Talk Notes (First Half)

Thomas Orton

September 2018

1 Navigating The References

This section is intended to quickly direct people to papers which go into more detail on topics that were presented in the talk today.

1. For more on belief propagation and its correspondence to free energy minimization (and other ideas in statistical physics), see [4].
2. For more on the stochastic block model, see [2]. [3] gives a more informal, reader friendly account of the SBM, and also makes some of the stability analysis of factored fixed points cleaner.
3. See [1] for information theoretic bounds on the SBM.
4. In our talk, we briefly discussed using BP on the Stochastic Block Model to get hardness conjectures for this problem. BP can be used in a similar way to get hardness conjectures for k-SAT. See [7] (more formal) and [6] (more informal) for details on how to apply BP to k-SAT for predicting computational hardness bounds.

2 Introduction

Last week, we saw how certain computational problems like 3SAT exhibit a thresholding behavior, similar to a phase transition in a physical system. This week, we'll continue to look at this phenomenon by exploring a number of heuristic methods which have been used to make hardness conjectures, and also have thresholding properties. In particular, we'll start by looking at Belief propagation for approximate inference on sparse graphs, and the corresponding physical energy minimization analog of belief propagation for physical systems. We'll then look at the stochastic block model, and talk about using BP to make inferences on random graphs. We'll see some heuristic techniques for determining when BP succeeds in inference, and when it fails, as well as some numerical simulation results of BP. Lastly, we'll talk about where this all fits into what is currently known about efficient algorithms, and information theoretic barriers, of the stochastic block model.

After this, we'll look at Approximate Message Passing, which can be seen as a variant of BP for dense random graphs, and its ability to infer parameters of a random model. After calculating the regime where AMP works, we will use a completely different heuristic argument (replica symmetry breaking) to show that the threshold we calculated coincides with a heuristic prediction of when inference should be possible.

3 Factor Graphs, and Belief Propagation

3.1 Motivation

Suppose someone gives you a probabilistic model on $\vec{x} = (x_1, ..., x_n) \in \chi^N$ which can be decomposed in a special way, say

$$P(\vec{x}) \propto \prod_{a \in F} f_a(\vec{x}) \tag{1}$$

Where each f_a only depends on the variables V_a. Recall from last week that we can express constraint satisfaction problems in these kinds of models, where each f_a is associated with a particular constraint. For example, given a 3SAT formula $\phi = \wedge_a c_a$, we can let $f_a(\vec{x}) = 1$ iff $c_a(\vec{x})$ is satisfied. Then each f_a only depends on 3 variables, and $P(\vec{x})$ only has support on satisfying assignments of ϕ.

A central problem in computer science is trying to find satisfying assignments to constraint satisfaction problems, i.e. finding values $\vec{x}$ in the support of $P(\vec{x})$. Suppose that we knew the value $P(x_1 = 1) > 0$. Then we would know that there exists some satisfying assignment where $x_1 = 1$. Using this knowledge, we could recursively try to find $\vec{x}$'s in the support of $P(1, x_2, ..., x_n)$, and iteratively come up with a satisfying assignment to our constraint satisfaction problem. A natural question is therefore the following: suppose we're given a factor graph where each $f_a(\vec{x})$ only depends on $\mathcal{O}(1)$ variables. When can we try to efficiently compute the marginals

$$P(x_i) := \sum_{\vec{x} - x_i} P(\vec{x}) \tag{2}$$

for each i?

A well known efficient algorithm for this problem exists when the factor graph is a tree, and the name of the algorithm depends on what field you study. Even though belief propagation is only guaranteed to work exactly for trees, we might hope that if our factor graph is "tree like", then BP will still give a useful answer. We might even go further than this, and try to analyze exactly when BP fails for a random constraint satisfaction problem. For example, you can do this for k-SAT when k is large, and then learn something about the solution threshold for k-SAT. It therefore might be natural to try and study when BP succeeds and fails for different kinds of problems.

Today, we'll actually only be considering models where each ψ_a only depends on two variables, which allows us to simplify our factor graph and consider the belief propagation for this special case.

3.2 Deriving BP

We're given an undirected tree representing our probabilistic model, precisely $P(\vec{x}) \propto \prod_{i,j} f_{i,j}(x_i, x_j) \prod_i f_i(x_i)$. How to we compute the marginals?

Notice that in our graphical model, for any variable x_i, if we condition on $x_i = r$, then all the children of x_i are conditionally independent. This suggests a natural "tree recursion" approach to solving the problem. To facilitate this observation, let's define "messages" $\psi_r^{i \to j}$, which we interpret as the marginal $P(x_i = r)$ in the model where the edge $i \to j$ has been removed from our graph (or equivalently, the model where all nodes on the non-reversing path starting at $i \to j$ have been removed). Let's denote the associated subtree of this marginal by $T^{i \to j}$. We let $V_{i \to j}$ be the set of variables occurring in $T^{i \to j}$, and let $T^{i \to j}(V_{i \to j}) := \prod_{l,w \in T^{i \to j}} f_{l,w}(x_l, x_w) \prod_{w \in V_{i \to j}} f_w(x_w)$ be the associated probability distribution up to normalizing constant, i.e. the product of the edges and nodes of $T^{i \to j}$.

Now suppose i has children $(\partial i) - j$. Then we can compute the marginal $\psi_r^{i \to j}$ directly:

$$\begin{aligned}
\psi_{x_i}^{i \to j} &\propto \sum_{V_{i \to j} - i} T^{i \to j}(V_{i \to j}) && (3)\\
&= \sum_{(\partial i) - j} \sum_{V_{i \to j} - i - \partial i} f_i(x_i) \prod_{k \in (\partial i) - j} f_{i,k}(x_i, x_k) T^{k \to i}(V_{k \to i}) && (4)\\
&= f_i(x_i) \sum_{(\partial i) - j} \prod_{k \in (\partial i) - j} f_{i,k}(x_i, x_k) \sum_{V_{k \to i} - k} T^{k \to i}(V_{k \to i}) && (5)\\
&\propto f_i(x_i) \sum_{(\partial i) - j} \prod_{k \in (\partial i) - j} f_{i,k}(x_i, x_k) \psi_{x_k}^{k \to i} && (6)\\
&= f_i(x_i) \prod_{k \in (\partial i) - j} \sum_{x_k \in \chi} f_{i,k}(x_i, x_k) \psi_{x_k}^{k \to i} && (7)
\end{aligned}$$

This is a purely algebraic way of computing this. A more intuitive way to get this result is as follows: suppose we know the marginals of all of i's children in each of the graphical models rooted at the children. Consider the graphical model which consists only of node i and $\partial(i) - j$, where the probabilities of each child are given by their marginals. The probability density associated with this model is proportional to

$$f_i(x_i) \prod_{\substack{k:(i,k) \in E, \\ k \neq j}} \psi_{x_k}^{k \to i} f_{i,k}(x_i, x_k)$$

Then we have

$$\psi_r^{i\to j} \propto \sum_{(\partial i)-j} f_i(r) \prod_{\substack{k:(i,k)\in E,\\ k\neq j}} \psi_{x_k}^{k\to i} f_{i,k}(r,x_k) \tag{8}$$

$$= f_i(r) \prod_{\substack{k:(i,k)\in E,\ x_k\in\chi\\ k\neq j}} \sum \psi_{x_k}^{k\to i} f_{i,k}(r,x_k) \tag{9}$$

Where the assumption we used was that if we condition on x_i, then the children of x_i are independent. It's useful to keep this assumption in mind when thinking about how BP behaves on more general graphs. (*)

A similar calculation yields that we can calculate the marginal $P(x_i = r) = \psi_r^i$ as

$$\psi_r^i \propto f_i(r) \prod_{k:(i,k)\in E} \sum_{x_k\in\chi} \psi_{x_k}^{k\to i} f_{i,k}(r,x_k) \tag{10}$$

This discussion easily leads to the following proposition:

Proposition 1. *Suppose we initialize messages $\psi_r^{i\to j}$ arbitrarily, and update them in parallel according to (4). Then if the factor graph has diameter d, then after d steps each $\psi_r^{i\to j}$ converges, and (5) gives the correct marginals.*

4 Free Energy Minimization

4.1 Potts model (Refresh of last week)

We've just seen a statistical/algorithmic view of how to compute marginals in a factor graph. It turns out that there's also a physical way to think about this, which leads to a qualitatively similar algorithm. Recall from last week that another interpretation of a pairwise factor-able PDF is that of particles interacting with each other via pairwise forces. In particular, we can imagine each particle x_i interacting with x_j via a force of strength

$$J_{(i,j)}(x_i,x_j) := \ln f_{i,j}(x_i,x_j) \tag{11}$$

and in addition, interacting with an external field

$$h(x_i) := \ln f_i(x_i) \tag{12}$$

we imagine that each of our particles take values from a discrete set χ. When $\chi = \{0,1\}$, we recover the Ising model, and in general we have a Potts model. The energy function of this system is then

$$E[\vec{x}] = -\sum_{(i,j)} J_{(i,j)}(x_i,x_j) - \sum_i h_i(x) \tag{13}$$

with probability distribution given by

$$P(\vec{x}) = \frac{1}{Z} e^{-E[\vec{x}]/T} \tag{14}$$

where we recover the original probability distribution when $T = 1$. Now, for $\chi = \{0,1\}$, computing the marginals $P(x_i)$ corresponds to the equivalent physical problem of computing the "magnetizations" $P(x_i = 1) - P(x_i = 0)$.

4.2 Free energy minimization

We're now going to try a different approach to computing the marginals: let's define a distribution $b(\vec{x})$, which we will hope to be a good approximation to $P(\vec{x})$. We can measure the "distance" between P and b by the KL divergence

$$KL(b(\vec{x})||p(\vec{x})) = \sum_{\vec{x}} b(\vec{x}) \ln \frac{b(\vec{x})}{P(\vec{x})} \tag{15}$$

$$= \sum_{\vec{x}} b(\vec{x}) E(\vec{x}) + \sum_{\vec{x}} b(\vec{x}) \ln b(\vec{x}) + \ln Z \tag{16}$$

which equals 0 iff the two distributions are equal. Let's define the Gibbs free energy as

$$G(b(\vec{x})) = \sum_{\vec{x}} b(\vec{x}) E(\vec{x}) + \sum_{\vec{x}} b(\vec{x}) \ln b(\vec{x}) =: U(b(\vec{x})) - S(b(\vec{x})) \tag{17}$$

So the minimum value of G is $F = -\ln Z$ which is called the Helmholz free energy, U is the "average energy" and S is the "entropy".

Now for the "free energy minimization part". We want to minimize G as a function of b, so that we can have that b is a good approximation of P. If this happens, then maybe we can hope to "read out" the marginals of b directly. How do we do this in a way which makes it easy to "read out" the marginals? We'll do this by trying to write G as a function of the marginals $b(x_i, x_j)$ and $b(x_i)$ of b. We then try to minimize G by optimizing over values for $b(x_i, x_j)$ and $b(x_i)$ under the consistency constraints $b(x_i) = \sum_{x_j \in \chi} b(x_i, x_j)$. Here's some algebra which shows us how far this endeavor takes us:

$$U(b(\vec{x})) = \sum_{\vec{x}} b(\vec{x}) \left(- \sum_{i,j} J_{i,j}(x_i, x_j) - \sum_i h_i(x_i) \right) \tag{18}$$

$$= - \sum_{i,j} \sum_{x_i, x_j} b(x_i, x_j) J_{i,j}(x_i, x_j) - \sum_i \sum_{x_i} b(x_i) h_i(x_i) \tag{19}$$

This is good news: since P only depends on pairwise interactions, the average energy component of G only depends on $b(x_i, x_j)$ and $b(x_i)$. However, it is not so clear how to express the entropy as a function of one node and two node beliefs. However, maybe we can try to pretend that our model is really a "tree". In this case, the following is true:

Claim 1. *If our model is a tree, and $\{b(x_i, x_j)\}_i$ and $\{b(x_i)\}_i$ are the associated marginals of our probabilistic model $b(\vec{x})$, then we have*

$$b(\vec{x}) = \frac{\prod_{i,j} b(x_i, x_j)}{\prod_i b(x_i)^{q_i - 1}} \tag{20}$$

where q_i is the degree of vertex x_i in the tree.

Proof. Imagine a tree rooted at x_i, with children ∂i. We can think of sampling from this tree as first sampling from x_i via its marginal $b(x_i)$, and then by recursively sampling the children conditioned on x_i. Associate with $\{T_k\}_{k \in \partial i}$ the subtrees of the children of i, i.e. $T_j(V_j)$ is equal to the probability of the occurrence V_k on the probabilistic model of the tree rooted at vertex j. Then we have

$$b(\vec{x}) = b(x_i) \prod_{k \in \partial i} b(x_k | x_i) \prod_{k \in \partial i} T_k(V_k | x_k) \tag{21}$$

$$= b(x_i) \frac{1}{b(x_i)^{q_i}} \prod_{k \in \partial i} b(x_i, x_k) \prod_{k \in \partial i} \frac{1}{b(x_k)} \prod_{k \in \partial i} T_k(V_k) \tag{22}$$

$$= \frac{\prod_{(i,j) \in E} b(x_i, x_j)}{\prod_{i \in [N]} b(x_i)^{q_i - 1}} \tag{23}$$

where the last line follows inductively, since each T_k only sees $q_k - 1$ edges of x_k. □

If we make this assumption, then we can write the Bethe approximation entropy as

$$S_{Bethe} = -\sum_{i,j}\sum_{x_i,x_j} b(x_i,x_j)\ln(b(x_i,x_j)) + \sum_i (q_i-1)\sum_{x_i} b(x_i)\ln b(x_i) \tag{24}$$

And we define the Bethe free energy as $U + S_{Bethe}$. Bethe free energy is in general not an upper bound on the true free energy. Note that if we make the assignments $E_i(x_i) = h_i(x_i)$, $E_{i,j}(x_i,x_j) = -J_{i,j}(x_i,x_j) - h_i(x_i) - h_j(x_j)$, then we can rewrite U as

$$U = \sum_{i,j}\sum_{x_i,x_j} b(x_i,x_j)E_{i,j}(x_i,x_j) + \sum_i (q_i-1)\sum_{x_i} b(x_i)E_i(x_i) \tag{25}$$

which is similar in form to the Bethe free energy. In general, we have

$$G_{Bethe}(b(x_i),b(x_i,x_j)) = \sum_{i,j}\sum_{x_i,x_j} b(x_i,x_j)(E_{i,j}(x_i,x_j) + \ln(b(x_i,x_j))) - \sum_i (q_i-1)\sum_{x_i} b(x_i)(E_i(x_i) + \ln b(x_i)) \tag{26}$$

which is exactly the Gibbs free energy for pairwise MRF with no loops. Since BP gives the correct marginals on trees, we can say that the BP beliefs are the global minima of the Bethe free energy. However, the following is also true

Proposition 2. *A set of beliefs gives a BP fixed point in any graph iff they are local stationary points of the Bethe free energy.*

Proof. See page 20 of [4]. □

5 The block model

5.1 Definitions

This section will heavily follow the paper [2]. The stochastic block model is designed to capture a variety of interesting problems, depending on its settings of parameters. The question we'll be looking at is the following: suppose we generate a random graph, where each vertex of the graph comes from one of q groups each with probability $n_1,...,n_q$. We add an edge between vertices (i,j) in groups a,b resp. with probability $p_{a,b}$. For sparse graphs, we define $c_{i,j} := Np_{i,j}$, where we think of $p_{i,j}$ as $\mathcal{O}(\frac{1}{N})$. The problem is the following: given such a random graph, can you label the vertices so that, up to permutation, the labels you choose have high correlation to the true hidden labels which were used to generate the graph? Here are some typical settings of parameters which represent different problems:

1. Community detection, where we have q groups. We set $n_i = \frac{1}{q}$, and $c_{i,j} = c_{in}$ if $i = j$ and c_{out} otherwise, with $c_{in} > c_{out}$ (assortative structure).
   1.1. It seems hard to do better than random choice when the average in degree is ≤ 7.
2. Planted graph partitioning, where $p_{i,j}$ may not necessarily be $\mathcal{O}(\frac{1}{N})$.
   2.1. When $p_{in} - p_{out} > \mathcal{O}(\log N/N)$, the planted partition is with high probability equal to the best partition.
   2.2. The planted partition can easily be found provided $p_{in} - p_{out} > N^{-\frac{1}{2}+\epsilon}$
   2.3. We are interested in cases where polynomial time algorithms can find partitions which are correlated with the true partition. This is possible when $p_{in} - p_{out} > \sqrt{qp_{in} + q(q-1)p_{out}}/\sqrt{N}$
3. Planted graph colouring, where $p_{in} = 0$.

We'll concern ourselves with the case where our graph is sparse, and we need to try and come up with an assignment for the vertices such that we have high correlation with the true labeling of vertices. In particular, if we come up with a labeling $\{t_i\}$, and the true labeling is $\{q_i\}$, then we'll measure our performance by

$$Q(\{t_i\},\{q_i\}) := \max_{\pi\in S_q} \frac{\frac{1}{N}\sum_{i\in[N]}\delta_{t_i,\pi(q_i)} - max_{a\in[q]}n_a}{1-\max_{a\in[q]} n_a} \tag{27}$$

where we maximize over all permutations π. Thus we have $Q = 1$ when we choose a labeling which (up to permutation) agrees with the true labeling, and $Q = 0$ when we trivially guess that every vertex belongs to the largest group.

Given $\{c_{a,b}\}, \{n_a\}$, we can write down the probability of a labeling $\{q_i\}$ as

$$P(G, \{q_i\}) \propto \prod_{\substack{(i,j) \notin E \\ i \neq j}} (1 - p_{q_i,q_j}) \prod_{(i,j) \in E} p_{q_i,q_j} \prod_{i \in [q]} n_{q_i} \tag{28}$$

How might we try to infer $\{q_i\}$ such that we have maximum correlation (up to permutation) with the true labeling? The answer is to use (up to symmetry breaking) the mle of the marginal distribution of each q_i.

5.2 Cavity Method

Let's proceed by applying BP to the first of these parameter settings. Suppose we're given a random graph with edge list E. What does our probabilistic model look like? Well, in this case, every variable is actually connected to every other variable, so we have a fully connected graph. However, some of the connections between variables are much weaker than others. In full, our BP update equations are

$$\psi_{t_i}^{i \to j} = \frac{1}{Z^{i \to j}} n_{t_i} \prod_{\substack{(i,k) \notin E \\ k \neq j}} \left[\sum_{t_k \in [q]} \left(1 - \frac{c_{t_1,t_k}}{N}\right) \psi_{t_k}^{k \to i} \right] \prod_{\substack{(i,k) \in E \\ k \neq j}} \left[\sum_{t_k \in [q]} c_{t_i,t_k} \psi_{t_k}^{k \to i} \right] \tag{29}$$

$$= \frac{1}{Z^{i \to j}} n_{t_i} \prod_{\substack{(i,k) \notin E \\ k \neq j}} \left[1 - \frac{1}{N} \sum_{t_k \in [q]} c_{t_1,t_k} \psi_{t_k}^{k \to i} \right] \prod_{\substack{(i,k) \in E \\ k \neq j}} \left[\sum_{t_k \in [q]} c_{t_i,t_k} \psi_{t_k}^{k \to i} \right] \tag{30}$$

Likewise

$$\psi_{t_i}^{i} = \frac{1}{Z^{i}} n_{t_i} \prod_{(i,k) \notin E} \left[\sum_{t_k \in [q]} \left(1 - \frac{c_{t_1,t_k}}{N}\right) \psi_{t_k}^{k \to i} \right] \prod_{(i,k) \in E} \left[\sum_{t_k \in [q]} c_{t_i,t_k} \psi_{t_k}^{k \to i} \right] \tag{31}$$

what we want to do is approximate these equations so that we only have to pass messages along the actual edges of our graph. The first observation is the following: Suppose we have two nodes $j, j' \in [N]$ such that $(i,j), (i,j') \notin E$. Then we see that $\psi_{t_i}^{i \to j} = \psi_{t_i}^{i \to j'} + \mathcal{O}\left(\frac{1}{N}\right)$, since the only difference between these two variables are two factors of order $(1 - \mathcal{O}\left(\frac{1}{N}\right))$ which appear in the first product of the BP equations. Thus, we send essentially the same messages to non-neighbours j of i in our random graph. In general though, we have:

$$\psi_{t_i}^{i \to j} = \frac{1}{Z^{i \to j}} n_{t_i} \prod_{\substack{(i,k) \notin E \\ k \neq j}} \left[1 - \frac{1}{N} \sum_{t_k \in [q]} c_{t_1,t_k} \psi_{t_k}^{k \to i} \right] \prod_{\substack{(i,k) \in E \\ k \neq j}} \left[\sum_{t_k \in [q]} c_{t_i,t_k} \psi_{t_k}^{k \to i} \right] \tag{32}$$

$$\approx \frac{1}{Z^{i \to j}} n_{t_i} \prod_{k \in [N]} \left[1 - \frac{1}{N} \sum_{t_k \in [q]} c_{t_1,t_k} \psi_{t_k}^{k \to i} \right] \prod_{\substack{(i,k) \in E \\ k \neq j}} \left[\sum_{t_k \in [q]} c_{t_i,t_k} \psi_{t_k}^{k \to i} \right] \tag{33}$$

$$\approx \frac{1}{Z^{i \to j}} n_{t_i} e^{-h_{t_i}} \prod_{k \in (\partial i) - j} \left[\sum_{t_k \in [q]} c_{t_i,t_k} \psi_{t_k}^{k \to i} \right] \tag{34}$$

The first approximation follows because we expect the number of neighbours of i to be constant. We've also defined a variable

$$h_{t_i} := \frac{1}{N} \sum_{k \in [N]} \sum_{t_k \in [q]} c_{t_i,t_k} \psi_{t_k}^{k \to i} \tag{35}$$

and we've used the approximation $e^{-x} \approx 1 - x$ for small x. We think of the term h_{t_i} as defining an "auxiliary external field". We'll use this approximate BP equation to find solutions for our problem. This has the advantage that the computation time is $\mathcal{O}(|E|)$ instead of $\mathcal{O}(N^2)$, so we can deal with large sparse graphs computationally. It also allows us to see how a

large dense graph with only sparse strong connections still behaves like a sparse tree-like graph from the perspective of Belief Propagation.

From now on, we'll only consider factored block models, which in some sense represent a "hard" setting of parameters. These are models which satisfy the condition that each group q has the same average degree c. In particular, we require

$$\sum_{d=1}^{q} c_{a,d} n_d = \sum_{d=1}^{q} c_{b,d} n_d = c \tag{36}$$

An important observation for this setting of parameters is that

$$\psi_{t_i}^{i \to j} = n_{t_i} \tag{37}$$

is always a fixed point of our BP equations, which is known as a factored fixed point. When BP ever reaches such a fixed point, we get that $Q = 0$ and the algorithm fails. However, we might hope that if we randomly initialize $\{\psi^{i \to j}\}$, then BP might converge to some non-trivial fixed point which gives us some information about the original labeling of the vertices.

5.3 Numerical Results

Now that we have our BP equations, we can run numerical simulations to try and get a feel of when BP works. We'll set our parameters with all group sizes n_a being equal, and with $c_{a,a} = c_{in} > c_{out} = c_{a,b}$ for $a \neq b$ and vary the ratio $\epsilon := c_{out}/c_{in}$, and see when BP finds solutions which are correlated "better than guessing" to the original labeling used to generate the graph. When we do this, we get images which look like this:

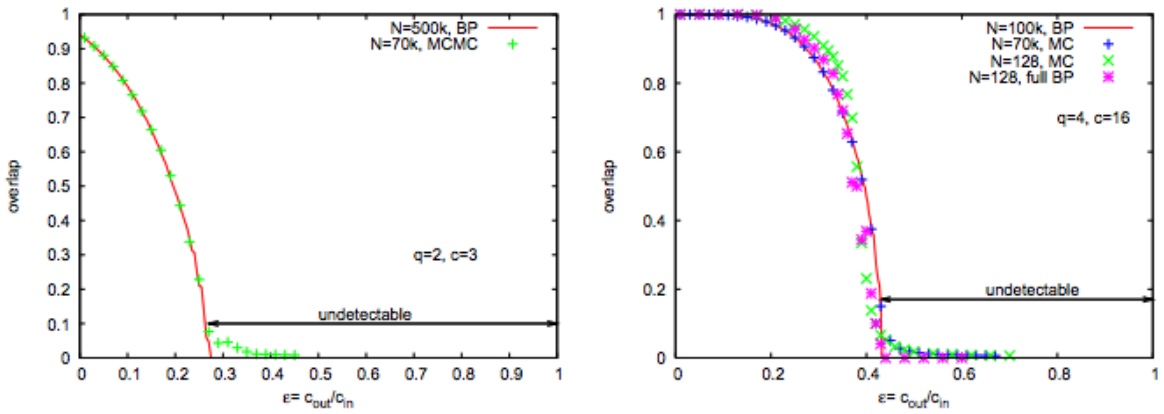

Figure 1: Image taken from [2]

In general, we see a thresholding behavior at a certain value independent of N. We also see that our approximate BP performs the same as running a full BP, and that BP gives the same performance as MCMC sampling.

5.4 Heuristic Stability Calculations

How might we analytically try to determine when BP fails for certain settings of q and $\{c_{a,b}\}$? One way we might heuristically try to do this, is to calculate the stability of the factored fixed point. If the fixed point is stable, this suggests that BP will converge to a factored point. If however it is unstable, then we might hope that BP converges to something informative.

Following in this spirit, here's a heuristic way of calculating the stability of the factored fixed point. Let's pretend that our BP equations occur on a tree, which is a reasonable approximation in the sparse graph case. Let our tree be rooted at node k_0 and have depth d. Let's try to approximately calculate the influence on k_0 of perturbing a leaf k_d from its factored fixed

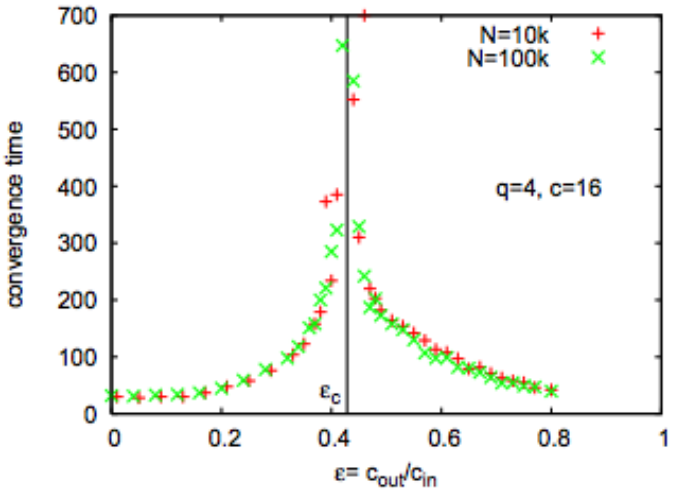

Figure 2: Image taken from [2]

point. In particular, let the path from the leaf to the root be $k_d, ..., k_0$. We're going to apply a perturbation $\psi_t^{k_d} = n_t + \epsilon_t^k$ for each $t \in [q]$. The next thing we'll do is define the matrix

$$T_i^{a,b} := \frac{\partial \psi_a^{k_i}}{\partial \psi_b^{k_{i+1}}}|_{\psi_t = n_t} \tag{38}$$

$$\approx \left[\frac{\psi_a^{k_i} c_{a,b}}{\sum_{r\in[q]} c_{a,r}\psi_r^{k_{i+1}}} - \psi_a^{k_i} \sum_{s\in[q]} \frac{\psi_s^{k_i} c_{s,b}}{\sum_{r\in[q] c_{s,r}\psi_r^{k_{i+1}}}} \right] |_{\psi_t = n_t} \tag{39}$$

$$= \left[\frac{n_a c_{a,b}}{c} - n_a \frac{c}{c} \right] \tag{40}$$

$$= n_a \left(\frac{c_{a,b}}{c} - 1 \right) \tag{41}$$

The perturbation effect on $\epsilon_{t_0}^{k_0}$ is then

$$\sum_{\{t_i\}_{i=1,...,d}} \left[\prod_{i=0}^{d-1} T_i^{t_i, t_{i+1}} \right] \tag{42}$$

Since $T_i^{a,b}$ does not depend on i, we can write our perturbation $\epsilon^{k_d} = (\epsilon_1^{k_d}, ..., \epsilon_q^{k_d})^T$ as a vector and write

$$\epsilon^{k_0} = T^d \epsilon^{k_d} \approx \lambda^d \epsilon^{k_d} \tag{43}$$

(this is just the chain rule) where λ is the largest eigenvalue of T. Now, on a random tree, we have approximately c^d leaves. If we assume that the perturbation effect from each leaf is independent, and that ϵ^{k_d} has 0 mean, then the net mean perturbation from all the leaves will be 0. The variance will be

$$\left\langle (\epsilon_{t_0}^{k_0})^2 \right\rangle \approx \left\langle \left(\sum_{l=1}^{c^d} \lambda^d \epsilon_{t_0}^l \right)^2 \right\rangle \tag{44}$$

$$= c^d \lambda^{2d} \left\langle (\epsilon_{t_0}^{k_d})^2 \right\rangle \tag{45}$$

if we assume that the cross terms vanish in expectation (if we perturbed all the $\psi_t^{k_d}$ in the same direction, our normalization conditions would cancel this out). We then end up with the stability condition $c\lambda^2 = 1$. If we restrict our attention to graphs of the form $c_{a,a} = c_{in} > c_{a,b} = c_{out}$ for $a \neq b$, and have all our groups with size $n_a = \frac{1}{q}$, then T is known to have eigenvalues

$\lambda_1 = 0$ with eigenvector $(1, ..., 1)$, and $\lambda_2 = (c_{in} - c_{out})/qc$. The stability threshold then becomes $|c_{in} - c_{out}| = q\sqrt{c}$. This condition is known as the Almeida-Thouless local stability condition for spin glasses, and the Kesten-Stigum bound on reconstruction on trees. It is also observed empirically that BP and MCMC succeed above this threshold, and converge to factored fixed points below this threshold.

5.5 Information Theoretic Results, and what we know algorithmically

We've just seen a threshold for when BP is able to solve the community detection problem. Specifically, when $c < \frac{1}{\lambda^2}$, BP doesn't do better than chance. It's natural to ask whether this is because BP is not powerful enough, or whether there really isn't enough information in the random graph to recover the true labeling.

It turns out that for $q = 2$, there is not enough information below the threshold $c < \frac{1}{\lambda^2}$ [3]. However, it can be shown information-theoretically [1] that the true critical value at which one can find a correlated labeling is $c_c = \Theta\left(\frac{\log(q)}{q\lambda^2}\right)$. In particular, when $q > 11$, there exists exponential time algorithms which recover a correlated labeling below the Kesten-Stigum threshold.

Moreover, consider the planted colouring problem, where $c_{in} = 0$. Then our bound becomes $c > (q-1)^2$, but it is known how to do better when $c > 2q\ln(q)$ [2].

6 References

References

[1] Jess Banks, Cristopher Moore, Joe Neeman, Praneeth Netrapalli, *Information-theoretic thresholds for community detection in sparse networks.* AJMLR: Workshop and Conference Proceedings vol 49:1–34, 2016. Link

[2] Aurelien Decelle , Florent Krzakala , Cristopher Moore , Lenka Zdeborov, *Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications.* 2013. Link

[3] Cristopher Moore, *The Computer Science and Physics of Community Detection: Landscapes, Phase Transitions, and Hardness.* 2017. Link

[4] Jonathan Yedidia, William Freeman, Yair Weiss, *Understanding Belief Propogation and its Generalizations.* Link.

[5] Afonso Banderia, Amelia Perry, Alexander Wein *Notes on computational-to-statistical gaps: predictions using statistical physics.* Link

[6] Stephan Mertens, Marc Mezard, Riccardo Zecchina *Threshold Values of Random K-SAT from the Cavity Method.* Link

[7] Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian *Clusters of solutions and replica symmetry breaking in random k-satisfiability.* Link