Gaussian Belief with Dynamic Data and in Dynamic Network

Workshop on

Techniques and Challenges from Statistical Physics

CRM@UAB, Barcelona

René Pfitzner and E.A., *Europhys Lett:* (2009), arXiv:0905:0266

Physicist think they know everything about Gaussians.

When a new result appears they (we) get very impressed.

Exactness of Belief Propagation in Gaussian models is important.

So this talk could just as well be classified under *Challenges from Computer Science and Signal Processing.*

Because exactness of Gaussian BP was discovered there.

Outline:

The exactness results [Weiss 2001; Johnson, Malioutov, Willsky 2006]

Some recent applications in CS and Signal Processing

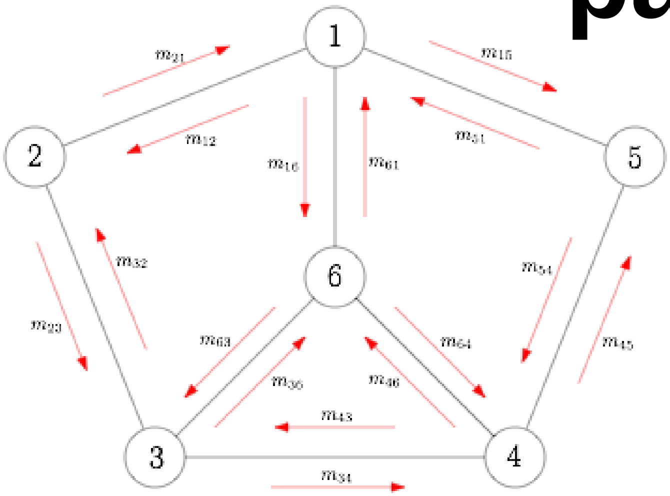
Our application (data aggregation)

Our numerical results

Our unexpected numerical results

KTH/CSC

Belief Propagation in pair-wise models



Model:

$$p(x) \propto \prod_{i \in V} \Psi_i(x_i) \prod_{\{i,j\} \in E} \Psi_{ij}(x_i, x_j)$$

Marginal probability:

$$p_i(x_i) = \frac{1}{Z_i} \Psi_i(x_i) \prod_{k \in N(i)} m_{k \to i}(x_i)$$

Message update rule:

$$m_{i \to j} = \int \Psi_{ij} \Psi_i \prod_{k \in N(i) \setminus j} m_{k \to i} dx_i$$

Gaussian BP

$$P(x) \propto \exp\left(-\tfrac{1}{2} x_i J_{ij} x_j + h_i x_i\right)$$ "Information matrix" form

$$m_{i\to j}(x_j) \propto \exp\left(-\tfrac{1}{2} L_{i\to j} x_j^2 + \nu_{i\to j} x_j\right)$$ Gaussian messages

$$L_{i\to j} = -J_{ij}^2 \frac{1}{J_{ii} + K_{i\to j}} \qquad K_{i\to j} = \sum_{k\in\partial i\setminus j} L_{k\to i}$$ GaBP variance update

$$\nu_{i\to j} = -J_{ij} \frac{h_i + \mu_{i\to j}}{J_{ii} + K_{i\to j}} \qquad \mu_{i\to j} = \sum_{k\in\partial i\setminus j} \nu_{k\to i}$$ GaBP mean update

$$b_i(x_i) \propto \exp\left(-\tfrac{1}{2} K_i x_i^2 + \mu_i x_i\right)$$ Gaussian marginals

$$K_i = J_{ii} + \sum_{k\in\partial i} L_{k\to i} \qquad \mu_i = h_i + \sum_{k\in\partial i} \nu_{k\to i}$$ GaBP output

So what? Computing the marginals is matrix inversion

$$P(\chi_i) \propto \exp\left(-\tfrac{1}{2}(x_i-(J^{-1}h)_i)^2/J_{ii}^{-1}\right)$$

Correlations = inverse of J

The walk-sum interpretation

Scale to $J_{ii}=1 \quad J_{ij}=1_{ij}-R_{ij} \qquad J_{ij}^{-1}=1_{ij}+R_{ij}+\sum_k R_{ik}R_{kj}+\ldots$

$$m_i=(J^{-1}h)_i=\sum_j h_j(\text{ Walks from j to i, picking up factors of R })$$

These walks can be done on a computational tree.

The algebra of partial sums of the walks on trees is exactly Gaussian BP update.

Hence Gaussian BP ***converges*** and ***is exact for the means*** whenever the spectral radius of the matrix |R| (element-wise absolute values) is less than 1.

Johnson, Malioutov, Willsky, NIPS **18** & *J. Machine Learning Res.* **7**: 2031-2064 (2006)

Some applications

GaBP is a distributed iterative linear equation solver

...has been reported to work better than classical iterative solution methods

Shental et al, Gaussian BP Solver for Systems of Linear Equations, ISIT 2008

ρ(|R|)<1 not a strict condition, can be fixed by preconditioner

...ρ(R)<1 is of course a strict condition. An alternate algorithm (incomplete computation of matrix inverse by GaBP) has also shown good behaviour

Johnson et al, Fixing Convergence of Gaussian Belief Propagation, arXiv:0901:4192

KTH/CSC

Some more recent applications

Fault identification + lattice decoders + other stuff

...using Gaussian BP on Gaussian mixture models (I will not try to explain)

Bickson, Avissar, Dolev, Boyd, Ihler, Baron, arXiv:0908.2005

Avissar, Bickson, Dolev, Ihler, 47th Allerton Conference, Sept. 2009

Distributed Netwon method

...recent work by Boyd, Dolev and collaborators. A new approach to network optimization, where the Newton step can be done using GaBP

...which means a way for nodes to agree on values (x,y,z,...) such that conditions F(x,y,z,...) = 0, G(x,y,z,..) = 0... are true, just by passing messages

Our application: data aggregation

In network management, sensor networks etc

This functionality can also be achieved by gossiping (physically a diffusion processes), or by maintaining a spanning tree, where data aggregated from the leaves to the root

The spanning tree has issues under churn, when nodes come and go

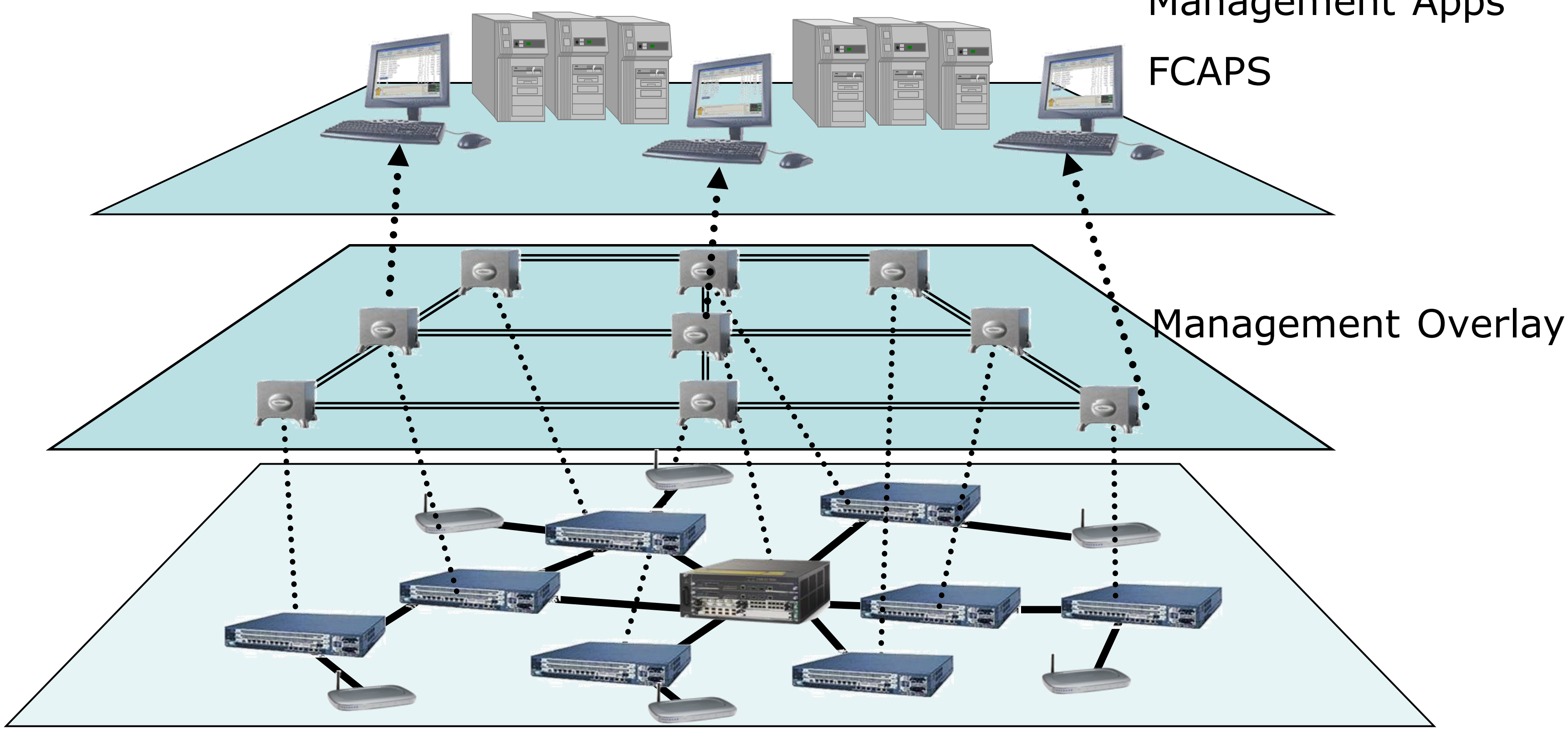
Gossiping has problems with convergence speed in large networks, and with leakage, if nodes leave and take their data with them

Consensus Propagation, a special case of Gaussian BP, may be a trade-off

Moallemi and Van Roy, "Consensus Propagation", IEEE Transactions on Information Theory, Vol. 52, No. 11, November 2006

Distributed management

Monitoring and data aggregation



Courtesy R Stadler

Consensus Propagation

$$P(x) \propto \exp\left(-\tfrac{1}{2}(x_i - y_i)^2 - \tfrac{1}{2}\beta Q_{ij}(x_i - x_j)^2\right)$$

This is a walk-summable model for any finite β, and Gaussian BP will hence converge, and give exact means of marginals

The average of the x's is centered around the average of y's:

$$P\left(\frac{1}{N}\sum x_i\right) \propto \exp\left(-\frac{1}{2N}\left(\Sigma x_i - \Sigma y_i\right)^2\right)$$

When β goes to infinity all the x's are the same. Hence the median of all these local variables then hold the global average of the y's. Hence the name Consensus Propagation.

Dynamism

Dynamic data = the local y's change in time

...agreeing on a moving target

Competition between the dynamics of the data and the BP dynamics

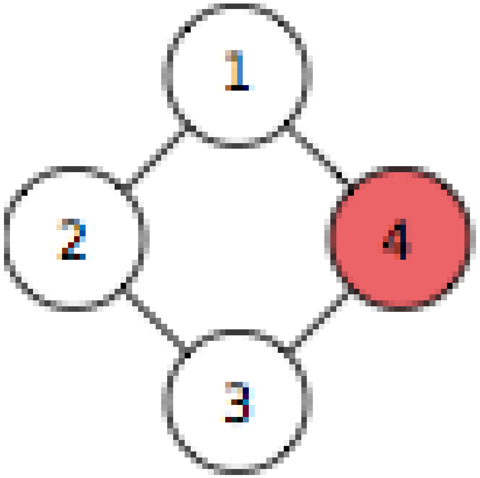
Dynamic network = the couplings Q change in time

...somewhat related to work using Gaussian BP as a linear equation solver with different preconditioners at different time steps, and where links may fail

Chandrasekaran, Johnson and Willsky, "Estimation in Gaussian Graphical Models…", IEEE Transactions on Signal Processing **56**: 1916-1930 (2008)

KTH/CSC

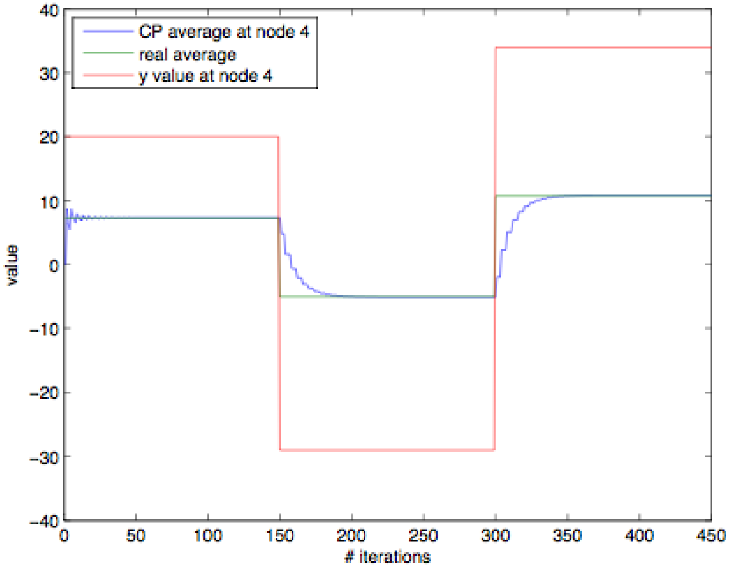
CP on small graph with dynamic data



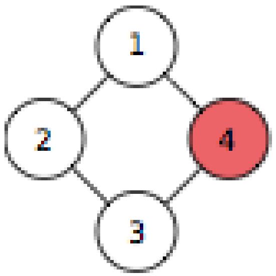
$$\mu_{ij}^{(t+1)} = \frac{y_i + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)} \mu_{ki}^{(t)}}{1 + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)}}$$

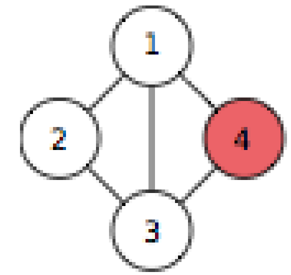
$$K_{ij}^{(t+1)} = \frac{1 + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)}}{1 + \frac{1}{\beta Q_{ij}} (1 + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)})}$$

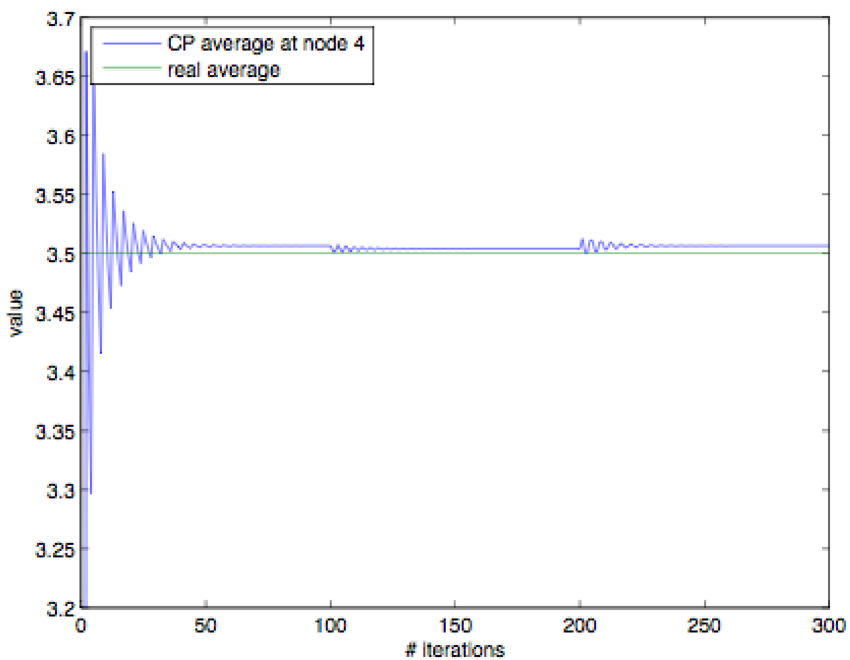
$$\overline{y} \approx \frac{y_i + \sum_{k\in N(i)} K_{ki}^{(t)} \mu_{ki}^{(t)}}{1 + \sum_{k\in N(i)} K_{ki}^{(t)}}$$

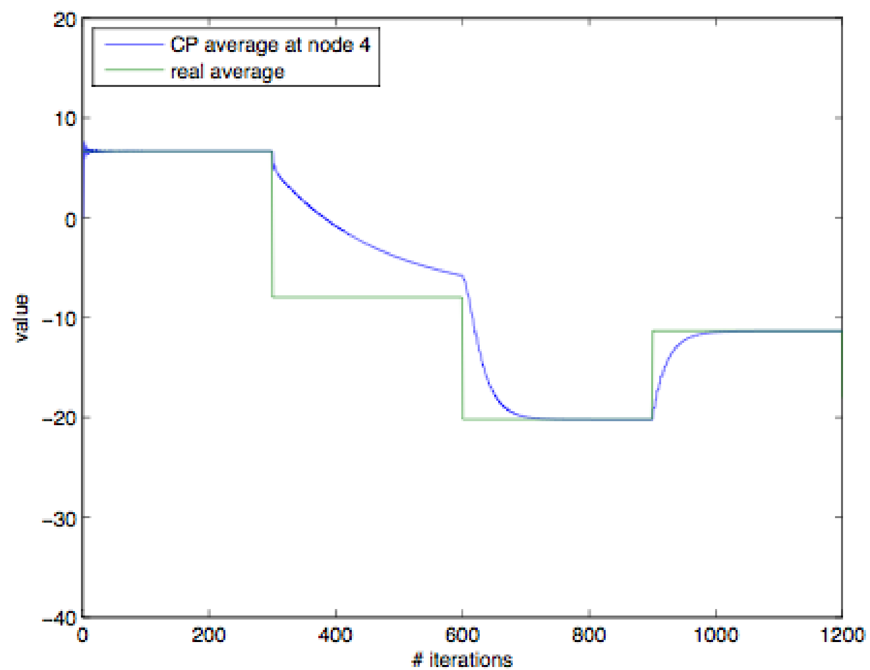


It is better to start from all-zero messages — not obvious (two runs)

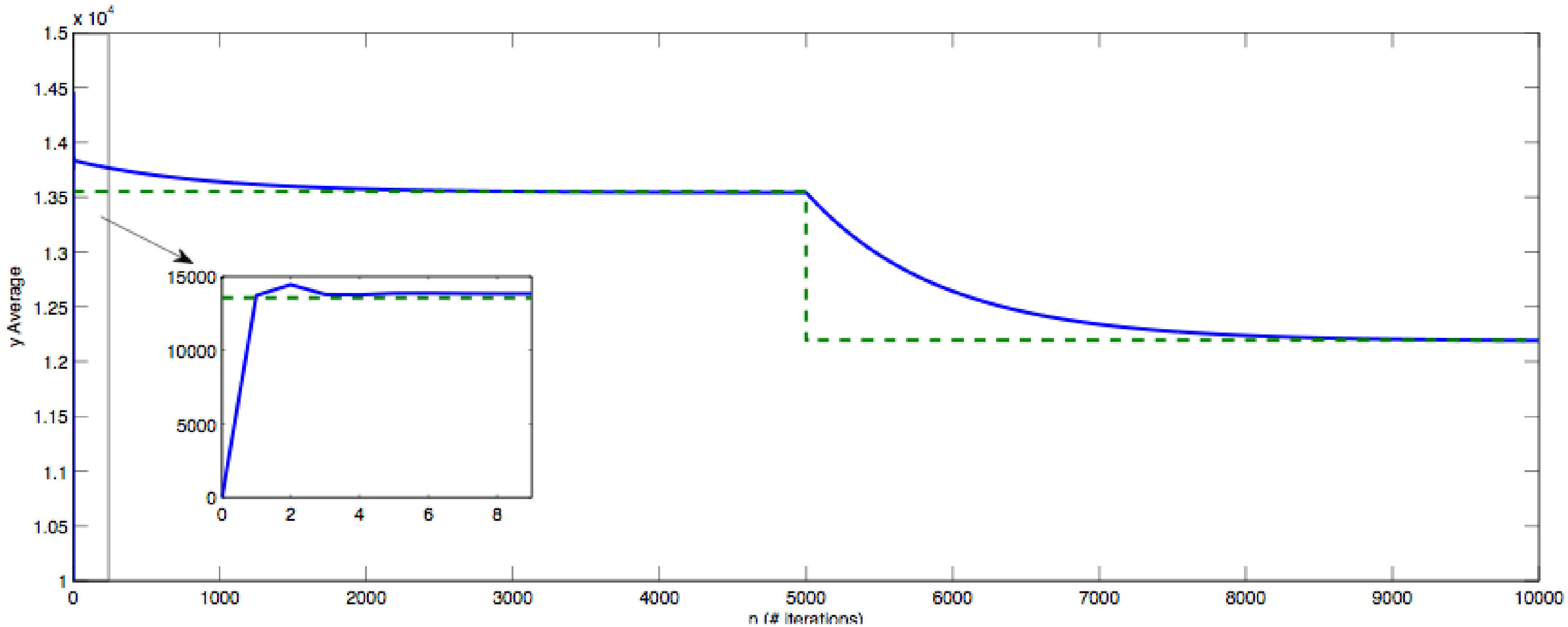








Also if connectivity is a large Erdös-Renyi graph convergence is similar



Convergence

It should perhaps not be a surprise that dynamic network improves convergence of Consensus Propagation

...consensus P computes an average, and mixing improves averaging

It was a suprise that it can improve so much

The limiting factor is in any case dynamic data

...so most of the rest of the talk concerns dynamic data only. The CP update rules are iterated map equations, and with only dynamic data they are eventually linear equation (of the μ messages)

KTH/CSC

Linearization

$$\begin{pmatrix} \mu_{ij}^{(n+1)} \\ \cdots \\ K_{ij}^{(n+1)} \\ \cdots \end{pmatrix} = \mathbb{R} \begin{pmatrix} \mu_{ij}^{(n)} \\ \cdots \\ K_{ij}^{(n)} \\ \cdots \end{pmatrix}$$

$$\mu_{ij}^{(t+1)} = \frac{y_i + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)} \mu_{ki}^{(t)}}{1 + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)}}$$

$$K_{ij}^{(t+1)} = \frac{1 + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)}}{1 + \frac{1}{\beta Q_{ij}}\left(1 + \sum_{k\in N(i)\setminus j} K_{ki}^{(t)}\right)}$$

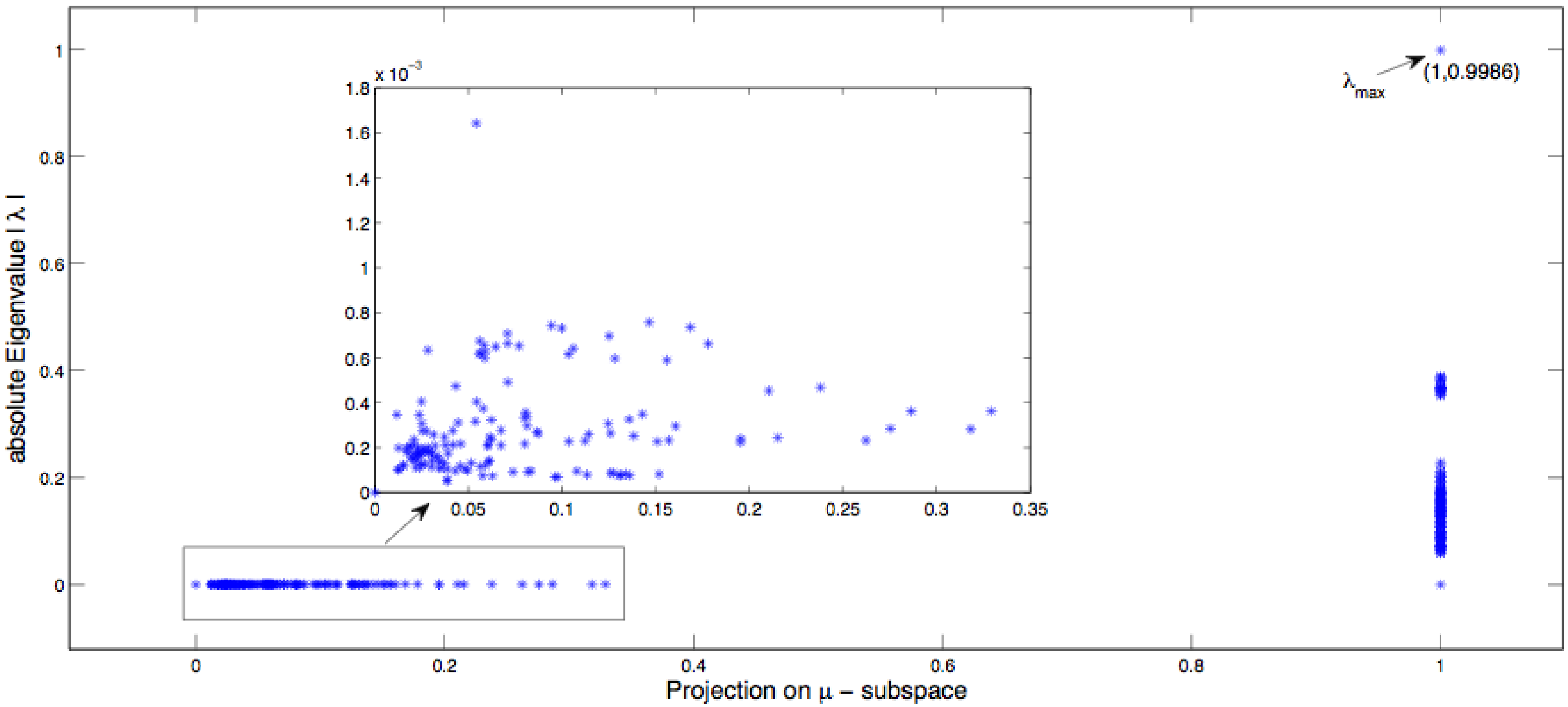
$$\begin{pmatrix} \mu_{ij}^{(n+1)} \\ \cdots \\ K_{ij}^{(n+1)} \\ \cdots \end{pmatrix} = \begin{pmatrix} \mu_{ij}^{*} + \Delta\mu_{ij}^{(n+1)} \\ \cdots \\ K_{ij}^{*} + \Delta K_{ij}^{(n+1)} \\ \cdots \end{pmatrix} \approx \begin{pmatrix} \mu_{ij}^{*} \\ \cdots \\ K_{ij}^{*} \\ \cdots \end{pmatrix} + \mathbb{R}' \begin{pmatrix} \Delta\mu_{ij}^{(n)} \\ \cdots \\ \Delta K_{ij}^{(n)} \\ \cdots \end{pmatrix}$$

K messages do not depend on μ messages

$$R' = \begin{pmatrix} A & C \\ 0 & B \end{pmatrix}$$

KTH/CSC

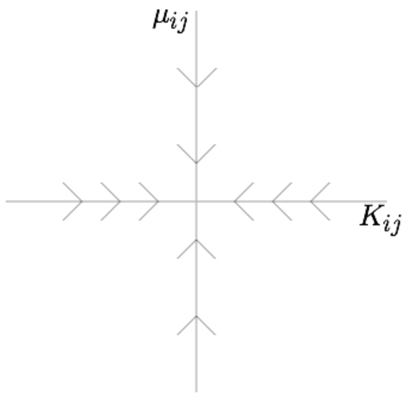
Largest eigenvalues of R' lie in the μ directions



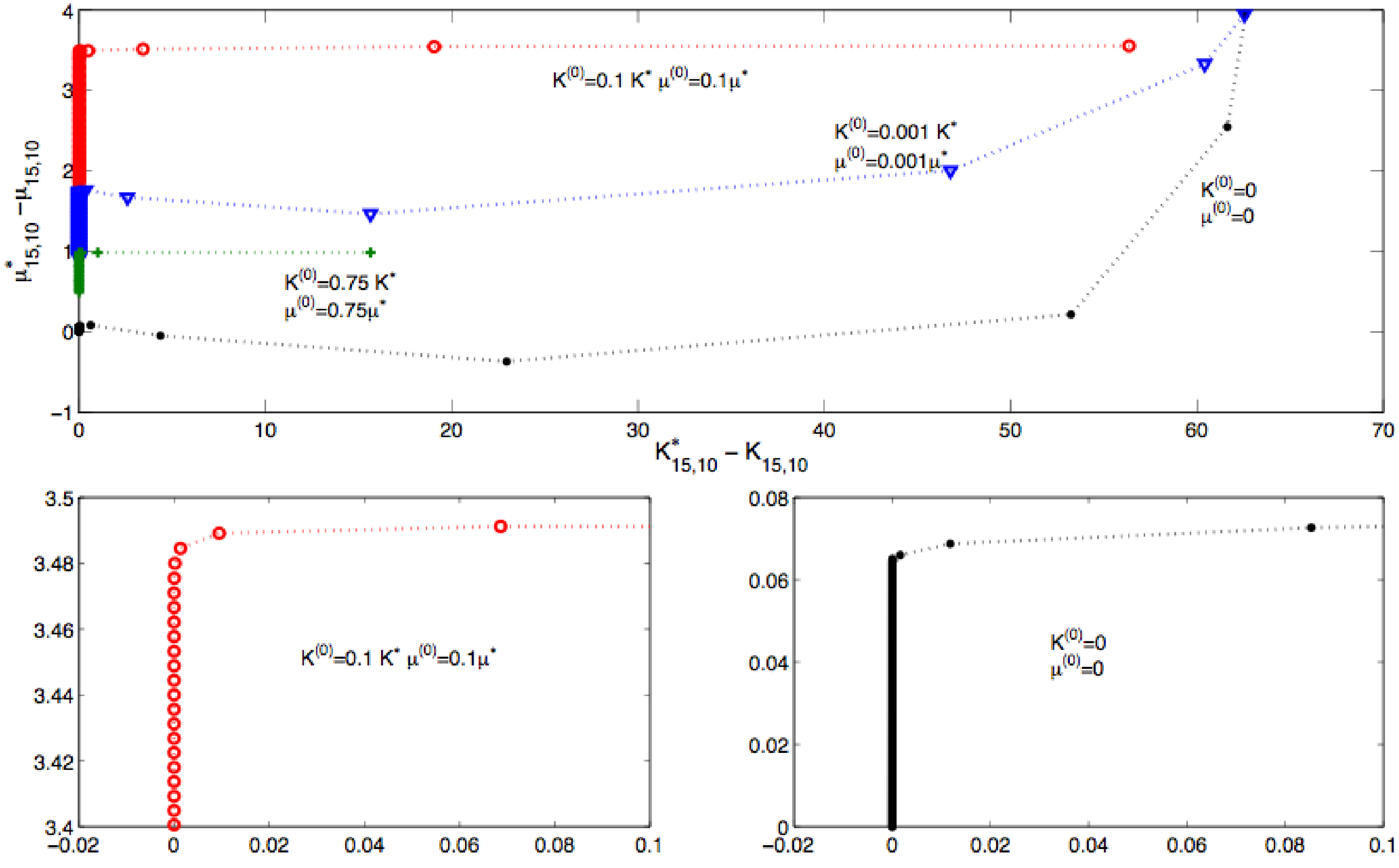
The largest eigenvalues of submatrix A (μ) are much larger than the largest eigenvalue of submatrix B (K direction)

Erdös-Renyi graphs with different N and c – fast and slow directions, extending to fast and slow manifolds

N	c	$\lambda_{max}(A)$	$\lambda_{max}(B)$
20	18	0.99949152	0.00054415
30	14	0.99924356	0.00083034
40	10	0.99895415	0.00119833
50	8	0.99851962	0.00186674



Phase-space plots of Consensus P dynamics



KTH/CSC

Largest eigenvalue is self-averaging in E-R graphs and independent of graph size N

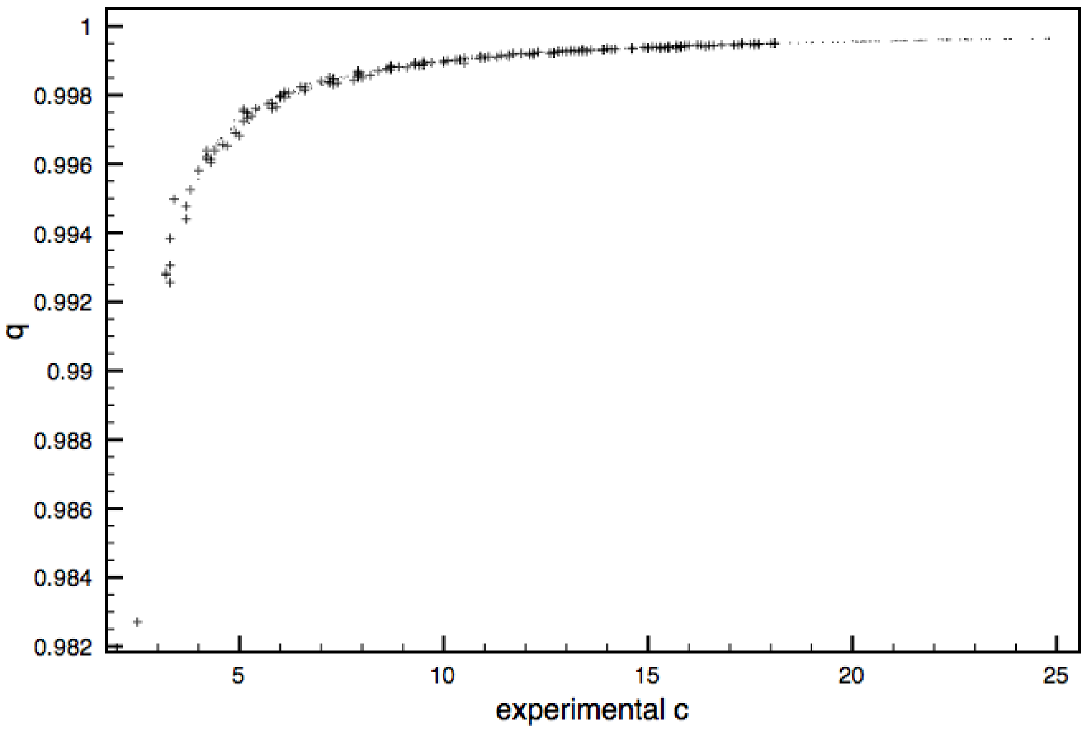
Erdös-Renyi graphs of c around 8 (random realizations) of different size from 20 to 20 000.

For small N computed eigenvalue of linearized GaBP; for large N estimated convergence ratio from iterating GaBP.

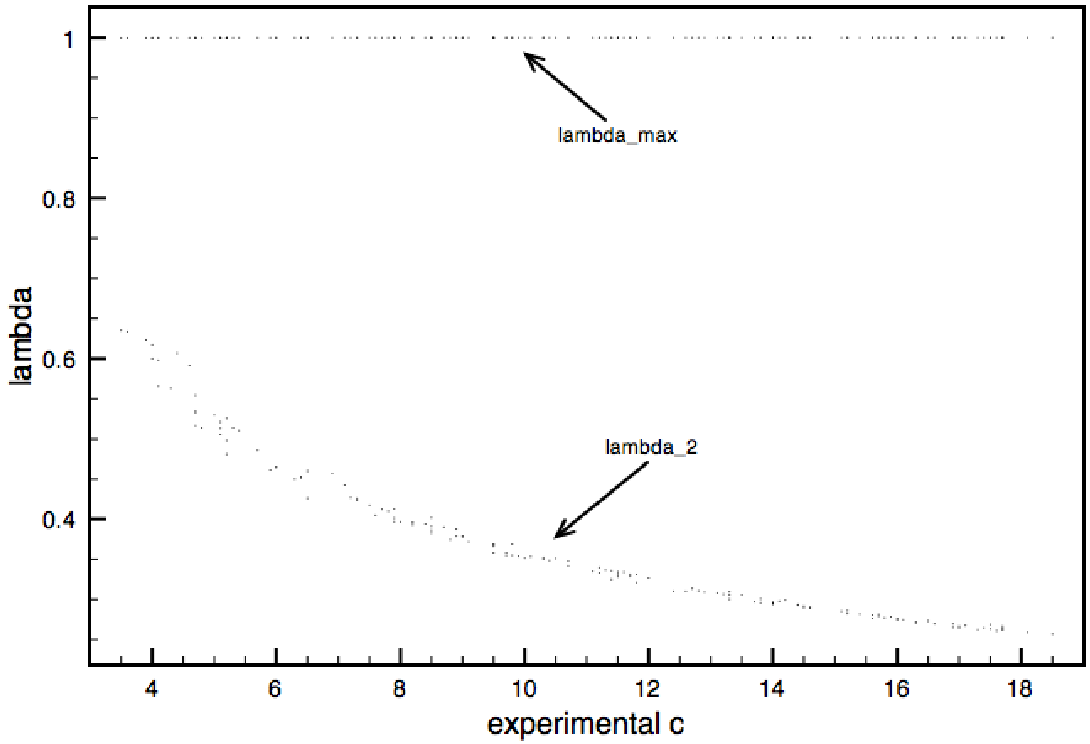
N	p	$[c_{exp}]_{100}$	$\sigma[c_{exp}]_{100}$	q_{100}
20	0.4	7.6	0.6	0.9985
40	0.2	7.8	0.5	0.9985
80	0.1	7.9	0.4	0.9985
160	0.05	8.0	0.3	0.9985

N	p	q
5000	0.0016	0.99850
10000	0.0008	0.99851
20000	0.0004	0.99850

Independence of N for different c. Convergence slows with increasing values of c.

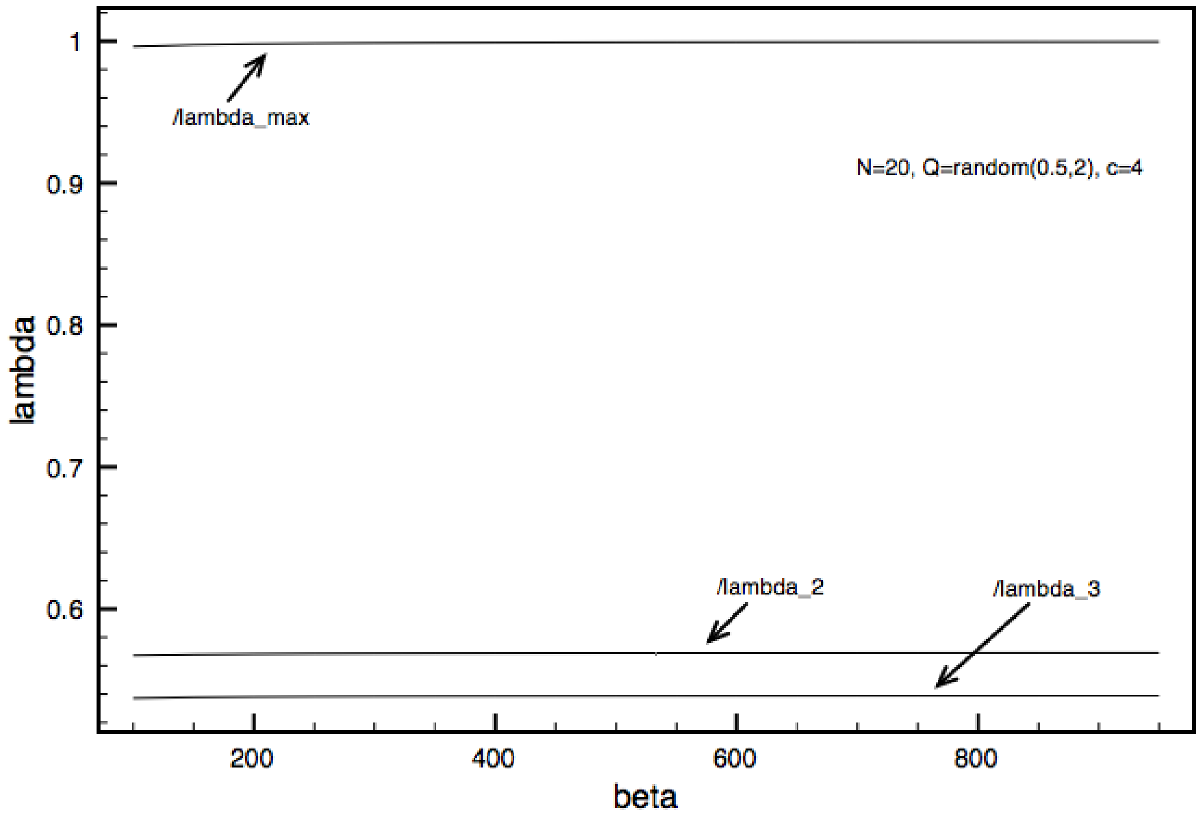


There is a spectral gap, and it increases with c

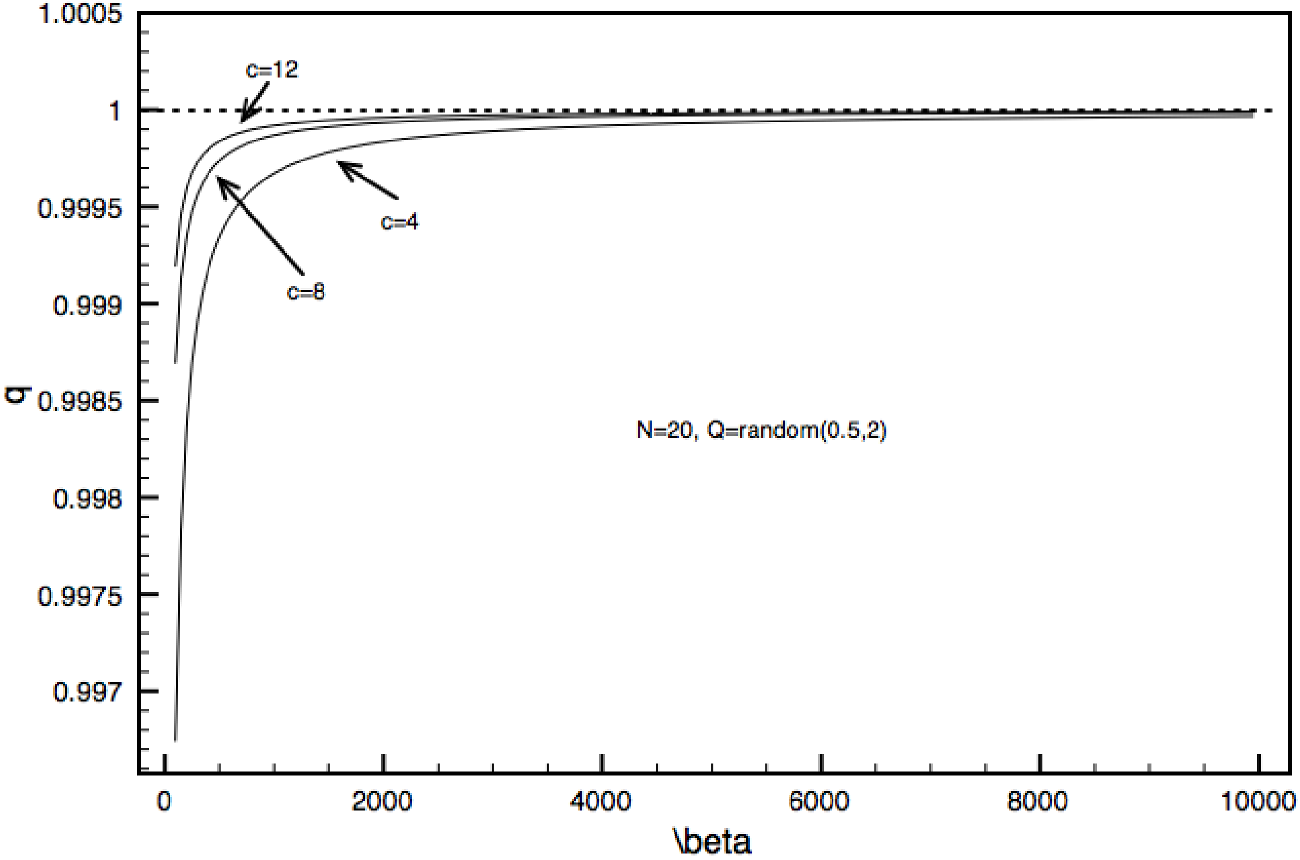


KTH/CSC

There is a spectral gap, and it remains as β tends to infinity

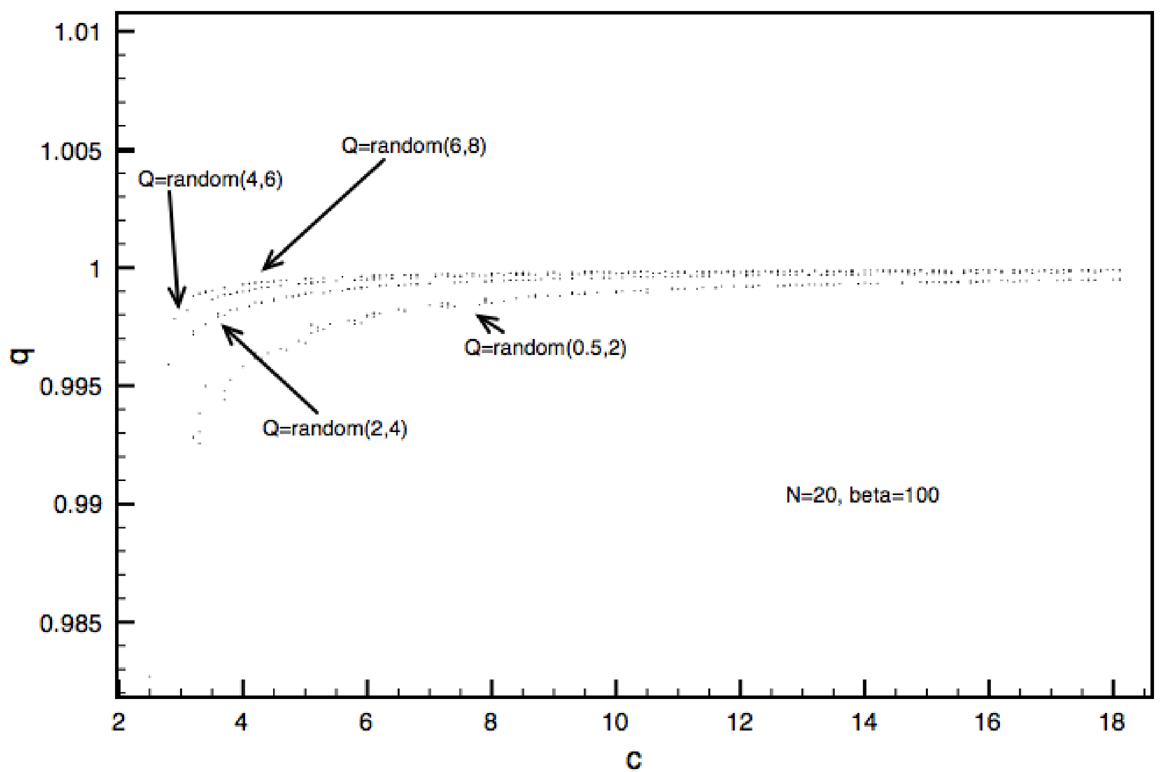


But the leading eigenvalue itself of course tends to one

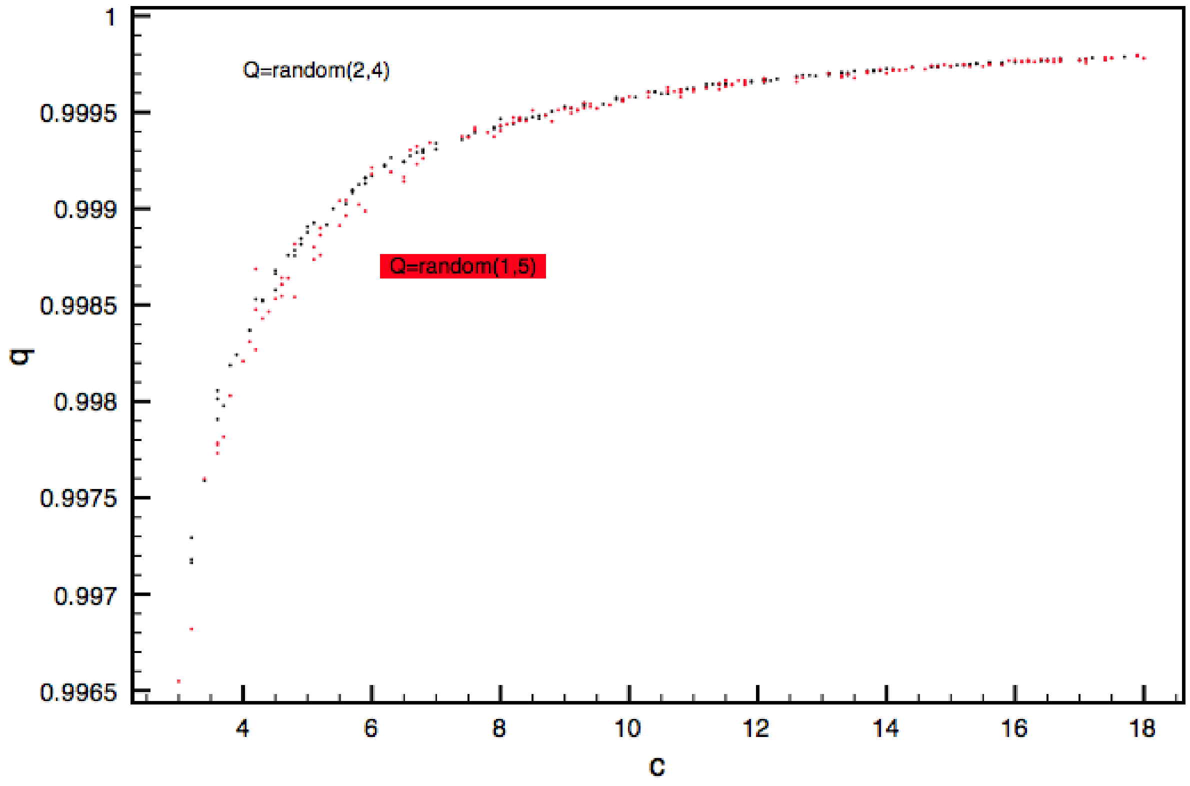


KTH/CSC

And eigenvalue depends on the ensemble of couplings Q



But perhaps only (or mainly) through <Q>



Challenges

Delineate where and when Consenus Propagation outperforms gossiping and tree-based aggregation.

Computing the eigenvalues of linearized Consensus Propagation in Erdös-Renyi graphs is a random matrix problem. Can they be computed with random matrix techniques?

Can one exploit the spectral gap to improve convergence and push closer to the infinite β limit?

Can one compute other things than averages? The Gaussian is the heat kernel. Are there other kernels out there with good properties for Belief Propagation?

KTH/CSC

Thanks to

HELSINKI UNIVERSITY OF TECHNOLOGY

Rene Pfitzner

Erasmus Mundus

Vetenskapsrådet

KTH/CSC

ACADEMY OF FINLAND

Rolf Stadler

Mads Dam

Supriya Krishnamurthy