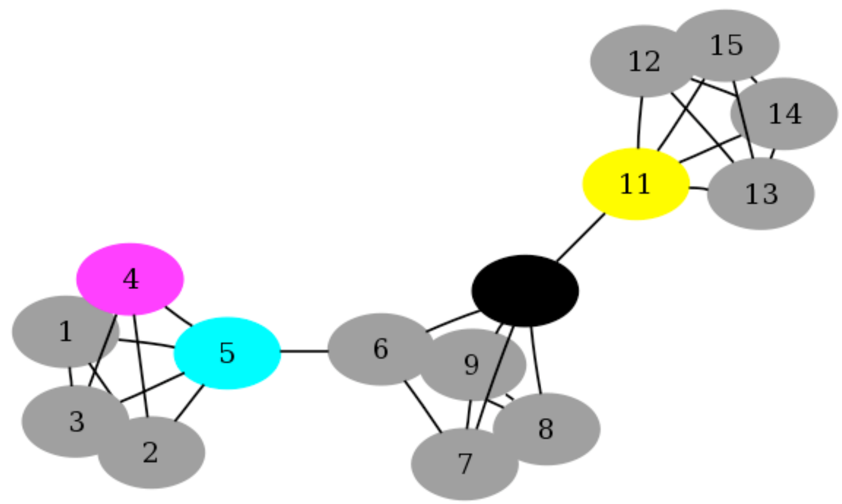
- Automatic complexity control: probability of occupation for extraneous modules goes to zero



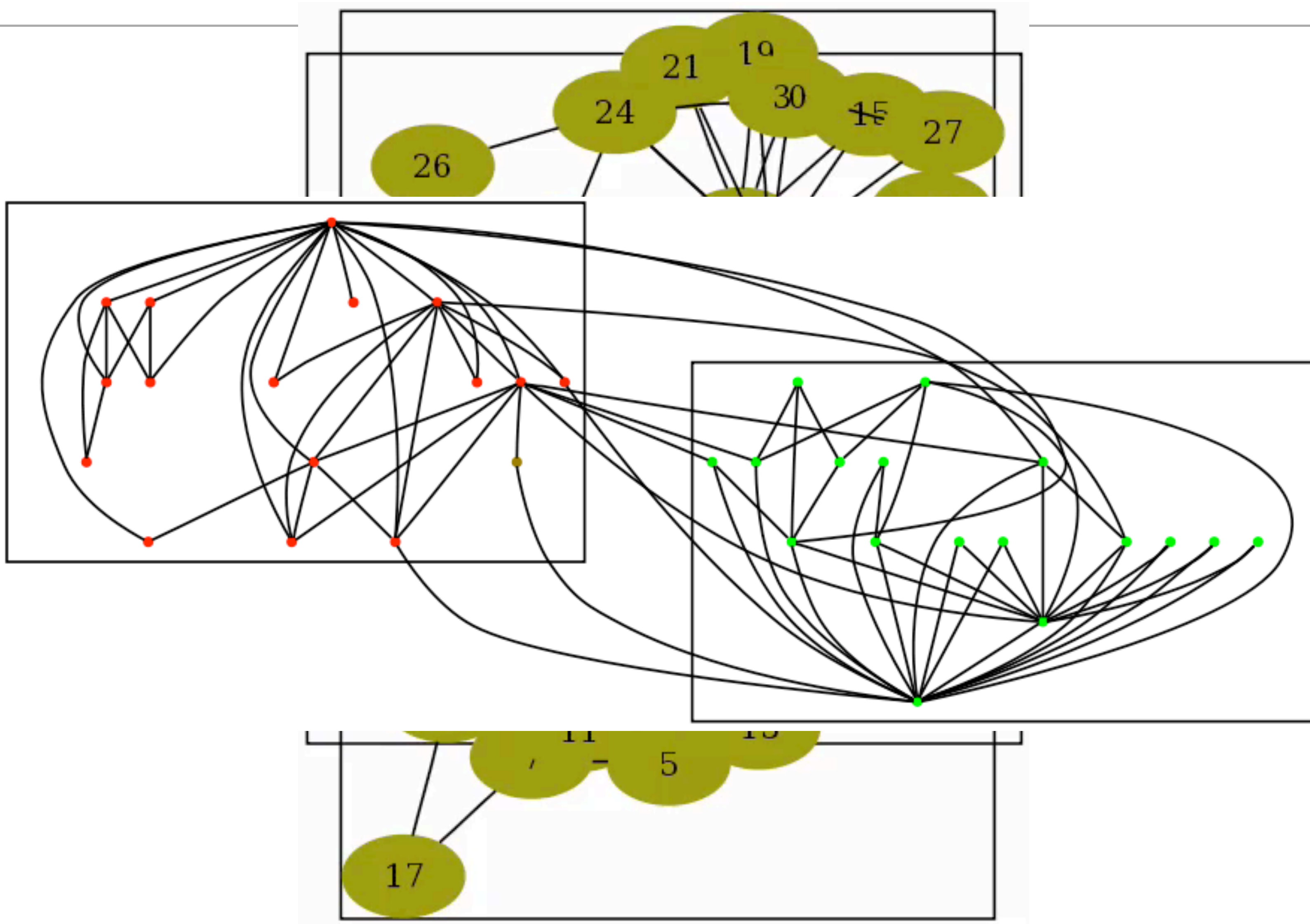
Validation: Toy Graph (K=4)

- Automatic complexity control: probability of occupation for extraneous modules goes to zero

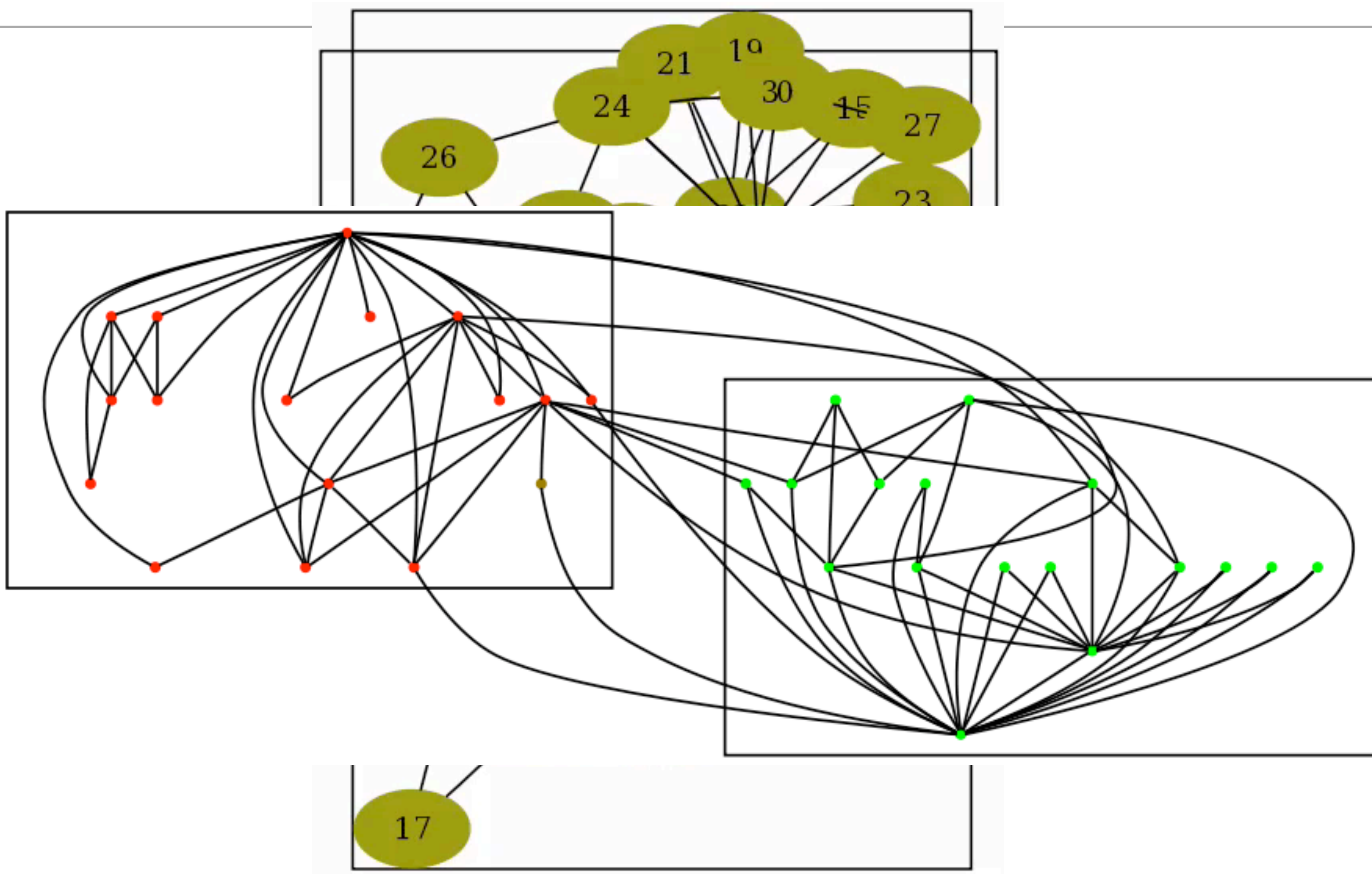




Validation: Zachary's karate network

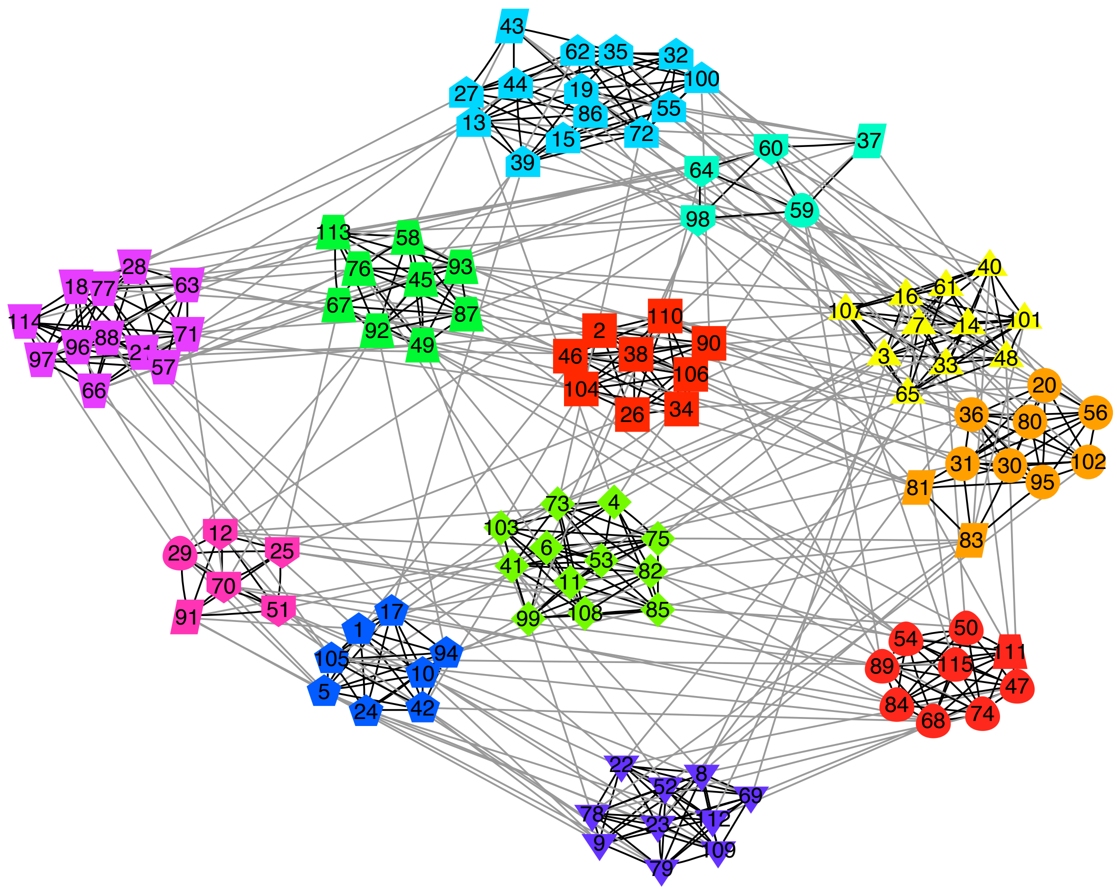


Popular sanity check: Zachary's karate network



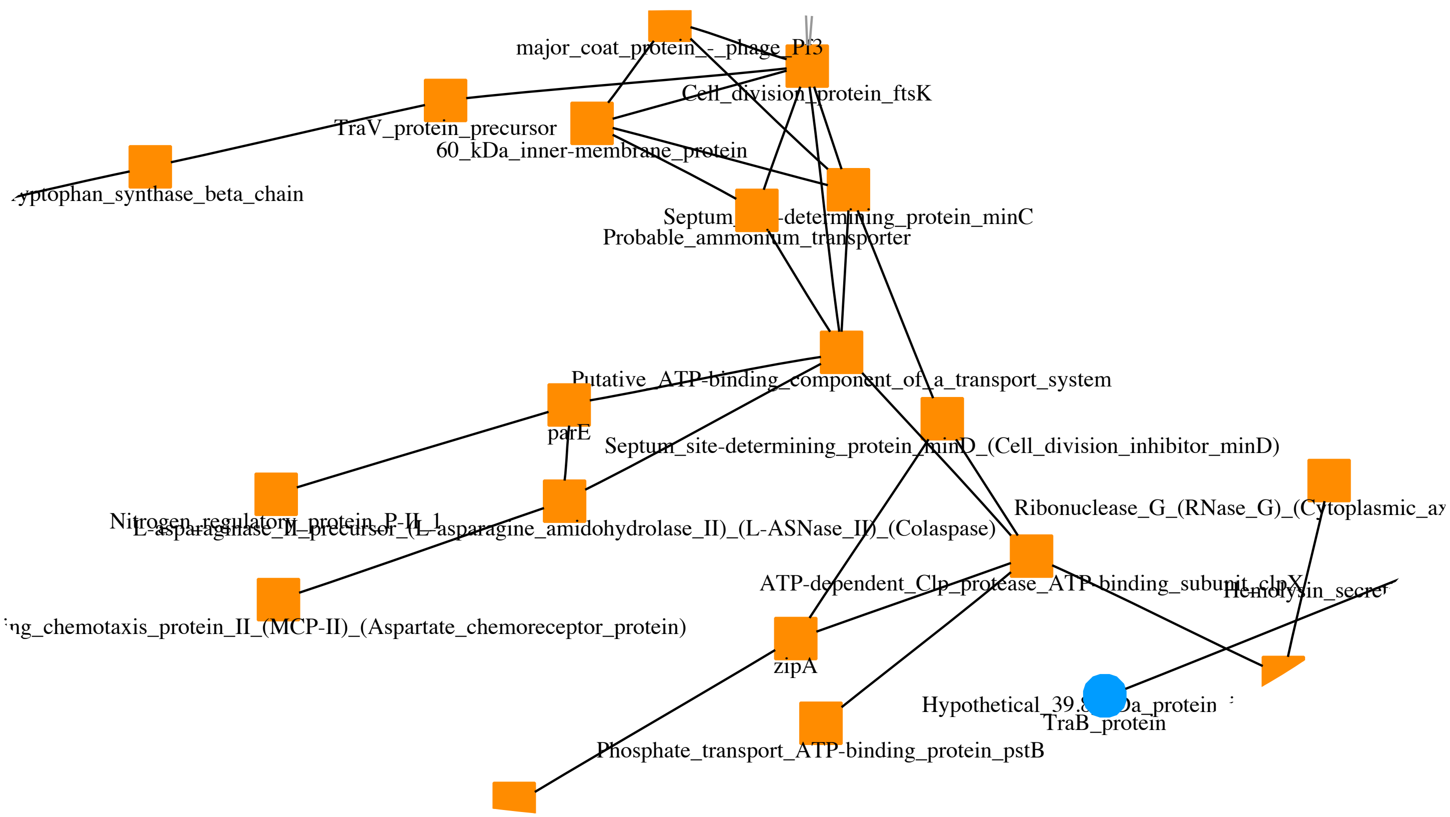
note: learns K=2

Popular sanity check: american football conferences



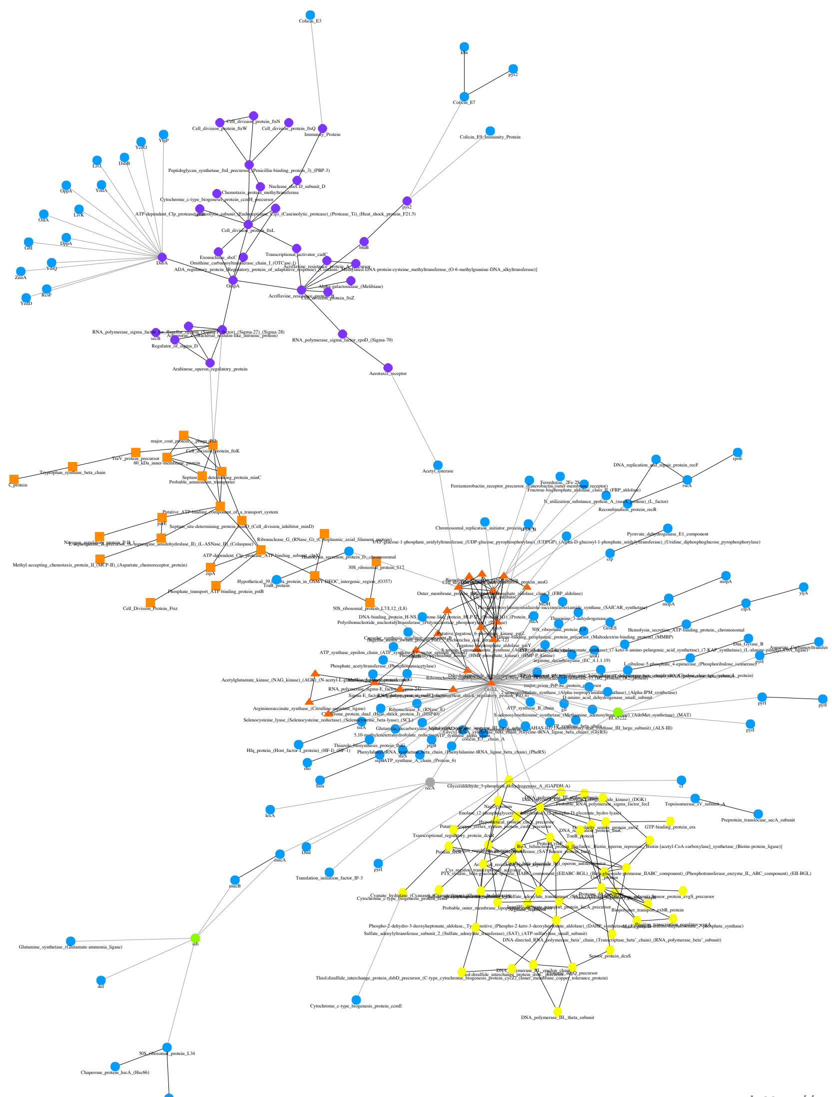
note: learns K=12

Application: E. Coli protein-protein network (cell divison)



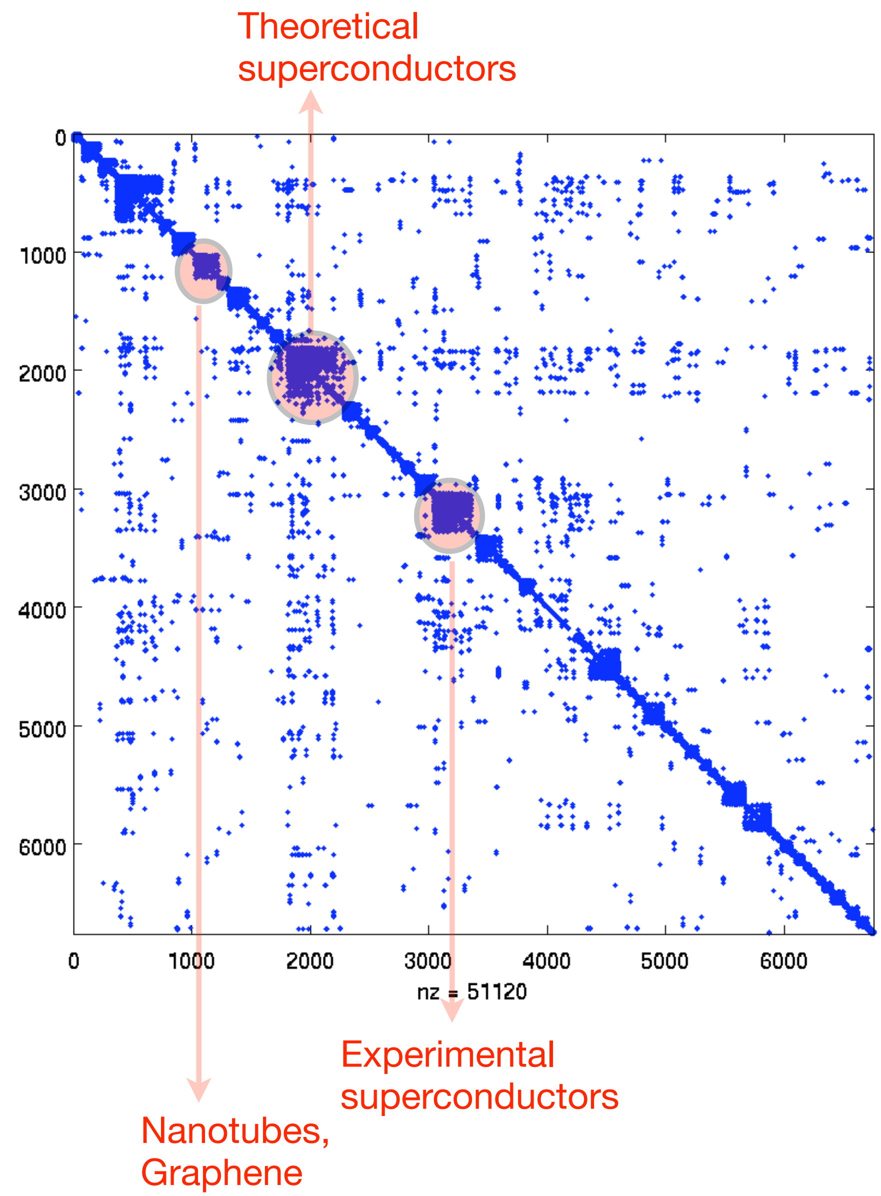
http://www.cs.purdue.edu/homes/koyuturk/mawish/

Application: E. Coli protein-protein network

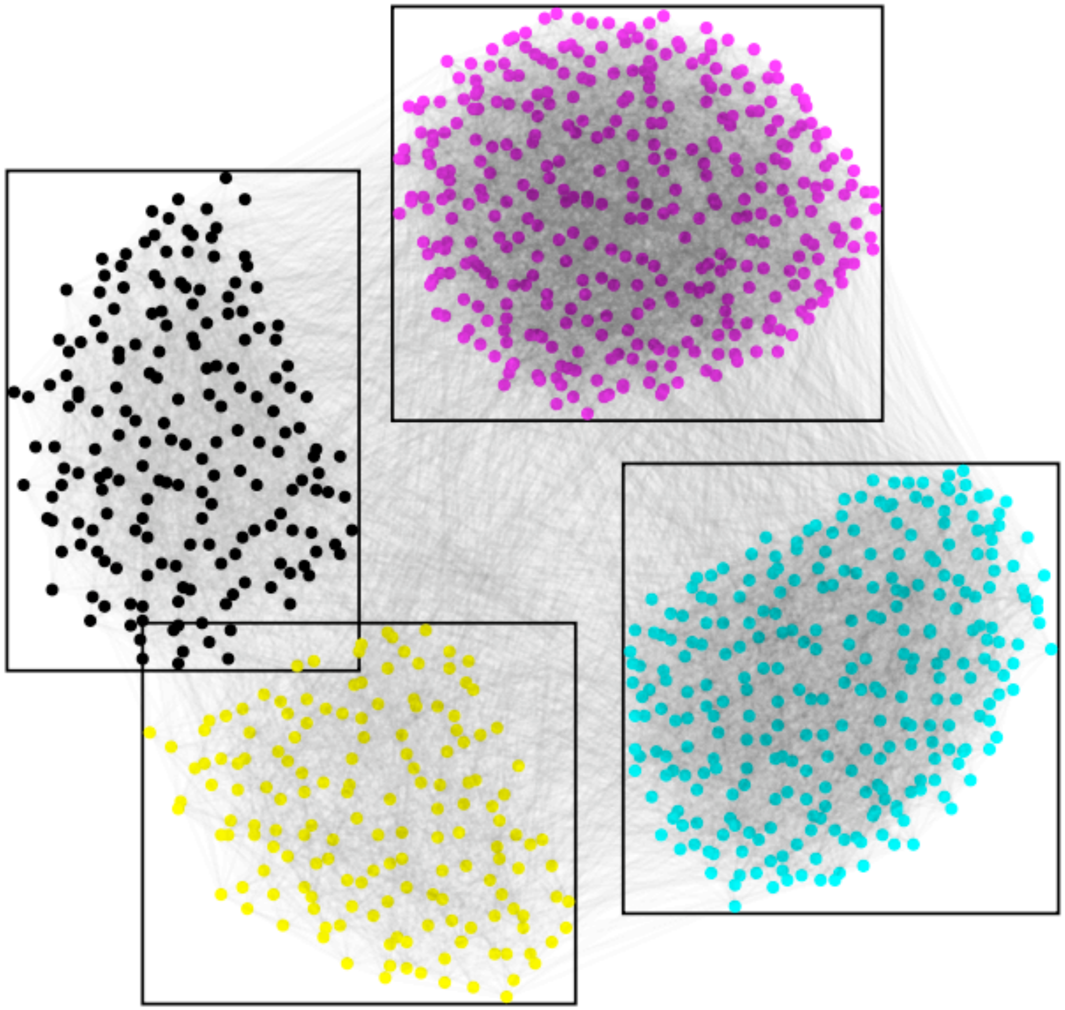


http://www.cs.purdue.edu/homes/koyuturk/mawish/

APS March Meeting 2008 co-authorship network

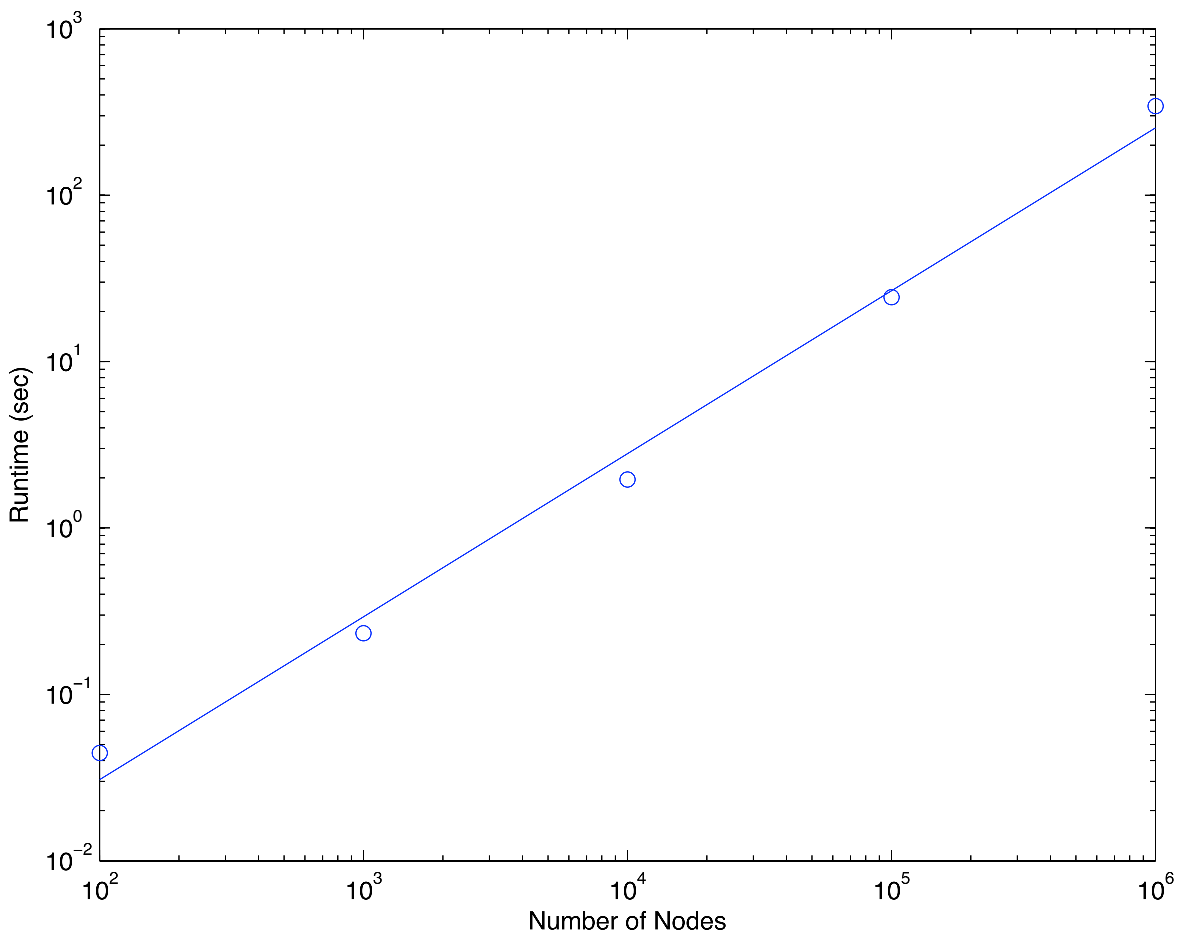


Validation: Large-scale network



Validation: Runtime

- Main loop runtime for 10^6 nodes in ~100 seconds



Validation: Complexity control

- Comparison of our method (VB) to alternative method (ICL, similar to BIC) for synthetic N=60 node networks and K_{True}=3,4,5 modules

$K_{\text{True}}/K_{\text{VB}}$

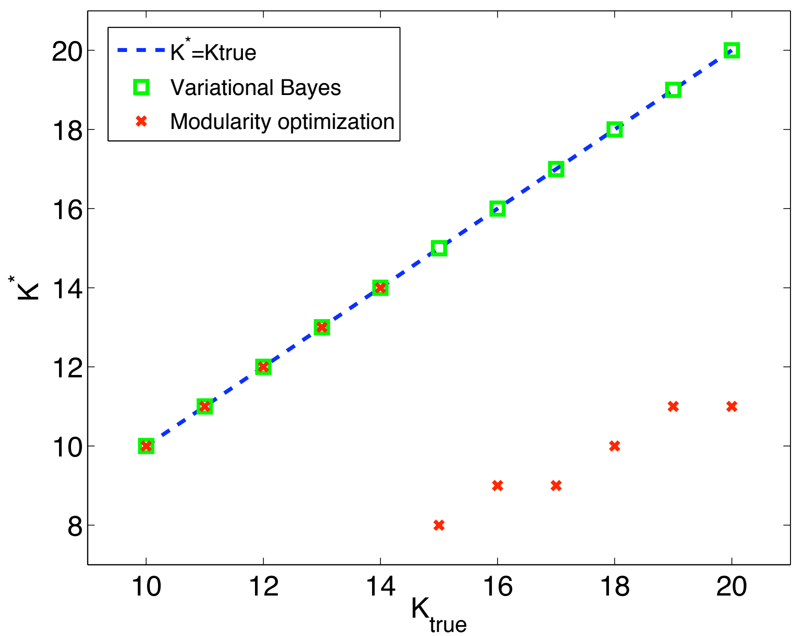
	2	3	4	5	6
3	0	**99**	1	0	0
4	0	0	**90**	10	0
5	0	1	5	**91**	3

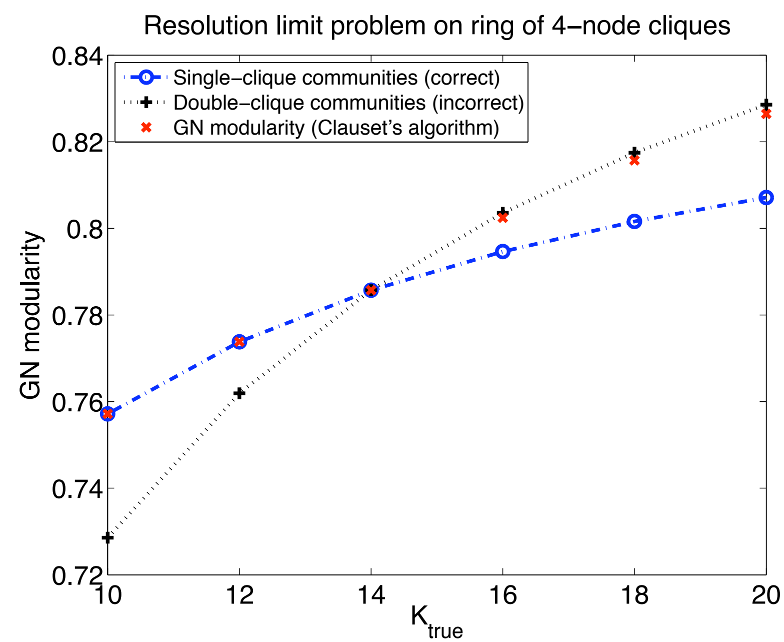
$K_{\text{True}}/K_{\text{ICL}}$

	2	3	4	5	6
3	0	**100**	0	0	0
4	4	25	**71**	0	0
5	26	55	17	**2**	0

- Fast online graph clustering via Erdos-Reyni mixture; Hugo Zanghi and Christophe Ambroise and Vincent Miele; SSB-RR-8

The “resolution limit” problem



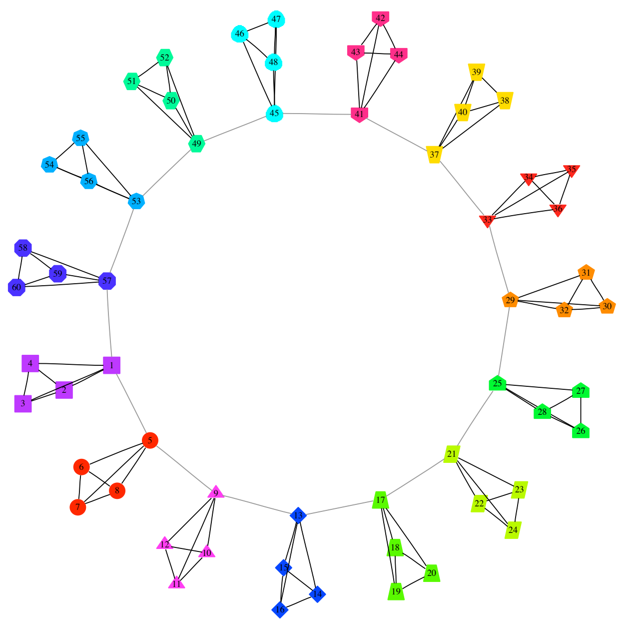


Girvan-Newman modularity or Potts model w/ *fixed parameters* suffers from a resolution limit, where size of detected modules depends on network size

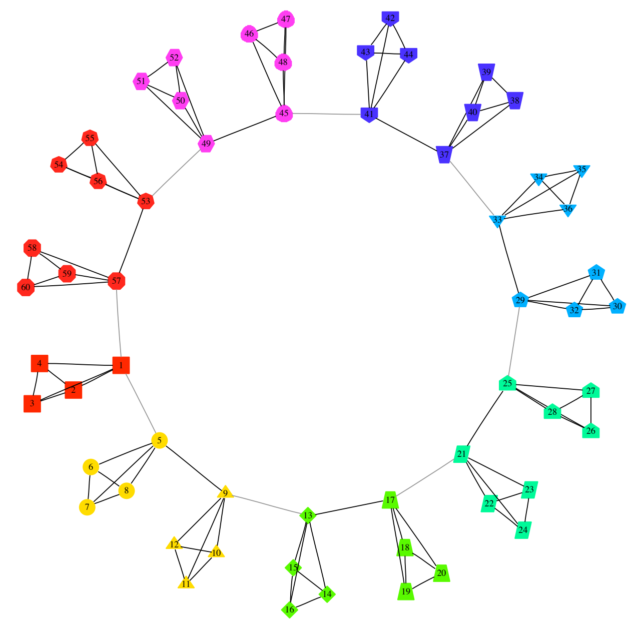
Fortunato et. al. (2007), Kumpula et. al. (2007)

The “resolution limit” problem

Variational Bayesian/MFT approach correctly infers complexity



Variational Bayes



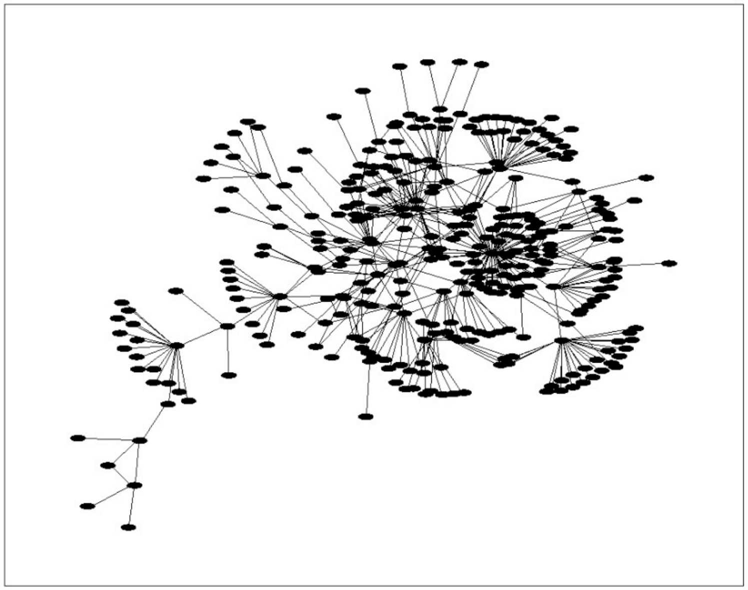
Girvan-Newman
modularity

big ideas

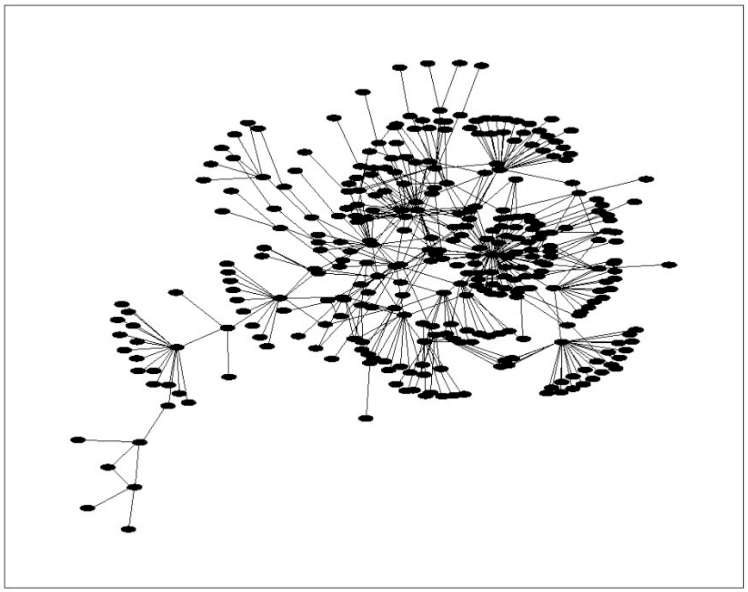
- model didn’t have to be SBM, could have more parameters
- selection could be between entirely different models rather than different K
- if you are willing to tell the data how to behave (generative modeling) you can learn
  - parameters
  - module assignments
  - number of modules
- do you have to tell the data how to behave?

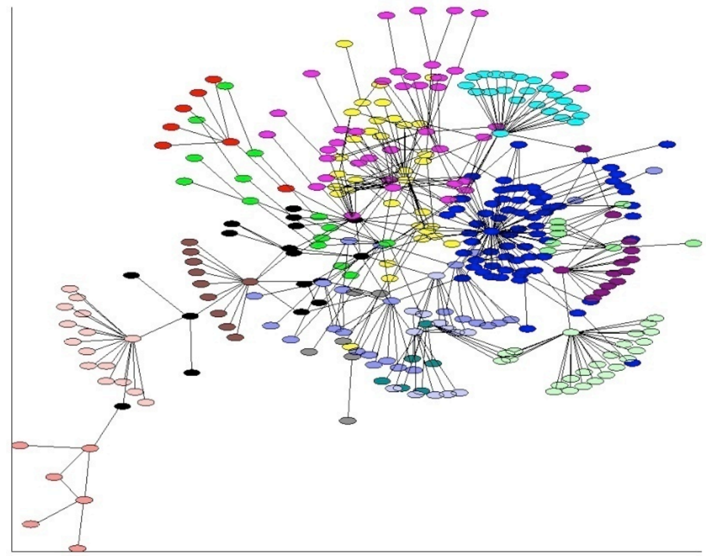
errors?

community detection:

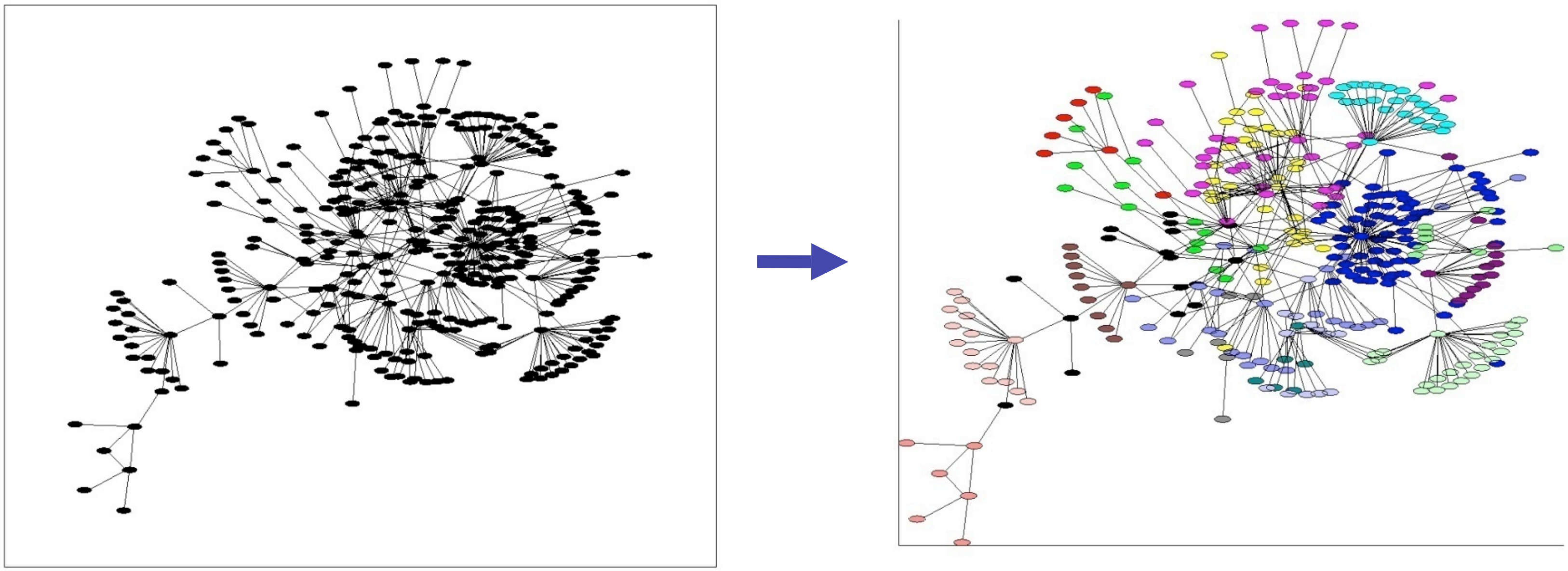


what just happened?

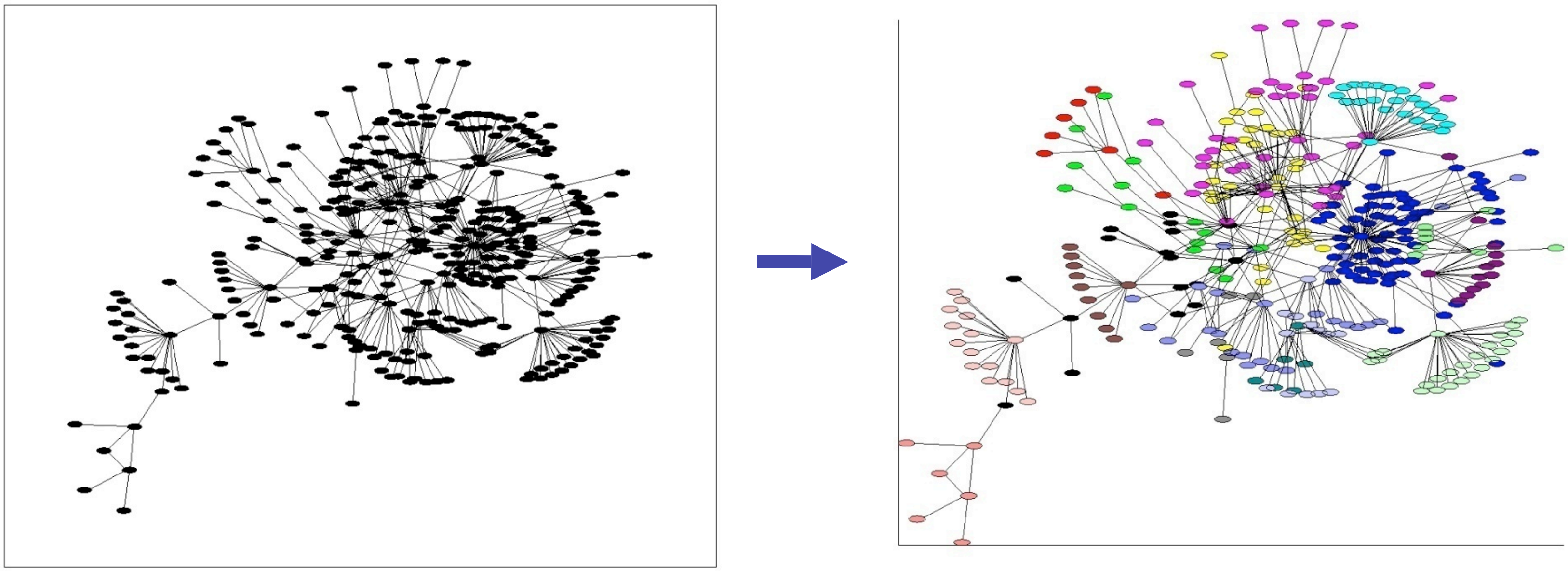




what just happened?



what just happened?



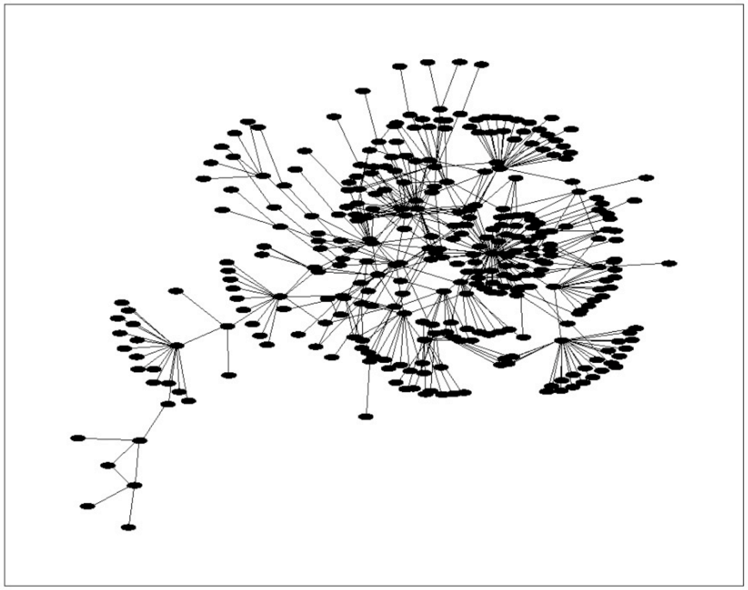
the dominant paradigm:

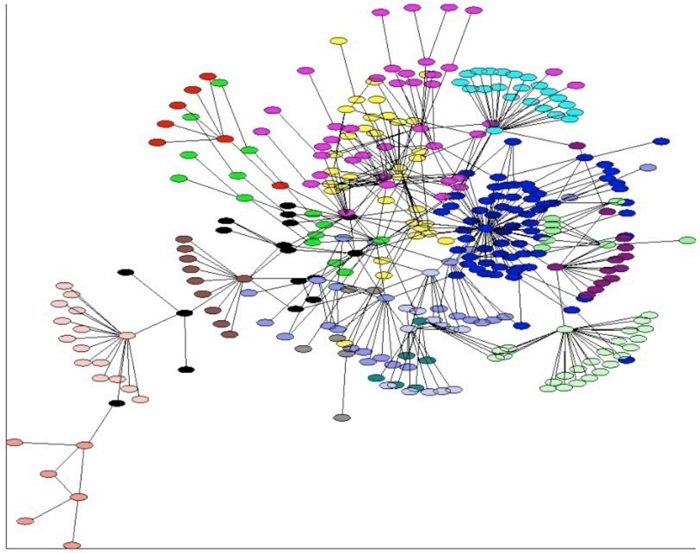
find connected bits , but avoid trivial solution

i. posit p(G) (e.g., configuration model, SBM)

ii. posit regularized cut (cf. Shi+Malik ’99)

what just happened?

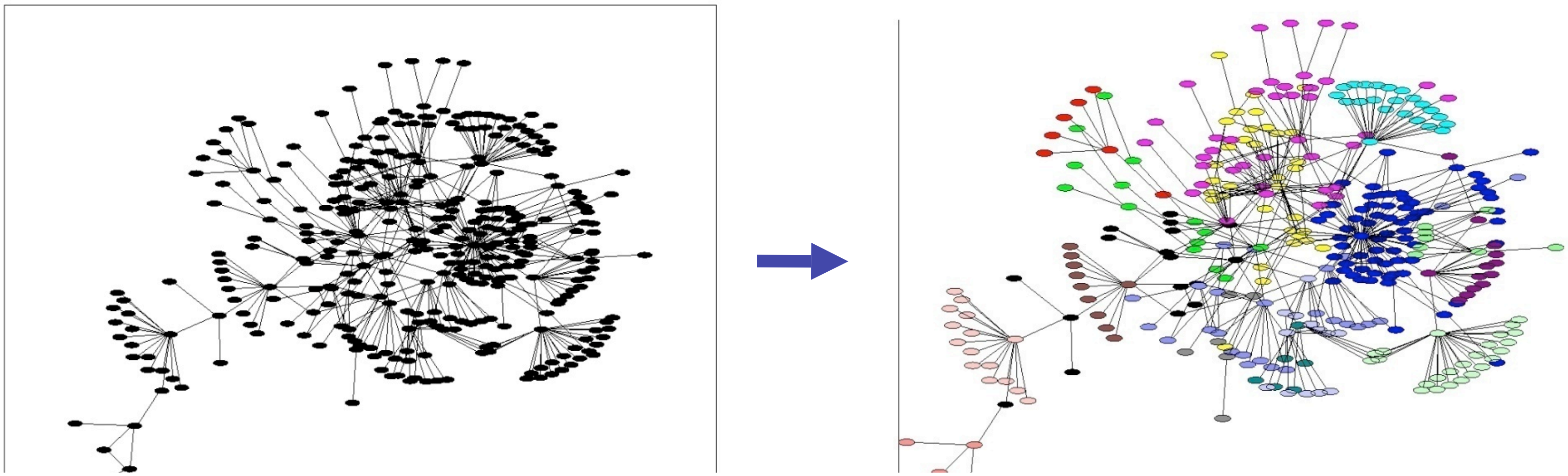




a rethink:

- compression (=summarizing=encoding)

what just happened?



a rethink:

- summarizing/compression/encoding

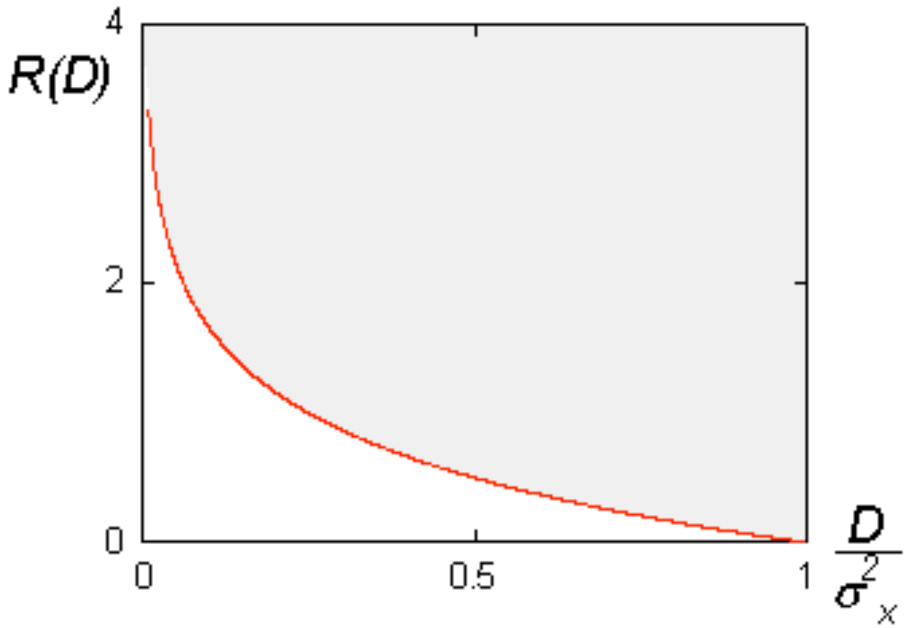
i. does not require p(G)

ii. gives order parameter for graph modularity

the method: “NIB”

what is compression?

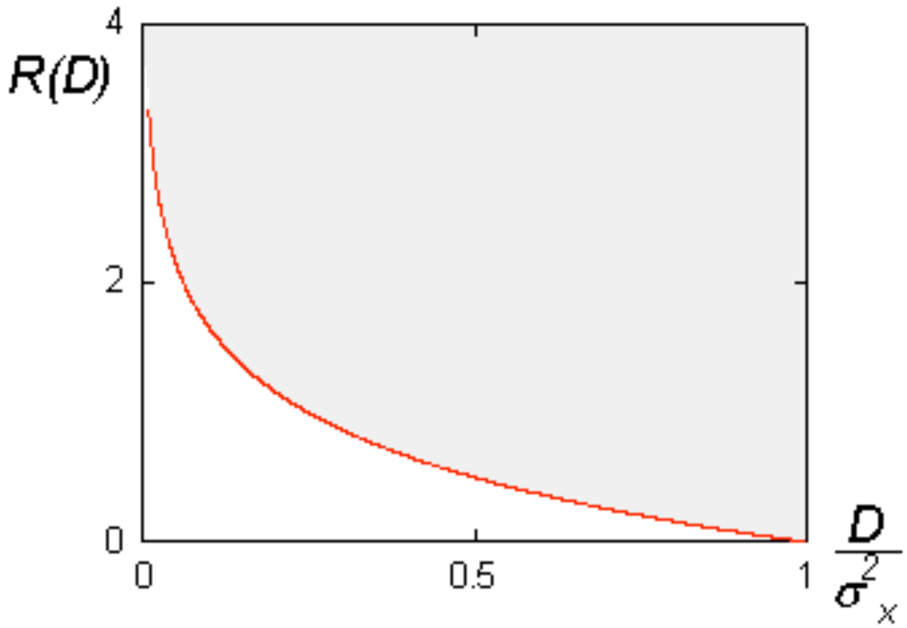
$$\inf_{p(z|x)} \quad I_Q(Z;X) \text{ subject to } D_Q \leq D^*$$ -wikipedia



- shannon's RDT (1948)
- $I = H(X) + H(Z) - H(X,Z)$

what is compression?

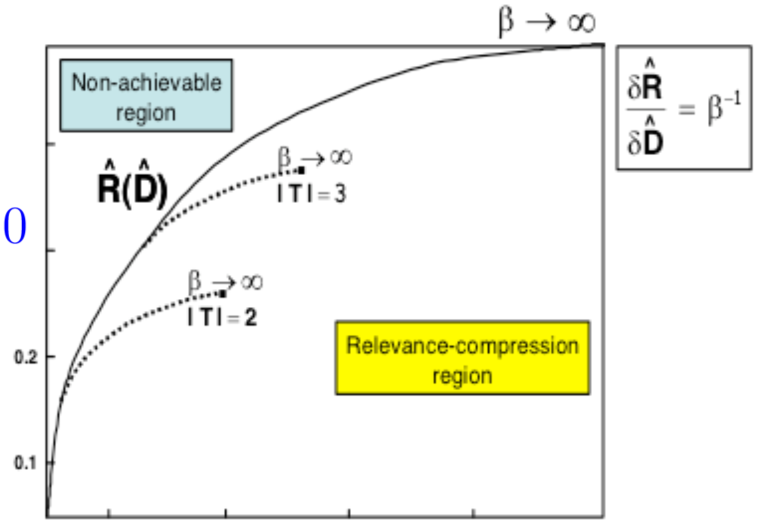
$$\inf_{p(z|x)} \quad I_Q(Z;X) \text{ subject to } D_Q \leq D^*$$ -wikipedia



- shannon's RDT (1948)
- $I = \langle \ln p(x,z)/p(x)p(z) \rangle$

what is relevant compression?

$$\mathcal{E} = I(X,Z) - I(Y,Z) > 0$$



$$\mathcal{S} = H(X) - I(X,Z) > 0$$

given p(x,y), minimize free energy over p(z|x):

$$\mathcal{F} = \mathcal{E} - T\mathcal{S}$$

- @ T=0, “hard” (0-1) clustering
- Tishby, Pereira+Bialek, physics/0004057

information modularity

what is the partition such that, if i tell you only which module r.w. starts in, you lose as little information as possible about location of r.w. later

information modularity

what is the partition such that, if i tell you only which module r.w. starts in, you lose as little information as possible about location of r.w. later

- this is words (interpretable)

information modularity

what is the partition such that, if i tell you only which module Z r.w. starts in, you lose as little information as possible about location Y of r.w. later

- this is words (interpretable)
- this is also math (calculable), since

$$I = \langle \ln p(x,z)/p(x)p(z) \rangle$$

method: interpretation

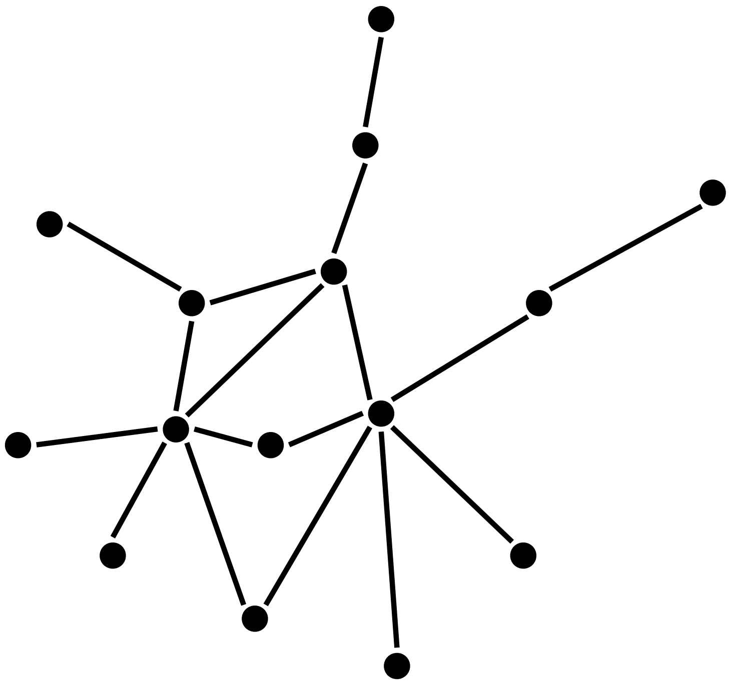
- community = module = cluster = codeword
- modularity = summarizability = structure

algorithm: partition the graph
such that if i only tell you which
module you started in, you lose
as little information as possible
about where you walk to later

information modularity

- Cluster vertices while preserving information about the network structure

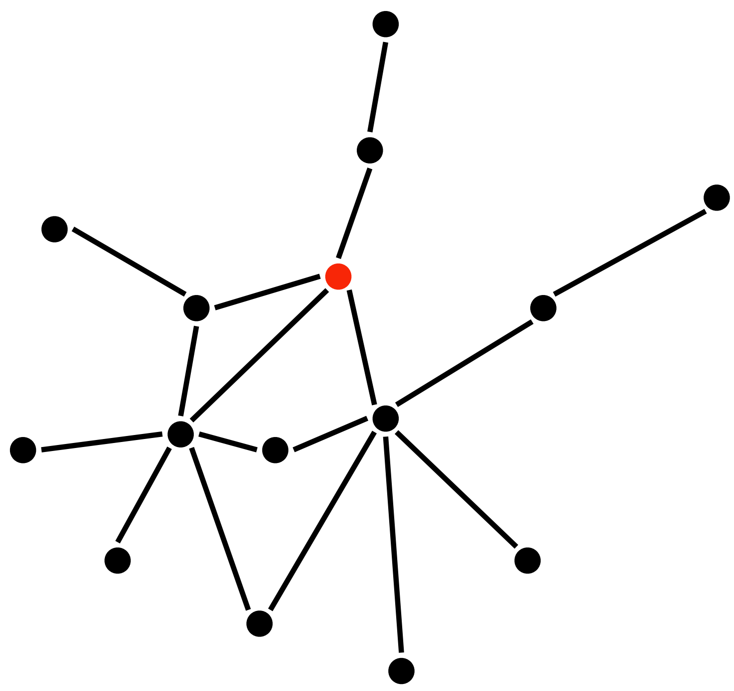
- graph diffusion



information modularity

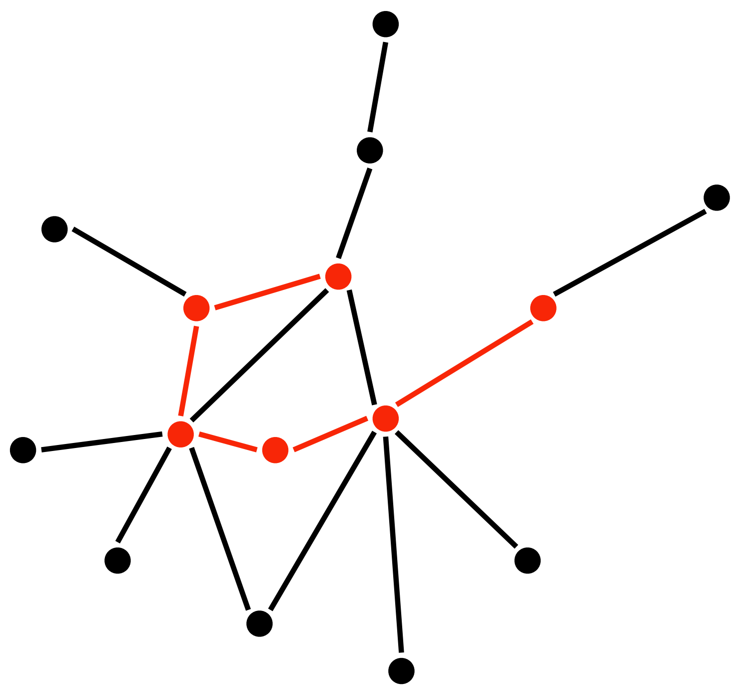
- Cluster vertices while preserving information about the network structure

- graph diffusion



information modularity

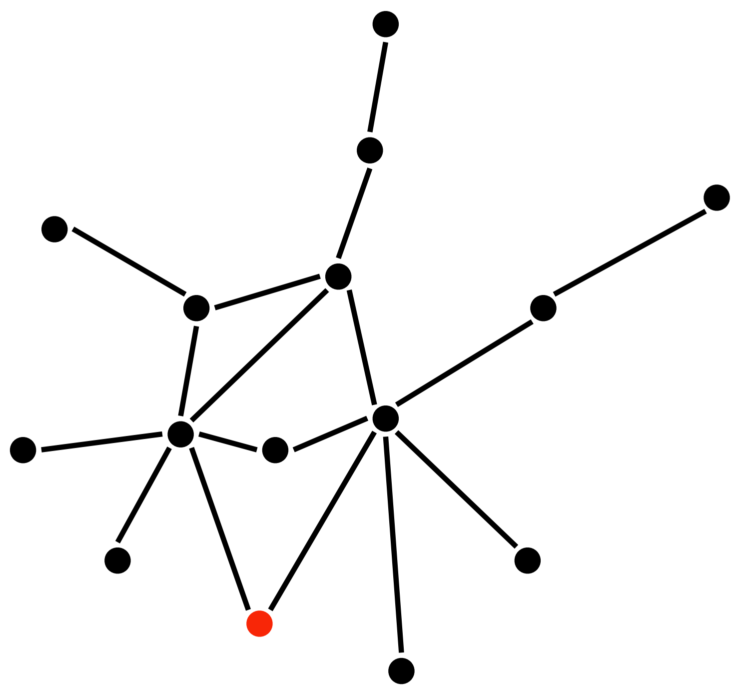
- Cluster vertices while preserving information about the network structure
- graph diffusion



information modularity

- Cluster vertices while preserving information about the network structure

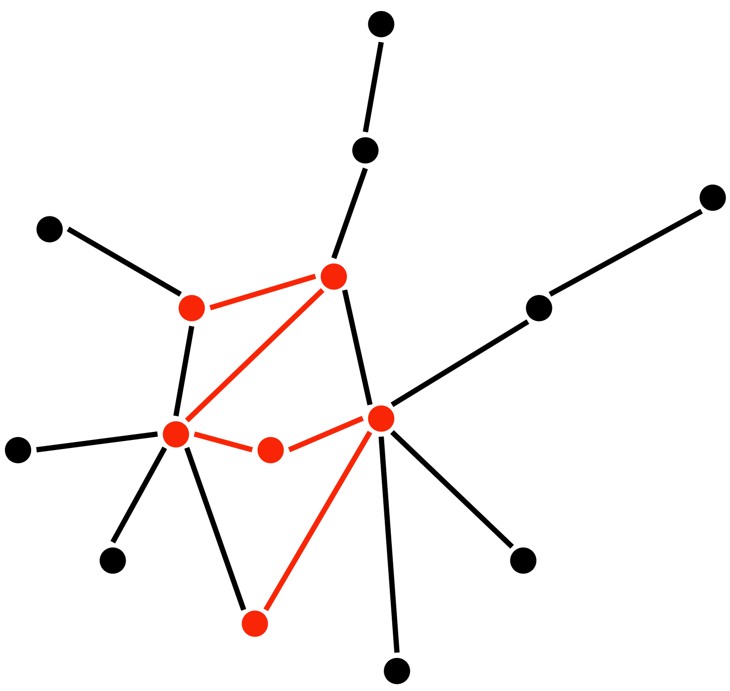
- graph diffusion



information modularity

- Cluster vertices while preserving information about the network structure

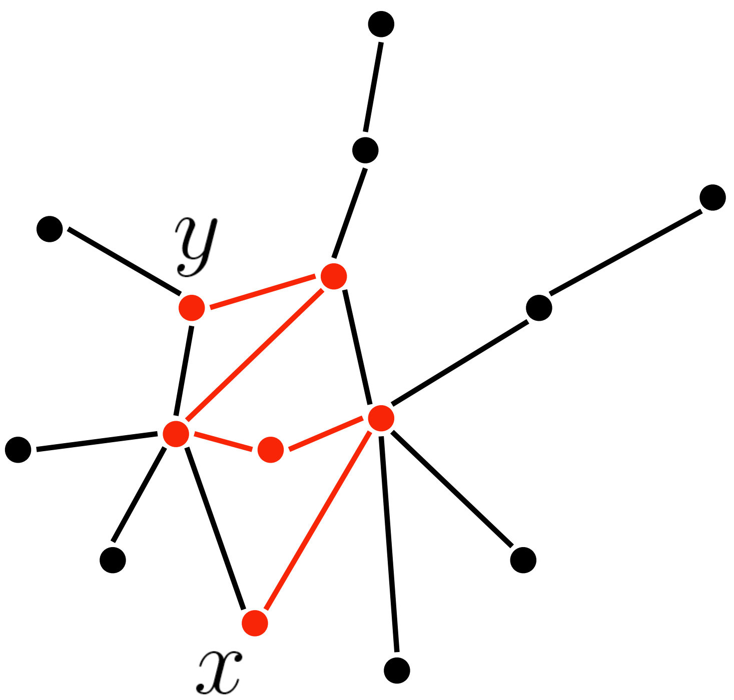
- graph diffusion



information modularity

- Cluster vertices while preserving information about the network structure

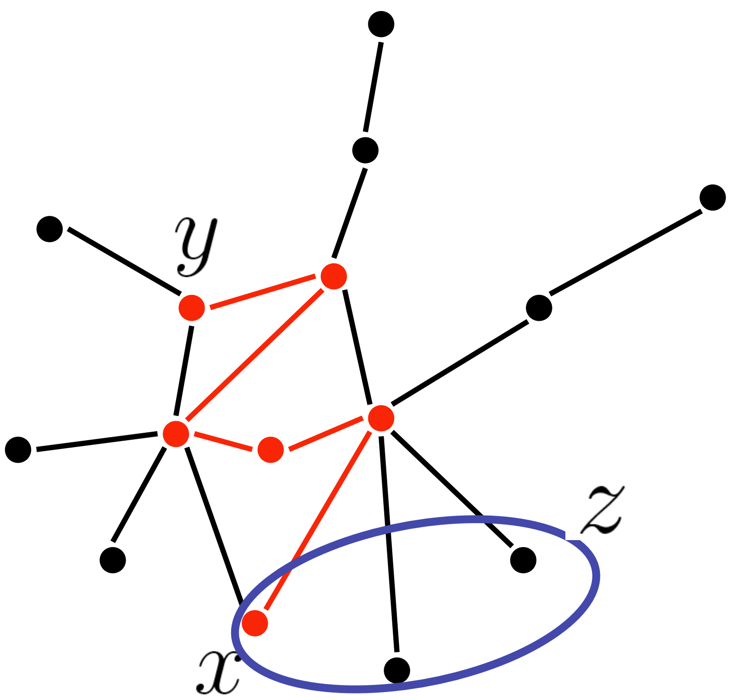
- graph diffusion



information modularity

- Cluster vertices while preserving information about the network structure

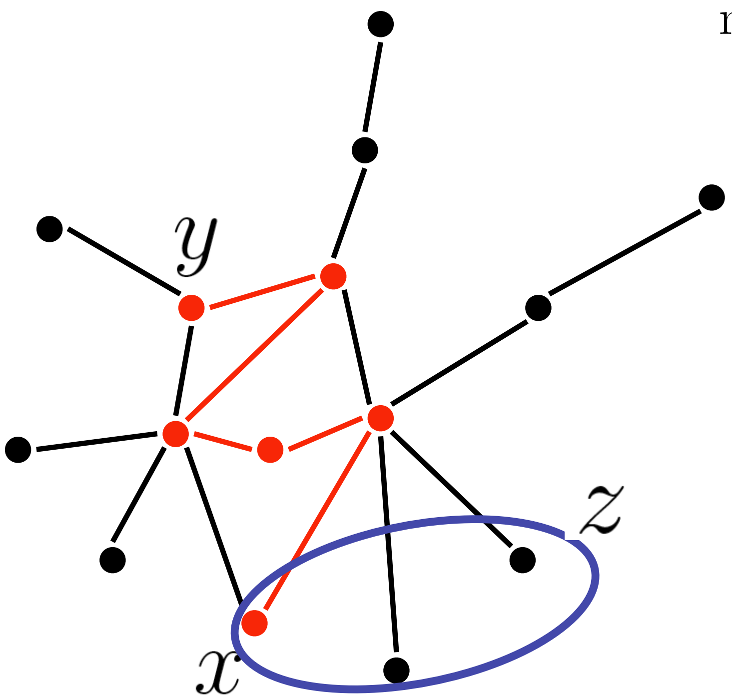
- graph diffusion



information modularity

- Cluster vertices while preserving information about the network structure

- graph diffusion



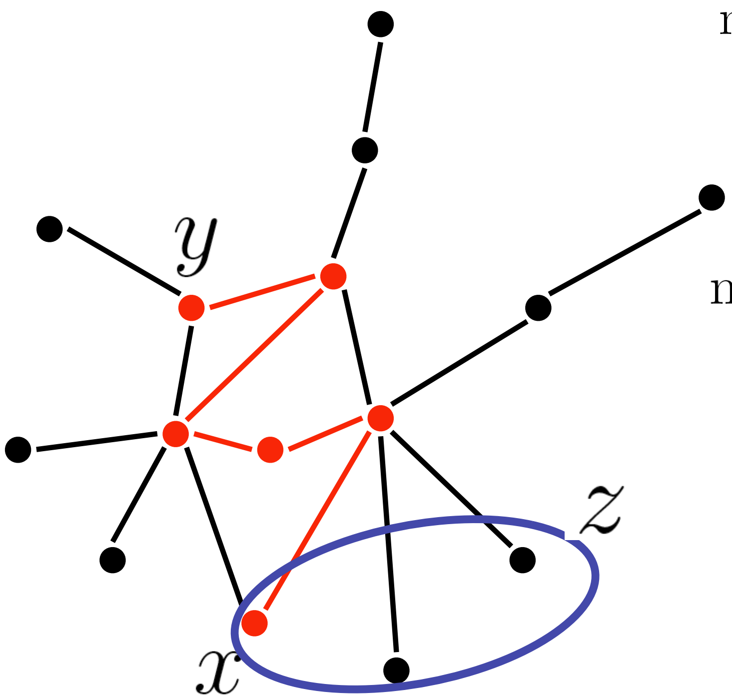
$$\text{maximize} \quad I[p(z,y)] = \sum_{z,y} p(z,y) \log \frac{p(z,y)}{p(z)p(y)}$$

(hard clustering)

information modularity

- Cluster vertices while preserving information about the network structure

- graph diffusion



$$\text{maximize} \quad I[p(z,y)] = \sum_{z,y} p(z,y) \log \frac{p(z,y)}{p(z)p(y)}$$

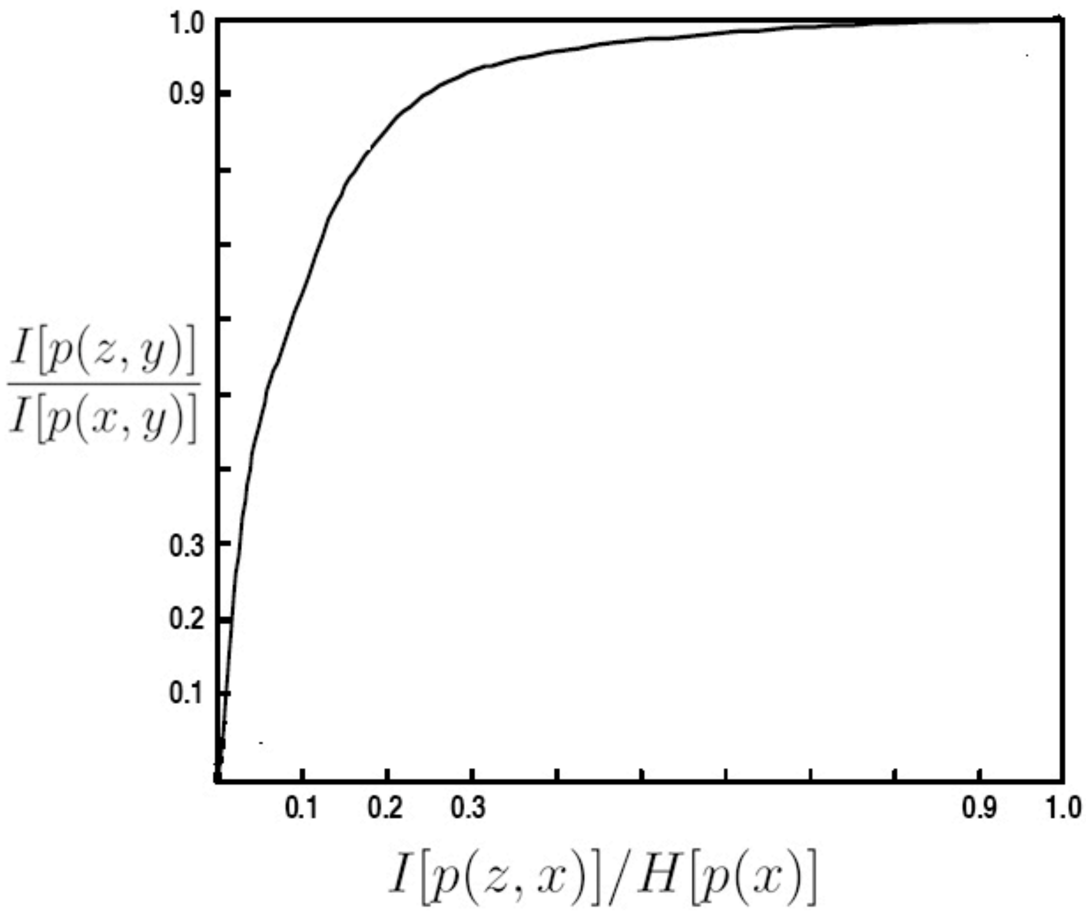
(hard clustering)

$$\text{minimize} \quad I[p(x,z)] - \beta I[p(z,y)]$$

(soft clustering)

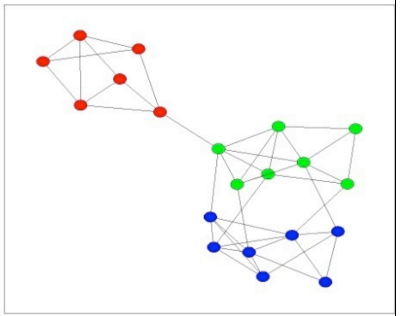
Defining Network Modularity

- Summarizability of network structure



method: diffusive distributions

Graph diffusion



- unbiased measure of connectivities

$$\partial_t \rho = \Delta \rho = \begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & -2 \end{pmatrix} \rho$$

- natural "length scale" of the graph

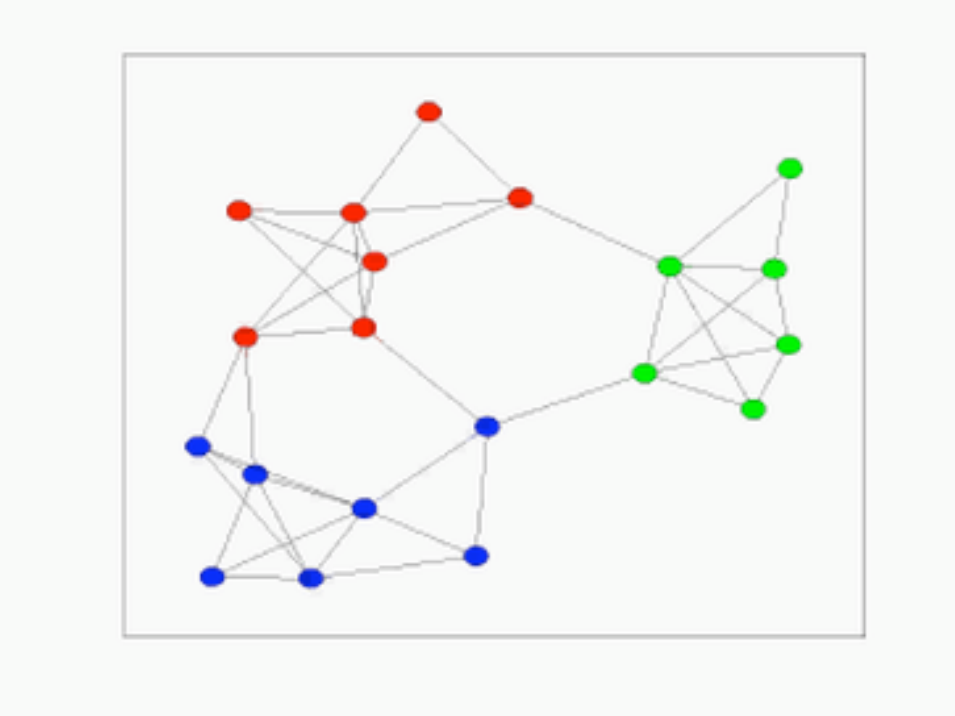
$$\Rightarrow \rho(y(t)|x(0)) \propto \mathrm{e}^{\Delta t} \qquad \rho \propto \sum_{\alpha=1}^{N_c} \mathrm{e}^{-\lambda_\alpha/\lambda_1} \mathbf{v}_\alpha \mathbf{v}_\alpha^T$$

- defines a joint distribution for a graph
- generalizes to joint dist. for networks
- sparse matrices=> approximation schemes

illustration: "hard" aglomerative case

- Graph provides joint distribution

$$\Rightarrow \rho(y(t)|x(0)) \propto \mathrm{e}^{\Delta t}$$



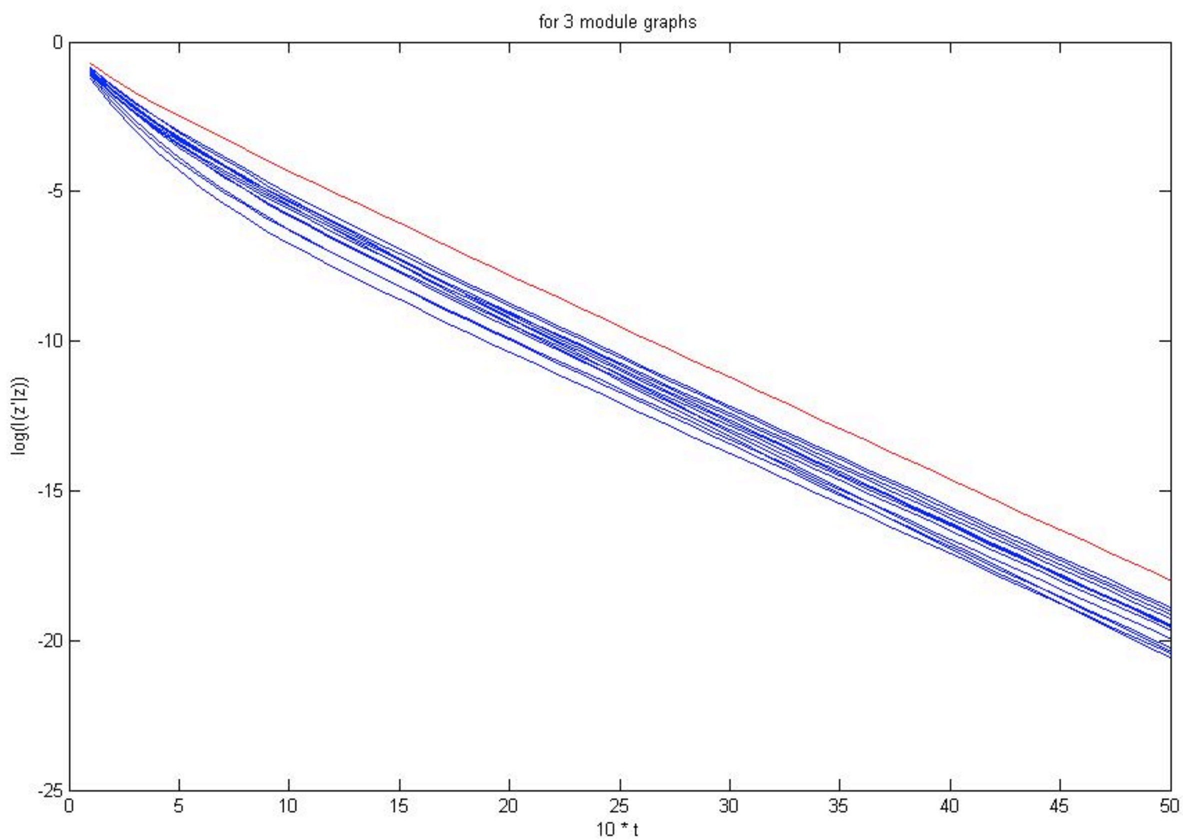
- "Bottlenecking"

$$\text{maximize } I(Y,Z) \ @ \ |Z|$$
$$p(y|z) = \textstyle\sum_x p(y|x)p(x|z)$$

- provides modules ...

$$\min_{ij} \quad I(\mathbf{y};\{z_i, z_j\})$$

dynamics: t-insensitive



SBM (synthetic) data, |z|=2; K=3

applications:

i. real data

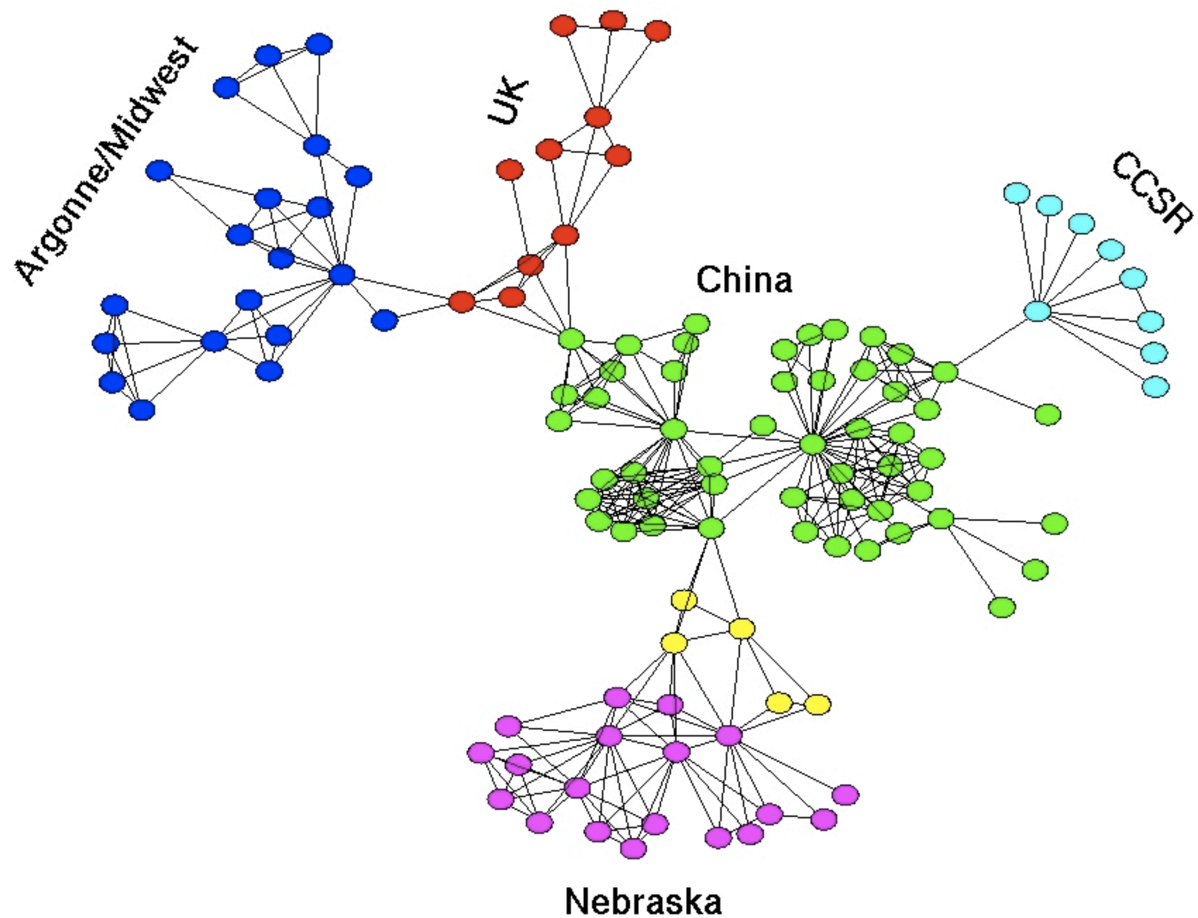
ii. fake data

central questions:

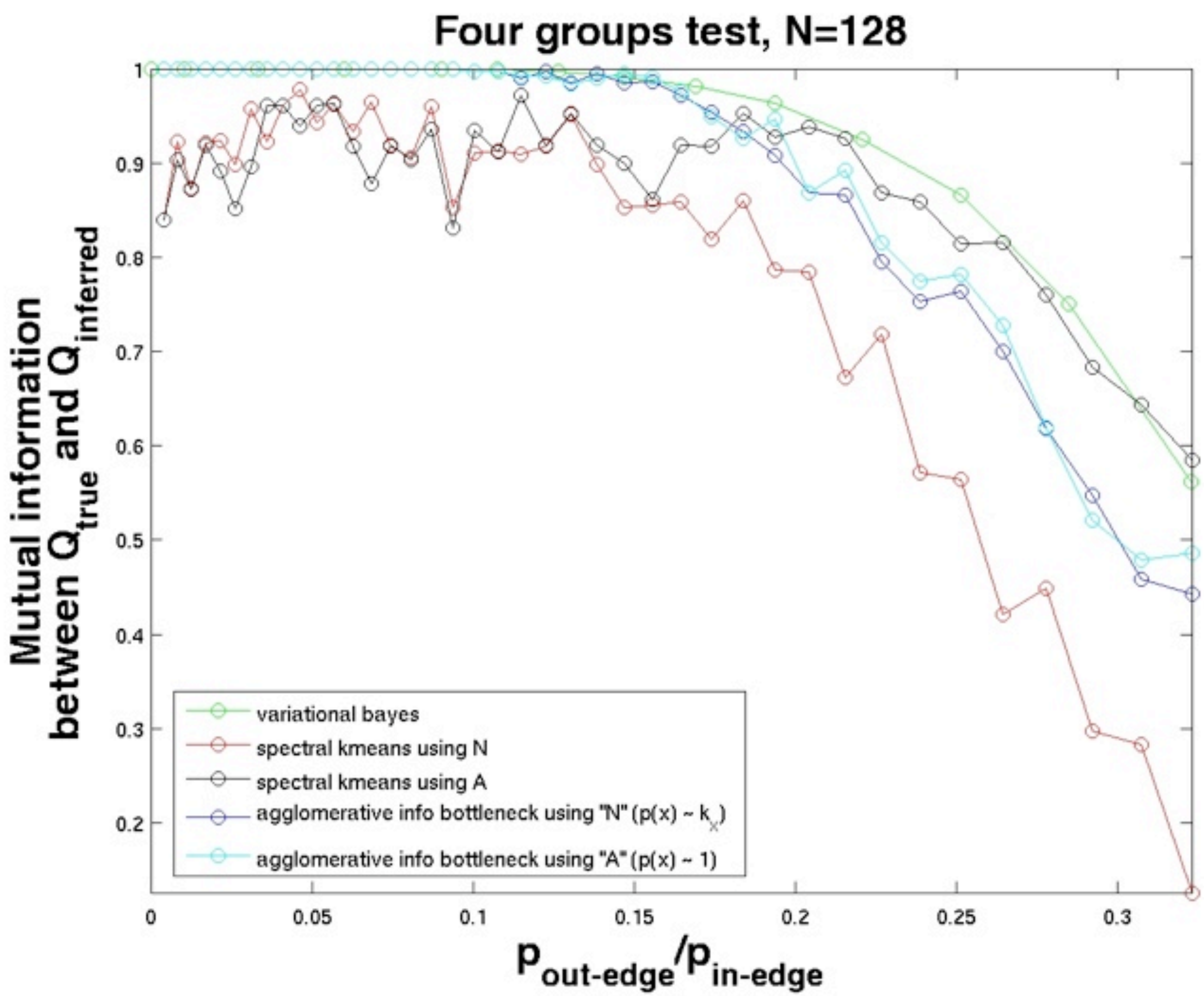
does it work?

why or why not?

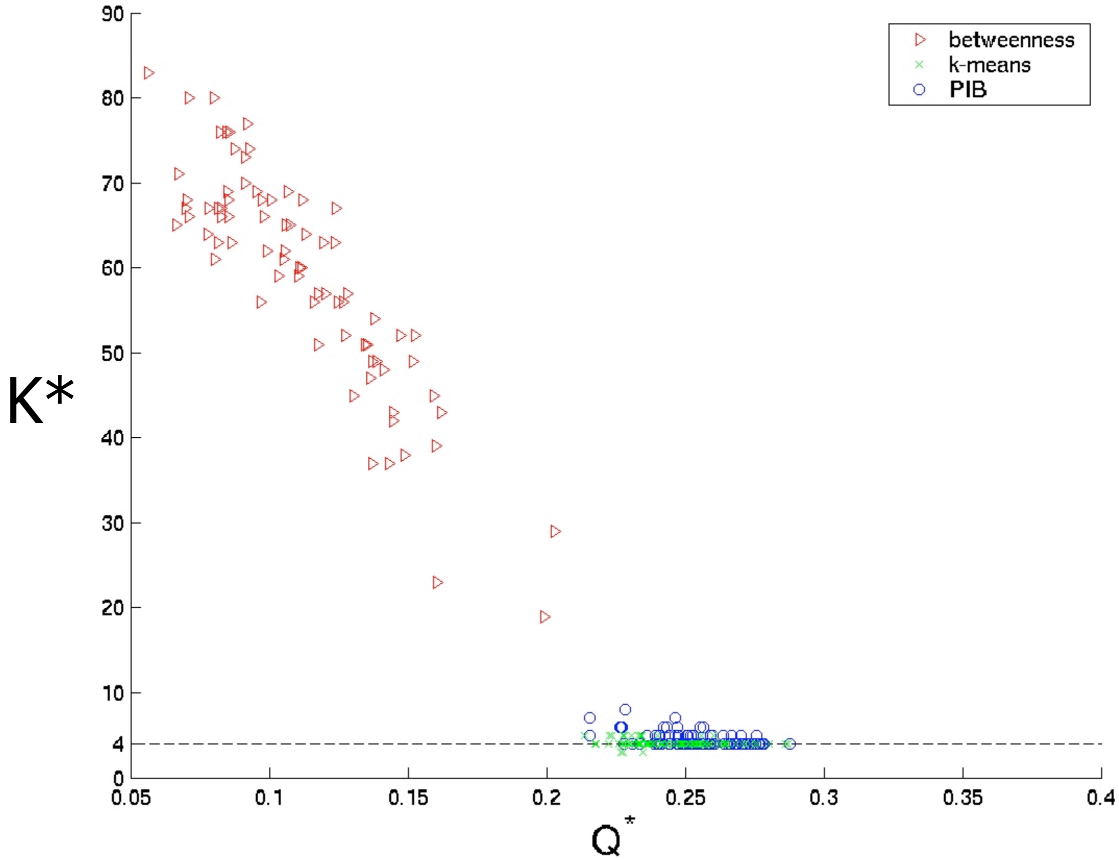
validation: social networks



validation w/generative model (SBM)



Results: K*=argmax(Q)



Why is K-means competitive with NIB?

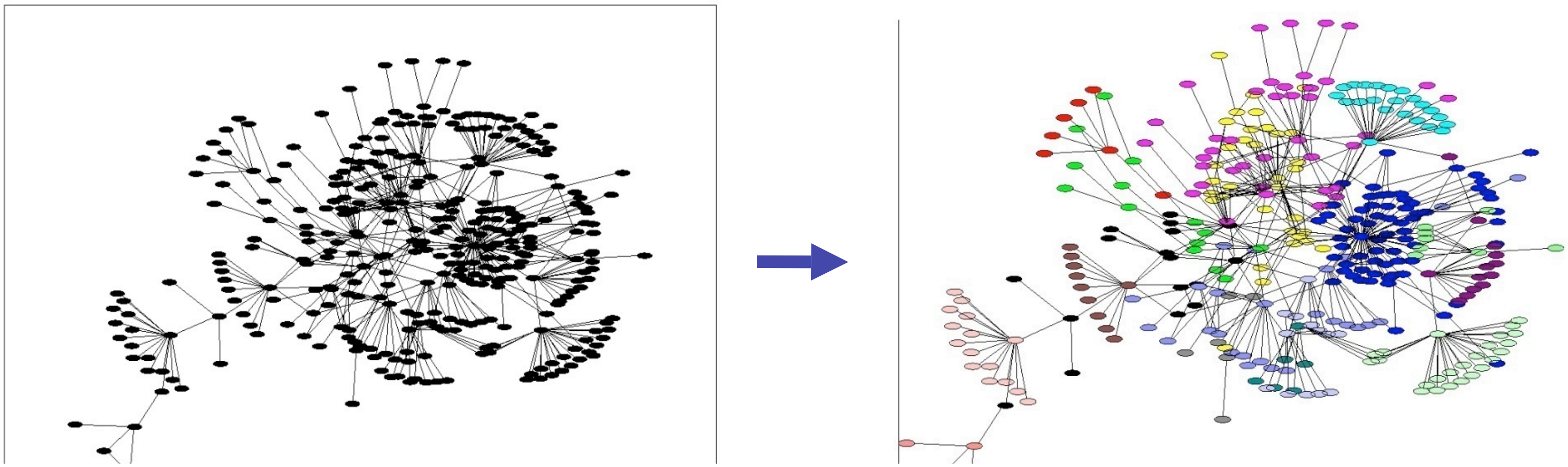
connection with spectral methods

"summarizable"=information-modular=edge modular?

agenda

- history
  - information-theoretic clustering
  - spectral partitioning
- mathematics
  - a rethink
  - a derivation
- computation
  - numerical experiments
  - pretty pictures
- philosophy

what just happened?



a rethink:

- summarizing/compression/encoding

i. does not require p(G)

ii. avoids trivial solution

iii. explains why spectral works/provides derivation of “N” ($\mathcal{N} \equiv -\frac{h^T L h}{1 - \langle h \rangle^2}$)

what just happened?

a rethink:

- summarizing/compression/encoding

i. does not require p(G)
ii. avoids trivial solution
iii. explains why spectral works/provides derivation of "N" ($\mathcal{N} \equiv -\frac{h^T L h}{1 - \langle h \rangle^2}$)
iv. generalizes to soft partitioning+"overlapping communities"
v. makes clear connection w/clustering
vi. exploits existing numerical approx. methods (for large graphs)
vii. gives order parameter for graph modularity

a rethink:

- summarizing/compression/encoding

i. does not require p(G)

ii. avoids trivial solution

iii. explains why spectral works/provides derivation of "N" ($\mathcal{N} \equiv -\frac{h^T L h}{1-\langle h \rangle^2}$)

iv. generalizes to soft partitioning +"overlapping communities"

v. makes clear connection w/clustering

vi. exploits existing numerical approx. methods (for large graphs)

vii. gives order parameter for gra
modularity w/o choosing scale

errors to set scale?
computational
approach

a rethink:

- summarizing/compression/encoding

i. does not require p(G)

ii. avoids trivial solution

iii. explains why spectral works/provides derivation of "N" ($\mathcal{N} \equiv -\frac{h^T L h}{1 - \langle h \rangle^2}$)

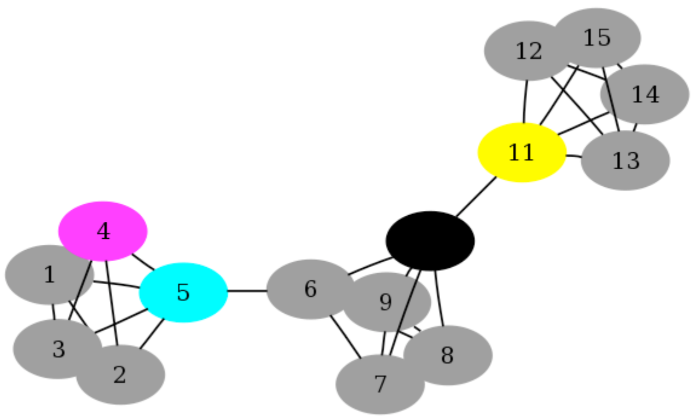
iv. generalizes to soft partitioning +"overlapping communities"

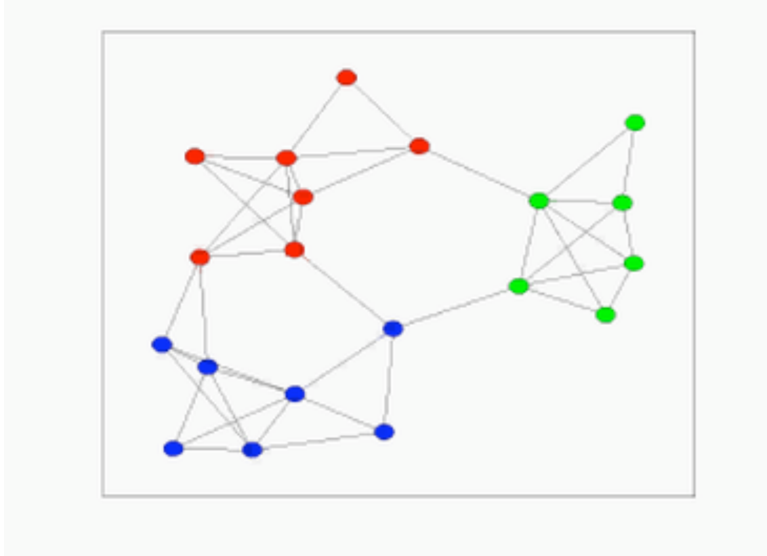
v. makes clear connection w/clustering

vi. exploits existing numerical approx. methods (for large graphs)

vii. gives order parameter for graph modularity w/o choosing scale or partition

punchlines:





assert p(G) ->

- inference of scale +modules,
- derivation/generalization of heuristics in literature

assert diffusive distortion ->

- optimal encoding,
- derivation/generalization of spectral heuristics,
- order parameter for graph modularity

inferring modules

- for more info
  - papers: arXiv: 0709.3512/PRL June 2007
  - jake's talking on july 4 in Helsinki
  - source code*: vbmod.sourceforge.net
- thanks
  - graduating student: jake hofman
  - funding: nih
  - invitation: Lek-Heng Lim

* try this at home. not "available by request". just available

for more info

papers: PRE+ arXiv:q-bio.QM/0411033

source code*: sourceforge.net

thanks

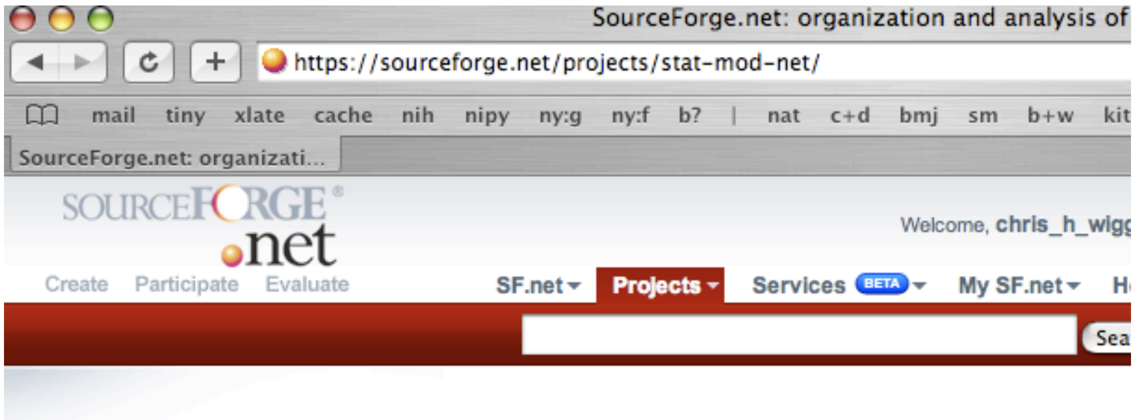
students: ziv, middendorf, raj

funding: nih/nsf/doe

invitation: Lek-Heng Lim

* try this at home. not “available by request”. just available

try this at home: sourceforge.net



social networks? technological networks? your network?