

Masters Project

Matching Recovery

Department of Mathematics
Kings College London
London
2021

Abstract

This paper is separated into two chapters. The first chapter provides initial material for the reader. This includes a brief introduction to statistical mechanics, along with its application to combinatorial optimisation. The technique known as Belief Propagation, useful for studying such optimisation problems, is then derived and applied to an example case. Statistical Inference is later addressed, the reader being provided an outline of the subject. Finally statistical inference is applied to dimer models, specifically the planted matching problem and discussed in depth through a review of relevant previous work.

The second chapter contains my novel work. We consider the transition point between partial and full recovery of a planted perfect matching in both the random regular graph of coordination c and the square lattice by means of a Student-Teacher statistical inference problem, with both graph structures being subject to varying levels of suppression of edges. Previous work shows analysis of the transition for sparse graphs. We look to extend this work and provide evidence for the similar behaviour we expect to be exhibited between the square lattice and the random regular lattice of coordination $c = 3$. This is achieved by considering an analytic approach, obtaining a boundary equation between partial and full recovery for the random regular graph and then a numerical approach, providing graphical evidence of similar behaviour.

Contents

Chapter 1. Statistical physics, optimisation, and inference	5
1. Introduction	5
2. A probabilistic approach	6
3. Computational Optimisation	7
4. Belief-Propagation	8
5. Assignment Problem	11
6. Statistical Inference	13
7. The planted matching problem	14
Chapter 2. Inferring planted matchings on lattices	19
1. Establishing inference in Graphs	19
2. Boundary of phase transition	20
3. Numerical Investigation	24
Appendix A. Supplementary material	31
1. Maximum Entropy Inference	31
2. On the $\beta \rightarrow 0$ limit	32
3. On the $\beta \rightarrow \infty$ limit	32
4. Factor graph of probability	33
Appendix. Bibliography	35
Appendix B. Self Assessment	37

CHAPTER 1

Statistical physics, optimisation, and inference

1. Introduction

Complex systems often exhibit various states in which they exist. These states correspond to different macroscopic properties to be exhibited by the system, many of which are the properties we naturally experience. To illustrate this, take water as an example. Water can exist in three states of matter¹: in the form of ice as a solid, as a liquid at room temperature and one atmosphere, and finally in an evaporated or gaseous form. All three states exhibit vastly different behaviour. As a consequence, one may be tempted to imagine the changes on a macroscopic level are reflective of possible changes at a microscopic level. This is however not the case. The disparate properties of states of matter are solely due to the microscopic interactions between the elementary components. Returning to our example, we know ice is formed if a body of water is brought to a low enough temperature and evaporation is caused if that same body of water is taken to a high enough temperature (keeping pressure fixed). In all cases the atoms of water do not change, only the energy they have as temperature increases or decreases. Water is but one example of complex systems in general; the nature of the interactions on a microscopic level remain the same, yet by increasing parameters on a macroscopic level, we force macroscopic properties to change drastically.

It is exactly this type of phenomena statistical physicists look to understand. Given a complex system, we can use the concepts and methods developed in statistical physics to describe not only the various macroscopic properties but also the location and details of the transitions between states.

Statistical physics is primarily founded on two principal axioms. The first principal idea is a comprehensive and detailed understanding of the microscopic interactions is not required for a description of a system. Many complex systems undergo far too many interactions for it to be feasible to know them all. Returning to the example of water, there are more than 10^{22} molecules in just a gram of pure water. Knowing the information of all interactions of just a gram of water is a task well beyond the capabilities we currently have. The concept adopted by statistical physics, is instead to assume the system at a microscopic level is too chaotic, forcing an alternative purely probabilistic description. The system in question would then exist in configurations, one for each possible outcome of the system at any given time. The second principal idea is that the macroscopic descriptions described by the probabilistic basis occur as a result of the law of large numbers. While on a microscopic level the details of each interaction may differ, due to the sheer number of them, fluctuations are negated by each other and the macroscopic descriptions ‘average’ out.

¹It is well known that the phase diagram of water is actually very complicated, but we simplify here for the sake of simplicity referring to the common experience.

2. A probabilistic approach

To begin our probabilistic description we start by defining our configuration space Ω which is the set of all possible configurations σ of our system. A weight function $w : \sigma \rightarrow w(\sigma)$ must also be defined over our system. When referring to configurations, we often refer to specifically the weighted configurations. The next step is to define a suitable cost or energy function, $E(\sigma) : \Omega \rightarrow \mathbb{R}$. This is any function taken from the set of observables, where an observable, $\mathcal{O} : \Omega \rightarrow \mathbb{R}$, is a function of the weighted configuration space. This definition allows for a variety of energy functions to be used. At this point, let us assume there is no more information we know about the complex system other than the weighted configuration space defined and the suitable energy function chosen. It may seem insufficient knowledge but in most cases this assumption is accurate as information about systems is often too difficult to collect. Furthermore a weighted configuration space and a suitable energy function are all that is needed to define a probabilistic description of our system. The only requirement our distribution must satisfy is that it does not assume additional information about our system.

To avoid this, we must find the distribution $p(\sigma)$ over the configuration space, given the arbitrary constraint $\langle E(\sigma) \rangle = E \in \mathbb{R}$, on configurations σ , that maximises entropy. Entropy, defined as:

$$(1) \quad S[p] = \sum_{\Omega} p(\sigma) \cdot \ln \left(\frac{1}{p(\sigma)} \right) = - \sum_{\Omega} p(\sigma) \cdot \ln(p(\sigma))$$

is a measure of information or determinism in a system, where high entropy indicates low determinism of events. To remove potential assumption about the distribution of events, by using the principle of Maximum Entropy Inference², we obtain the equation for the distribution that maximises the entropy of the system, along with the partition function, Z :

$$(2) \quad P(\sigma) = \frac{1}{Z} e^{-\beta E(\sigma)}, \quad Z = \sum_{\Omega} e^{-\beta E(\sigma)}$$

We define $\beta = \frac{1}{T}$ to be the inverse fictitious temperature. Thus we see this form of the probability distribution is similar to that of the Boltzmann distribution, where we have Z being some normalisation constant. It is interesting to note if we take $\beta \rightarrow 0$ the distribution of P tends to a uniform distribution³:

$$(3) \quad \lim_{\beta \rightarrow 0} P(\sigma) = \frac{1}{|\Omega|}$$

If, on the other hand $\beta \rightarrow \infty$ we find the distribution tends to⁴:

$$(4) \quad \lim_{\beta \rightarrow \infty} P(\sigma) = \frac{1}{|\Omega_0|} \mathbb{I}(\sigma \in \Omega_0)$$

where we have defined $\Omega_0 \in \Omega$ to be the set of configurations that minimise the energy function. What we find from this is in infinitely high temperature systems, each configuration becomes equally as likely, but as temperature tends to zero, the probability distribution concentrates on lower energy configurations and in the limit becomes uniform again, but is dominated by the configurations that minimise the energy function.

²For a derivation, see Appendix 1

³For a derivation, see Appendix 2

⁴For a derivation, see Appendix 3

3. Computational Optimisation

With the knowledge to implement a complex system in a purely probabilistic environment, we can look to investigate characteristics of our systems in more detail. Three examples of characteristics of a system one may wish to explore can be studied through the optimisation problem, the evaluation problem and the decision problem. To be more precise:

- Optimisation problem: Obtains the configuration, σ_{OPT} of the system that minimises the energy function E
- Evaluation problem: Evaluates the optimal cost, $E_{OPT} = E(\sigma_{OPT})$, of the energy function for the optimal configuration
- Decision problem: determines whether there exists a configuration in the configuration space with cost below a given value

The search for the solution to any one of these problems is not without its difficulties though. As is the case with large systems, investigations like these can be tedious to do by hand, so we employ the use of computational means to assist the process. This adds a new element to the problem: not only is it necessary to solve the problem, but the formulation of the solution needs to be considered so that it can be performed as an algorithm. To illustrate how this is done, let us consider the problem of the Minimal spanning tree.

3.1. Minimal Spanning Tree. The Minimal Spanning Tree problem, or MST, is as follows: consider you are given an undirected graph \mathcal{G} with vertex set \mathcal{V} and edge set \mathcal{E} , where each edge, $e \in \mathcal{E}$, is weighted with weight $w_e \in \mathcal{W}$. The cost, E of a subgraph, $\hat{\mathcal{G}}$, is defined as the sum of the weights on all edges in the subgraph: $E(\hat{\mathcal{G}}) = \sum_{e \in \hat{\mathcal{E}}} w_e$. The optimisation problem is simply to find the spanning tree \mathcal{T} - the subgraph of \mathcal{G} with vertex set equal to \mathcal{V} and containing no cycles - of lowest cost. A first approach may be to consider all the trees of graph \mathcal{G} , calculate the cost of each and compare them with each other to see which is the lowest, resulting in the identification of \mathcal{T} . After attempting this however, one will find for large graphs, simply identifying all possible trees of a graph becomes a formidable task. Consider the complete graph with N vertices. We know, from Cayley's [13], the number of spanning trees in this complete graph of N nodes is N^{N-2} . This grows exponentially as N increases, so as a general solution for any graph, this method isn't a feasible option. Thankfully, computer scientists and mathematicians have found faster and more efficient algorithms to solve the MST problem, one of which being Prims Algorithm. Prims Algorithm, initially created by Vojtěch Jarník, a Czech mathematician, in 1930 but rediscovered by American mathematician and computer scientist Robert Clay Prim in 1957, defines a tree from a graph using an algorithm on the edges of the graph. It starts by choosing an initial vertex of the graph at random. This itself is a tree, \mathcal{T}' , and the algorithm expands this tree iteratively, one edge at a time, until it is a minimum spanning tree, \mathcal{T}_{MST} . It does this by, at each iteration, adding the edge of lowest weight connected to the tree \mathcal{T}' . After each iteration we are guaranteed the size of the vertex set of \mathcal{T}' increases by 1, so after $N - 1$ iterations, we are left with the minimum spanning tree, \mathcal{T}_{MST} . As pseudo-code, this algorithm follows the structure:

MST Algorithm: find \mathcal{T}_{MST} on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with weight function ω by constructing $\mathcal{T}' = (\mathcal{V}', \mathcal{E}')$:

- (1) set $\mathcal{V}' = \{0\}$, $E = 0$
- (2) while $\mathcal{V}/\mathcal{V}' \neq \emptyset$
- (3) $Z = \{v = (v_1, v_2) \in \mathcal{V} \mid v_1 \in \mathcal{V}', v_2 \in \mathcal{V}/\mathcal{V}'\}$
- (4) $e = \min_{z \in Z} w(z) = w(z')$
- (5) $E+ = e, \mathcal{V}' = \{v_1\}$
- (6) return $\mathcal{T}_{MST} = \mathcal{T}'$

The benefit of being able to write the algorithm in pseudo code is that the algorithm can then be translated into a computing language. We also note the problem is of polynomial complexity class as the run time is bounded above by $N^3/6$, N being the size of the vertex set of \mathcal{G} .

4. Belief-Propagation

There are many methods one can use to allow different systems to be studied, but one very useful way is called Belief Propagation. It is dependent upon certain conditions of the system, one of which is the necessity for the solution to behave as a factor graph.

4.1. Factor Graphs. In their most rudimentary form, factor graphs are a graphical representation of polynomials. Consider we have a factorisable polynomial $P(\underline{x}) = \prod_i p_i(\underline{x})$ where each factor p_i is itself a polynomial of a number of the entries of \underline{x} . It is worth noting these factors do not necessarily have to be irreducible. We can construct a factor graph \mathcal{F} representing $P(\underline{x})$ by formalising the structure of the graph in the following way:

- Variable nodes - these are nodes of the graph, indicated as circles, that represent the variables of the polynomial - the set of variables of $P(\underline{x})$ is $\{x_i \mid x_i \in \underline{x}\}$ so the set of vertices of \mathcal{F} will have the same size
- Factor nodes - these, indicated as squares, represent each distinct factor of our polynomial $P(\underline{x})$, the number of which will be the same as the number of factors.
- Edges - edges exist only between variable nodes and factor nodes, adjacent nodes indicating whether a factor is dependent upon a variable or not.

From this definition we note factor graphs are inherently bipartite graphs as factors cannot connect to factors and variables cannot connect to variables.

4.2. BP messages. The reformulation of polynomials into factor graphs can be used on probabilities, specifically probabilities which can be factorised⁵. Suppose we have a system with configuration space $\underline{x} = (x_1, \dots, x_N) \in \Omega = \prod_{i=1}^N \Omega_i$ defined through the probability function P with M factors:

$$(5) \quad P(\underline{x}) = \frac{1}{Z} \prod_{a=1}^M f_a(x_{\partial a})$$

where, considering the factor graph \mathcal{F} of P , $\partial a = \{x_i \mid x_i \text{ is adjacent to factor node } a\}$. To understand the effect each node has on the system, from this distribution, we can look to the marginal distributions of each variable node x_i for such a complete description. More specifically we look for:

$$(6) \quad P_i(x_i) = \sum_{\Omega_k \neq i} P(\underline{x}) = \frac{1}{Z} \sum_{\Omega_k \neq i} \prod_{a=1}^M f_a(x_{\partial a})$$

By direct calculation, this may not seem at first easily solvable and in many cases an exact solution is not analytically possible. It is however possible to approximate certain distributions, using an exact result for a particular instance of graph topology.

For us to derive an exact solution, let us assume the graph is in fact a tree. This implies there are no loops or cycles in our factor graph. Suppose then we choose, arbitrarily, variable node i , which could lie anywhere in our tree. Variable node i has neighborhood ∂i of n factor nodes $\{a_j\}_{j \in \{0, \dots, n\}}$. Our assumption of the topology of our graph being a tree allows the graph to be partitioned into sub trees. Take factor node a_j in ∂i . This factor node connects to i and also could connect to other variable nodes 'behind' it. We use the term 'behind' to classify the set of nodes, variable and factor alike, of the graph with shortest path to variable node i passing

⁵For an example, see Appendix 4

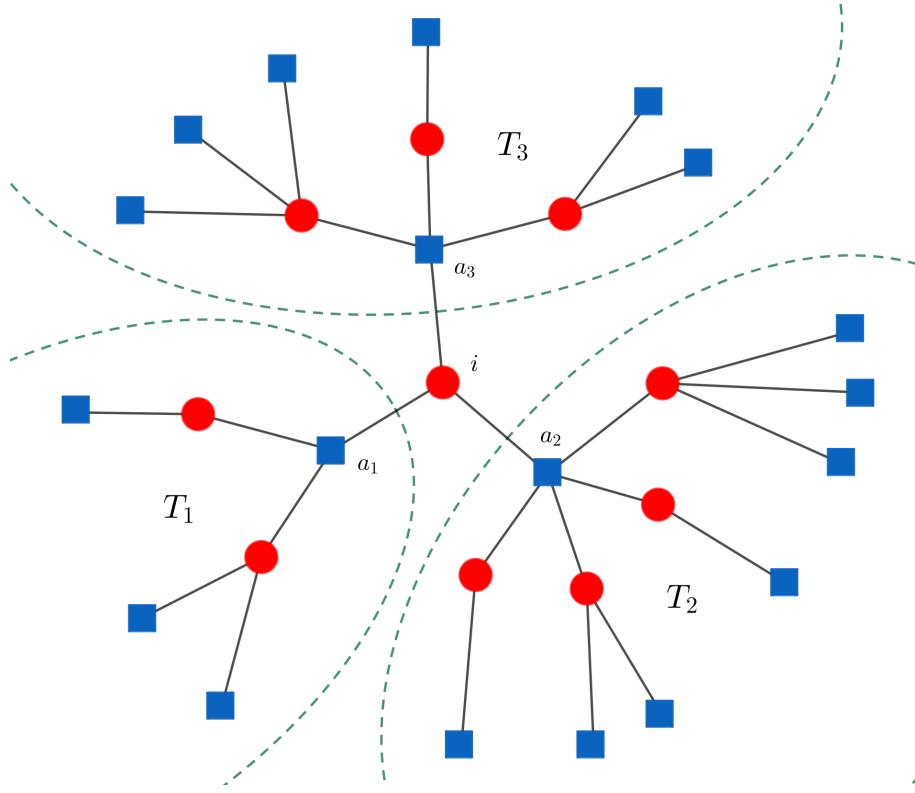


FIGURE 1. Example of division of factor graph into separate subtrees T_i for each factor node $a_i \in \partial i$

through factor node a_j . But our assumption constraint, preventing loops or cycles, means no nodes behind a_j connect to nodes behind other factor nodes $a_{k \neq j}$ in the neighborhood of i . To put it another way, if we remove the edges connecting variable i to its neighbors, we are left with n independent sub trees with roots a_j . As this graph is representative of our probability function, we see there exists a factorisation of $P(\underline{x})$ which groups the factors depending on which variables they are functions of - more specifically, these groups are non intersecting in their variables, aside from variable i upon which they all connect to in the graph, meaning they all use i as a variable. Formally, this can be expressed as follows. Let us define $T_j = (\mathcal{V}_{T_j}, \mathcal{E}_{T_j})$ to be the sub trees behind respective root factor nodes $a_j \in \partial i$. As the sub trees are independent of one another, we can define the polynomials:

$$(7) \quad T_j(\underline{x}_j) = \prod_{\mathcal{A}_j} (f_a(x_{\partial a})) = \left(\prod_{\mathcal{A}_j} f_a \right) (\underline{x}_j), \quad \mathcal{A}_j = \{\text{factor nodes in } T_j\}$$

$$(8) \quad \Rightarrow P(\underline{x}) = \frac{1}{Z} \prod_{j \in \partial i} T_j(\underline{x}_j)$$

such that \underline{x}_j is the set of variables $x_i \in \underline{x}$ which are in tree T_j . This allows the summation to easily factorise over the separate sub tree polynomials:

$$(9) \quad P_i(x_i) = \sum_{\Omega_k \neq i} P(\underline{x}) = \frac{1}{Z} \sum_{\Omega_k \neq i} \prod_{j \in \partial i} T_j(\underline{x}_j)$$

$$(10) \quad = \frac{1}{Z} \sum_{\Omega_k \in I(\underline{x}_1)} \dots \sum_{\Omega_k \in I(\underline{x}_n)} \prod_{j \in \partial i} T_j(\underline{x}_j)$$

$$(11) \quad = \frac{1}{Z} \prod_{j \in \partial i} \sum_{\Omega_k \in I(\underline{x}_j)} T_j(\underline{x}_j)$$

$$(12) \quad = \frac{1}{Z} \prod_{j \in \partial i} \hat{v}_{a_j \rightarrow i}(x_i)$$

$I(\underline{x}_j)$ denotes the set of indexes for variables in \underline{x}_j . In the final step we defined the incoming message $\hat{v}_{a_j \rightarrow i}(x_i) = \sum_{\Omega_k \in I(\underline{x}_j)} \prod_{\mathcal{A}_j} f_a(x_{\partial a})$. This expression, while more manageable than before, still contains many factors if the size of the subtree T_j is large. Instead we can look towards simplifying the incoming message by considering:

$$(13) \quad \hat{v}_{a_j \rightarrow i}(x_i) = \sum_{\Omega_k \in I(\underline{x}_j)} \prod_{\mathcal{A}_j} f_a(\underline{x}_{\partial a})$$

$$(14) \quad = \sum_{\Omega_k \in I(\underline{x}_j)} f_{a_j}(\underline{x}_{\partial a_j}) \prod_{\mathcal{A}_j / a_j} f_a(\underline{x}_{\partial a})$$

$$(15) \quad = \sum_{\Omega_k \in \partial a_j / i} f_{a_j}(\underline{x}_{\partial a_j}) \sum_{\Omega_k \in I(\underline{x}_j) / (\partial a_j / i)} \prod_{\mathcal{A}_j / a_j} f_a(\underline{x}_{\partial a})$$

This can be further rearranged by considering the sets $I(\underline{x}_j) / (\partial a_j / i)$ and \mathcal{A}_j / a_j . $I(\underline{x}_j) / (\partial a_j / i)$ is the set of all the rest of the variable nodes in T_j without the variables in the neighborhood of factor node a_j . Using the tree topology condition, we see this 'leftover' set of variables can be partitioned into smaller subsets. In order to do this, let us define another layer of sub trees, t_l in each T_j , with roots a_l which are defined as the factor nodes in the set $\partial k / a_j$ for all $k \in \partial a_j / i$. Just as before, these trees t_l can be considered as polynomials, whose inputs can be named \underline{x}_l . What we see is for all variables in $I(\underline{x}_j) / (\partial a_j / i)$, each is uniquely assigned one tree as no two trees can contain the same variable node, else the initial tree assumption will be contradicted. Therefore, the sub trees define the subsets into which the 'leftover' variables can be partitioned. The set \mathcal{A}_j / a_j can also be partitioned in a similar way by considering the sub trees t_l . We find $\mathcal{A}_j / a_j = \bigcup_{k \in \partial a_j / i} \bigcup_{l \in \partial k / a_j} \mathcal{A}_l$ with \mathcal{A}_l being similarly defined as the set of all factor nodes in tree t_l . This allows the expression of $\hat{v}_{a_j \rightarrow i}(x_i)$ to be written as:

$$(16) \quad \hat{v}_{a_j \rightarrow i}(x_i) = \sum_{\Omega_k \in \partial a_j / i} f_{a_j}(\underline{x}_{\partial a_j}) \sum_{\Omega_k \in I(\underline{x}_j) / (\partial a_j / i)} \prod_{\mathcal{A}_j / a_j} f_a(\underline{x}_{\partial a})$$

$$(17) \quad = \sum_{\Omega_k \in \partial a_j / i} f_{a_j}(\underline{x}_{\partial a_j}) \prod_{k \in \partial a_j / i} \prod_{l \in \partial k / a_j} \sum_{\Omega_k \in I(\underline{x}_l)} \prod_{\mathcal{A}_l} f_a(\underline{x}_{\partial a})$$

$$(18) \quad = \sum_{\Omega_k \in \partial a_j / i} f_{a_j}(\underline{x}_{\partial a_j}) \prod_{k \in \partial a_j / i} \prod_{l \in \partial k / a_j} \hat{v}_{a_l \rightarrow k}(x_k)$$

$$(19) \quad = \sum_{\Omega_k \in \partial a_j / i} f_{a_j}(\underline{x}_{\partial a_j}) \prod_{k \in \partial a_j / i} v_{k \rightarrow a_j}(\underline{x}_k)$$

In the penultimate line we have recognised the equation for \hat{v} can be substituted in for another variable node, in the case for all variable nodes $k \in \partial a_j / i$. The final step is to define the outgoing message $v_{k \rightarrow a_j}(\underline{x}_k) = \prod_{l \in \partial k / a_j} \hat{v}_{a_l \rightarrow k}(x_k)$.

What has been derived above is the technique known as Belief-Propagation. This technique relies on systems having a tree factor graph structure. If the system is in the suitable graph format, to implement Belief-Propagation, the calculation of the incoming and outgoing messages \hat{v} and v are all that is necessary. To do this, an iterative approach is used. Defining the iterative messages for neighboring factor and variable nodes a and i as:

$$(20) \quad \hat{v}_{a \rightarrow i}^t(x_i) = \sum_{\Omega_{k \in \partial a_j / i}} f_a(\underline{x}_{\partial a}) \prod_{k \in \partial a \setminus i} v_{k \rightarrow a}^t(x_k)$$

$$(21) \quad v_{i \rightarrow a}^{t+1}(x_i) = \prod_{b \in \partial i \setminus a} \hat{v}_{b \rightarrow i}^t(x_i)$$

these messages are calculated for all edges, computing first incoming edges $a \rightarrow i$ and then outgoing edges $i \rightarrow a$. The application of the BP algorithm is successful if the iterations converge to fixed points, which will be the final solution - as stated before, the marginals can be calculated, ideally once a threshold of convergence is reached, by taking the product of the incoming messages to each respective variable node. To run the algorithm though, an initial condition needs to be set, the common practice being to set the initial messages, $v_{i \rightarrow a}^0(x_i)$, to a uniform distribution over the configuration space Ω_i . For many graph structures, convergence to a solution is not guaranteed, but it is possible to prove the BP algorithm provides exact solutions on tree graphs. Furthermore, under certain conditions, solutions can be found for non tree factor graphs. Example conditions include graphs with short range correlations and long loop lengths to allow perturbations on nodes to diminish as they propagate through the graph, enough so to have negligible affect on the original perturbed node. These graphs would be considered locally tree like, an example being the random regular graph with vertex degree $c + 1$ such that $N \gg c$.

5. Assignment Problem

As an example of Belief-Propagation, the algorithm can be applied to the well known problem, the Assignment Problem.

Consider a set of N people P who are each about to be assigned a single job from a set of N jobs J . Each person has their own skills and their affinity with each of the N jobs is recorded in the job matrix E_{ij} . The assignment task is to assign each person to one job. This is recorded in the occupation matrix $\{n_{ij}\} \in \{0, 1\}^{N \times N}$, where we will denote the assignment of person i to job j as $n_{ij} = 1$. Once a matching of people to jobs is established, it is given a cost E which is simply the sum of all affinities in that given assignment. The problem is to find the assignment of people to jobs, the specific occupation matrix $\{n_{ij}\}^*$, which minimises the cost E .

To define the BP update rules we must first define a suitable probability distribution. Firstly we know there exist constraints on the system, one being that the assignment task can only assign at most 1 job to each person. This can be written in the form of a 'hard constraint' for a general assignment $\{n_{ij}\}$:

$$(22) \quad \forall i \in P, \quad \mathbb{I} \left(\sum_{j \in J} n_{ij} \leq 1 \right) = \begin{cases} 1, & \text{if } n_{ij} \text{ is a valid assignment} \\ 0, & \text{otherwise} \end{cases}$$

Similarly we see the converse is also true, one job can only be assigned one person to undertake it, and the hard constraint equation is the same, but set P is replaced with set J . To filter

the valid assignments, we can combine the two hard constraints into the following:

$$(23) \quad C_1(n_{ij}) = \prod_{j \in J} \mathbb{I} \left(\sum_{i \in P} n_{ij} \leq 1 \right) \prod_{i \in P} \mathbb{I} \left(\sum_{j \in J} n_{ij} \leq 1 \right) = \begin{cases} 1, & \text{if } n_{ij} \text{ is a valid assignment} \\ 0, & \text{otherwise} \end{cases}$$

The hard constraint removes many of the invalid assignments from the space $\{n_{ij}\} \in \{0, 1\}^{N \times N}$. It will be used to define the support of the function, refining the domain of the probability function to a smaller set. The hard constraint is not enough to fully define the support of P though as it allows for assignments which include unmatched peoples and jobs. To restrict the support of P fully we want only perfect matchings to be allowed to have non zero probabilities. To do this we can include a soft constraint. In general however this soft constraint is used to tackle problems other than just optimisation. The constraint is written as:

$$(24) \quad C_2(n_{ij}) = \prod_{i, j \in P, J} e^{2\beta\gamma n_{ij}}$$

where the 2 is included for ease of notation in future calculations. What is seen in the limit as $\gamma \rightarrow \infty$, the constraint becomes more concentrated on high occupation assignments. The most occupation allowed by the hard constraint C_1 is a 1 to 1 assignment, a total assignment of N , which corresponds to perfect matchings on the bipartite graph representation. The final constraint of P is the weight itself. C_3 takes the form of a soft constraint as well and by using the formulation of the probability function in the first section, we see it is of an exponential function. In the limit of $\beta \rightarrow \infty$ the constraint focuses on low cost assignments, which is exactly what we look to find. Thus, the probability distribution takes the final form:

$$(25) \quad P(\underline{n}) = \prod_{j \in J} \mathbb{I} \left(\sum_{i \in P} n_{ij} \leq 1 \right) \prod_{i \in P} \mathbb{I} \left(\sum_{j \in J} n_{ij} \leq 1 \right) \prod_{i, j \in P, J} e^{-\beta n_{ij}(E_{ij} - 2\gamma)}$$

where we have taken $\underline{n} \in \{0, 1\}^{N \times N}$. This distribution is in a suitable factorised form with variable nodes in $P \cup J$ and factor nodes in $\{E_{ij}\}$. $P(\underline{n})$ is a function of E_{ij} which is itself a function of neighbors, variable nodes, of a factor node. From this we can construct the corresponding BP update rules for the assignment problem. They take the form:

$$(26) \quad v_{E_{ij} \rightarrow i}(n_{ij}) \cong \hat{v}_{j \rightarrow E_{ij}}(n_{ij}) e^{-\beta n_{ij}(E_{ij} - 2\gamma)}$$

$$(27) \quad \hat{v}_{j \rightarrow E_{ij}}(n_{ij}) \cong \sum_{\{n_{kj}\}} \mathbb{I} \left[n_{ij} + \sum_{k \in P \setminus i} n_{kj} \leq 1 \right] \prod_{k \in P \setminus i} v_{E_{kj} \rightarrow i}(n_{kj})$$

5.1. Dimer problems. The assignment problem is a fascinating example of a combinatorial optimisation problem one can investigate. If one considers the factor graph of the assignment problem, it is clear that the problem is in fact an example of a dimer model problem.

A dimer model can be defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. To begin, we first consider a single dimer on \mathcal{G} . A dimer is an edge e of \mathcal{G} whose end vertices have degree one; the vertices adjacent to each other via the dimer edge e are not connected to any other edges. A dimer covering \mathcal{D} can then be defined as a subgraph of \mathcal{G} given two constraints: the first is the vertex set of \mathcal{D} is the same set as the vertex set of \mathcal{G} and the second is the edge set of \mathcal{D} includes only dimer edges such that all vertices lie in a dimer. This is equivalent to requiring a perfect matching over \mathcal{G} . As there is no condition when it comes to selecting edges to be dimers in the dimer covering, most graphs will allow multiple unique dimer coverings.

6. Statistical Inference

True to its title, statistical inference is the discipline in which data is analysed and from that, properties of the underlying probability distribution are inferred. Formally, one can consider the problem in a different way. Given a set of variables $x_i \in \{x_i\}$ upon which we can perform measurements or observations, resulting in the data $\{y_j\}$, the aim of statistical inference is to obtain the best inference of the initial parameters $\{x_i\}$ as possible from the potentially noisy measured or observed data $\{y_j\}$. When faced with such problems, there are two questions one can ask. The first is what conditions does the data need to satisfy for a sufficient or satisfactory recovery of the initial conditions, such as is there enough data and is the quality of the data high enough? The second is similar to questions asked for previously discussed problems: can the inference of the measured or observed data be conducted in some algorithmic way so as to be performed computationally? We find in fact both questions are related to the type of transition that is exhibited, between different types of recovery of the initial conditions.

The problem can be reduced to that of a Bayesian inference problem. This utilises Bayes formula:

$$(28) \quad P(\underline{x}|\underline{y}) = \frac{P(\underline{y}|\underline{x})}{P(\underline{y})} \cdot P(\underline{x}) = \frac{P(\underline{y}|\underline{x})}{Z(\underline{y})} \cdot P(\underline{x})$$

$$(29) \quad \text{Posterior} = \frac{\text{Likelihood}}{\text{Evidence}} \cdot \text{Prior}$$

where we have renamed the evidence $P(\underline{y}) = Z(\underline{y})$. To ascertain an inference of the initial parameters, we must use either the posterior $P(\underline{x}|\underline{y})$ or a function of the posterior, called an estimator. An estimator is a 'distance' function one extremises over the set of possible initial parameters, the argument of the extremisation being the solution to the inference problem. This implies there is a non unique solution that exists for a given statistical problem, depending on the estimator used. Defining the initial parameters set as the vector \underline{x}^* and the optimal estimator as $\hat{\underline{x}}$, some examples of standard estimators are:

- MAP - Maximum A Posteriori - The function here is simply the posterior probability, implying:

$$(30) \quad \hat{\underline{x}}^{\text{MAP}} = \arg \max_{\underline{x}} P(\underline{x}|\underline{y})$$

- MMSE - Minimum Mean Squared Error - This estimator finds the minimum squared error between $\hat{\underline{x}}$ and \underline{x} averaged over the posterior distribution:

$$(31) \quad \hat{\underline{x}}^{\text{MMSE}} = \arg \max_{\underline{x}} \left\langle \frac{1}{N} \sum_{i=1}^N (\hat{x}_i - x_i)^2 \right\rangle_{P_{\underline{x}|\underline{y}}}$$

- MMO - Maximum Mean Overlap - In cases of discrete valued inference problems, this estimator finds the $\hat{\underline{x}}$ which maximises the element-wise agreement between the inferred solution and \underline{x} , averaged over the posterior distribution:

$$(32) \quad \hat{x}_i^{\text{MMO}} = \arg \max_{x_i} P_i(x_i|\underline{y}) = \arg \max_{x_i} \int \prod_{j \neq i} dx_j P(\underline{x}|\underline{y})$$

We can refine our set of statistical inference problems when we consider the specific subset, the Student/Teacher (S-T) scenario. To understand this scenario first let us begin with our student and teacher. The teacher has a statistical inference problem for the student and as a result creates a set of variables \underline{x}^* , the 'ground truth', from some probability distribution P_{prior} . The teacher uses these variable to generate a set of data \underline{y} by using the statistical model

characterised by the distribution P_{model} . They give this data to the student and the students task is to infer \underline{x}^* from the data \underline{y} along with partial or full knowledge of both P_{model} and P_{prior} .

The S-T scenario can be used to understand other models, one of which being the Planted ensemble model. In general, the two are mostly the same, the planted ensemble problem being a subset of problems in the S-T superset. A teacher sets an inference problem for their student, defines a realisation of variables \underline{x}^* called the 'planted configuration' and also generates the data set \underline{y} called the 'planted disorder', giving both to the student. The distinction between general S-T and ensemble problem comes with the definition of \underline{y} . In the S-T problem, there are no restrictions on how \underline{y} is defined. This allows for definitions of \underline{y} such that all elements y_i can be independently distributed random variables. This is not case in the planted ensemble problem and we find the components are all correlated. This subtlety can be visualised best using a common example, graph colouring.

Suppose we consider a teacher with a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ who wants to colour the vertices of the graph with a finite set of colours. They generate an initial colouring of the vertices using a distribution of the colours which results in the planted configuration \underline{x}^* . This configuration is then used to generate a second, uncoloured graph with the same vertices and an edge set which includes randomly selected edges of the coloured complete graph that only connect vertices with different assigned colours. These edges form the data set \underline{y} . We see in this example, the selection of edges which form the planted disorder, each edge represented by a discrete boolean variable, is not uncorrelated to others in the edge set; each is wholly dependent on the planted configuration.

7. The planted matching problem

Statistical inference problems such as the ones described previously can be easily considered on dimers and the matching problem. This problem is studied in [12]. The authors investigate the recovery of planted dimer configurations in weighted random graphs, exploiting the effect of using separate distributions for the planted and non-planted edges. For general distributions, the transition between partial and full recovery is then found and detail is given about specific cases of distributions.

7.1. Intro. The paper starts by outlining the case which is to be studied. Given a complete graph \mathcal{G}_0 , a perfect matching \mathcal{M} is selected uniformly at random from all possible matchings and set as the planted configuration. The planted configuration \mathcal{M} is used to define another graph \mathcal{G} , whose vertex set is equal to the vertex set of \mathcal{G}_0 but has edge set equal to \mathcal{M} . Additional edges, all elements of the edge set of \mathcal{G}_0 without \mathcal{M} , are added at random with probability $\frac{c}{N}$ to the edge set of \mathcal{G} . Having considered all edges of \mathcal{G}_0 not in \mathcal{M} , weights are then assigned to the edges of \mathcal{G} . Continuous distributions \hat{p} and p are used as weights for edges in the planted matching and those not in the planted matching respectively. Straight away it is possible to identify possible weights given the distributions \hat{p} and p . As weights are distributed to the edges from the supports of the two distributions, there exists the potential that, if the supports are not equal, there are edges with assigned weights that could only come from one of the two distributions. These edges can be identified immediately and removed from the problem. What is left then is a graph with weighted edges, each of those weights having the potential of being distributed by either \hat{p} or p as their weight lies in the intersection of the two supports, which is named Γ . By removing the known edges, those that are left are conditioned as to whether they are in Γ or not. The

distributions \hat{p} and p become:

$$(33) \quad P(\hat{w}) = \frac{1}{\hat{\mu}} \hat{p}(w) \mathbb{I}(w \in \Gamma), \quad \hat{\mu} = \int_{\Gamma} \hat{p}(w)$$

$$(34) \quad P(w) = \frac{1}{\mu} p(w) \mathbb{I}(w \in \Gamma), \quad \mu = \int_{\Gamma} p(w)$$

Each node thus has $1 + \mathbf{Z}$ edges, one which is an element of the planted matching and \mathbf{Z} other edges in the planted disorder where \mathbf{Z} can be shown to be a Poisson random variable with mean $\gamma := c\hat{\mu}\mu$. The inference to be studied is whether it is possible to recover the original matching \mathcal{M} with only the weighted graph \mathcal{G} , and how the parameters \hat{p} , p and c affect recovery. To recover the planted matching an estimator is required. The authors select two to study, the MAP estimator which they call the block MAP estimator and the MMO estimator, which they call the symbol MAP estimator. More specifically, the MMO estimator minimises the distance, which in this case is defined to be the number of nonequivalent element-wise entries, between the planted matching \mathcal{M} and the inferred matching $\hat{\mathcal{M}}$. This quantity is named the reconstruction error and given the symbol ξ . Throughout the paper they focus on the block MAP estimator, making final comments on how their results identify behaviour of the symbol MAP estimator as well.

7.2. Belief Propagation. To recover the planted matching from the graph, it is suggested the Belief-Propagation algorithm can be used to find it. This is argued by considering the probability of a given planted matching on the graph. Given the distribution of the matching over the graph, ideally we look to the marginals, that being the probability of a specific edge being in the matching or not. By then considering the limit of these marginals as $\beta \rightarrow \infty$, we see the probability concentrate on the minimal solution. In the case of the paper, this corresponds to the solution of the block MAP estimator. It is also noted that in the $\beta = 1$ situation, we extract the solution for the symbol MAP estimator.

In the case of the distribution, for all planted matchings, $\underline{m} = \{m_e \mid e \in \mathcal{E}\}$ where $m_e = 1 \leftrightarrow$ edge e is in the planted matching \mathcal{E} , the distribution can be calculated as:

$$(35) \quad v(\underline{m}) \propto e^{-\beta \sum_{e \in \mathcal{E}} m_e \omega_e} \prod_{i \in \mathcal{V}} \mathbb{I} \left(\sum_{e \in \partial i} m_e = 1 \right)$$

where the effective weight $\omega_e = \omega(w_e) = -\ln \frac{\hat{P}(w_e)}{P(w_e)}$ is introduced and the normalisation coefficient is neglected. It is proposed the analytic computation of the marginals of this distribution is intractable, suggesting instead an approximate solution is more feasible, one in the form of the Belief-Propagation equations. The messages $v_{i \rightarrow e}(m)$ can be calculated and reparameterised by a new variable $h_{i \rightarrow e}$, resulting in $v_{i \rightarrow e} = v_{i \rightarrow e}(m, h_{i \rightarrow e})$. The cavity fields are defined for ease of calculation and are related to the BP messages through the equation:

$$(36) \quad v_{i \rightarrow e}(m) := \frac{e^{\beta m h_{i \rightarrow e}}}{1 + e^{\beta h_{i \rightarrow e}}}$$

The BP messages can then be rearranged into an iterative equation with $h_{i \rightarrow e}$ as its subject. The marginal of each edge e can be considered in terms of $h_{i \rightarrow e}$ as well but it is worth noting, just like the BP equations, the cavity field equations are functions of edges not including the edge to be calculated. Having reparameterised the marginal the inclusion rule is defined. If the marginal of an edge indicates there is a higher probability of that edge being in the matching than not being in the matching, it is inferred it was an element of \mathcal{M}

$$(37) \quad \hat{\mathcal{M}}_s = \left\{ e \in \mathcal{E} : v_e(1) > \frac{1}{2} \right\}$$

The inference solution $\hat{\mathcal{M}}_s$ can only be considered once the BP equations have reached some threshold of convergence to a fixed point of the equations.

7.3. Recursive Distributional Equations. While this is a valid method for inferring the matching for a given instance of the problem, one example is insufficient to gain insight into the behaviour of the system in general. For that we need to consider the 'average' behaviour exhibited. To this end the authors generalise the process, replacing realisations of randomly distributed variables in equations with a variable distributed accordingly. They start by defining two new random variables to study which describe the behaviour of the cavity equations in a general instance along with another two random variables to assist this. These complementary random variables are \hat{W} and W , which are distributed according to \hat{P} and P respectively and act as generic substitutes for the weights of an instance of the weighted graph. These are used in conjunction with the equation for ω to define $\hat{\Omega} = \omega(\hat{W})$ and $\Omega = \omega(W)$. Considering $h_{i \rightarrow e}$ as a random variable, $\hat{\mathbf{H}}$ and \mathbf{H} are introduced to describe the distributions of the cavity field random variables for edges in the planted matching and the non planted matching respectively. For any given instance of the graph, if edge $i \rightarrow e$ is a planted edge, i is incident to \mathbf{Z} non planted edges and if edge $i \rightarrow e$ is a non-planted edge, i is incident to a planted edge and \mathbf{Z} non-planted edges. Formally, these read as follows:

$$(38) \quad \hat{\mathbf{H}} \sim -\frac{1}{\beta} \ln \left(\sum_{i=1}^{\mathbf{Z}} e^{-\beta(\Omega_i - \mathbf{H}_i)} \right)$$

$$(39) \quad \mathbf{H} \sim -\frac{1}{\beta} \ln \left(e^{-\beta(\hat{\Omega}_i - \hat{\mathbf{H}}_i)} + \sum_{i=1}^{\mathbf{Z}} e^{-\beta(\Omega_i - \mathbf{H}_i)} \right)$$

In this new random variable setting, the inclusion rule can be interpreted as:

$$(40) \quad \mathbb{E}[\xi] = \frac{\hat{\mu}}{2} \mathbb{P}[\hat{\mathbf{H}} + \hat{\mathbf{H}}' \leq \hat{\Omega}] + \frac{\hat{\mu}\gamma}{2} \mathbb{P}[\hat{\mathbf{H}} + \hat{\mathbf{H}}' > \Omega]$$

What the authors find at this point is a second selection of edges can be made that are known for sure to be in the planted matching \mathcal{M} . If one considers a leaf on the graph, it must be the case this leaf edge is in the matching \mathcal{M} , otherwise the node at the end would be unmatched. The distributional equations reflect this property. If an edge e is a leaf, there exists one node of the leaf, i with no neighboring edges. We can substitute this example case into the equation for $\hat{\mathbf{H}}$ and we are left with:

$$(41) \quad \hat{\mathbf{H}} \sim -\frac{1}{\beta} \ln \left(\sum_{i=1}^0 e^{-\beta(\Omega_i - \mathbf{H}_i)} \right) \sim -\frac{1}{\beta} \ln(0) \sim \infty$$

The existence of leaves on a graph is an event which occurs with a non-zero probability, which can be described as having the probability \hat{q} if the leaf is in the matching \mathcal{M} and q if it is not in \mathcal{M} . As a result, the recursive distributional equations can be rewritten as:

$$(42) \quad \hat{\mathbf{H}} \sim \begin{cases} \infty, & \text{with probability } \hat{q} \\ \hat{H}, & \text{with probability } 1 - \hat{q} \end{cases}, \quad \mathbf{H} \sim \begin{cases} -\infty, & \text{with probability } q \\ H, & \text{with probability } 1 - q \end{cases}$$

where it can be shown $\hat{q} = q$ and must satisfy the relation $q = e^{-\gamma(1-q)}$ and \hat{H} and H are finite with probability 1. The authors make a note stating they exclude the event of $\hat{\mathbf{H}} \sim -\infty$ and $\mathbf{H} \sim \infty$

Substituting these into the equations for $\hat{\mathbf{H}}$ and \mathbf{H} , we obtain the equations satisfied by \hat{H} and H . It is shown:

$$(43) \quad \hat{H} \sim -\frac{1}{\beta} \ln \left(\sum_{i=1}^Z e^{-\beta(\Omega_i - H_i)} \right), \quad H \sim \begin{cases} \hat{\Omega} - \hat{H}, & \text{with probability } q \\ -\frac{1}{\beta} \ln \left(e^{-\beta(\hat{\Omega} - \hat{H})} + e^{-\beta\hat{H}'} \right), & \text{with probability } 1-q \end{cases}$$

where Z is a Poisson random variable with parameter $\gamma(1-q)$. The average reconstruction error is then re-evaluated to be:

$$(44) \quad \mathbb{E}[\xi] = \frac{\hat{\mu}(1-q)^2}{2} \mathbb{P}[\hat{H} + \hat{H}' \leq \hat{\Omega}] + \frac{\hat{\mu}(1-q)^2 \gamma}{2} \mathbb{P}[H + H' > \Omega]$$

7.4. Transition location for block MAP. To estimate the block MAP solution, the authors revisit the significance of the $\beta \rightarrow \infty$ limit. In particular, the Recursive Distributional Equations can be simplified in this limit:

$$(45) \quad \hat{H} \sim \min_{1 \leq i \leq Z} (\Omega_i - H_i), \quad H \sim \begin{cases} \hat{\Omega} - \hat{H}, & \text{with probability } q \\ \min(\hat{\Omega} - \hat{H}, \hat{H}'), & \text{with probability } 1-q \end{cases}$$

as well as the average reconstruction error:

$$(46) \quad \mathbb{E}[\xi] = \hat{\mu}(1-q)^2 \mathbb{P}[\hat{H} + \hat{H}' \leq \hat{\Omega}]$$

The authors now search for a continuous phase transition with respect to some unknown variable λ , upon which the average reconstruction error is dependant $\mathbb{E} = \mathbb{E}[\xi](\lambda)$. They propose there exists a $\bar{\lambda}$ such that for $\lambda < \bar{\lambda}$, $\mathbb{E}[\xi](\lambda) > 0$ indicating the system is in a state of partial recovery, whereas for $\lambda < \bar{\lambda}$, $\mathbb{E}[\xi](\lambda) = 0$ representing the state of full recovery. If this is the case though, the revised average reconstruction error equation implies as $\mathbb{E}[\xi] \rightarrow 0$, given the other coefficients are non-zero, $\mathbb{P}[\hat{H} + \hat{H}' \leq \hat{\Omega}] \rightarrow 0$, implying $\hat{H} + \hat{H}' > \hat{\Omega}$ for all $\hat{\Omega}$. As $\hat{\Omega}$ is a random variable, this is only possible if $\hat{H} \rightarrow \infty$. Knowing this we see the equations \hat{H} and H must behave as:

$$(47) \quad \hat{H} \sim \min_{1 \leq i \leq Z} [\Omega_i - H_i], \quad H \sim \hat{\Omega} - \hat{H}$$

If we define a new random variable $\Xi \sim \Omega - \hat{\Omega}$ we find the distributional equations can be simplified into a single equation for \hat{H} :

$$(48) \quad \hat{H} \sim \min_{1 \leq i \leq Z} [\Xi_i - \hat{H}_i]$$

The form the equation has been manipulated into has significant importance. The authors discovered a fascinating connection with this equation form and properties of a process known as Branching Random Walks (BRW).

Branching Random Walk processes describe the progression along a single spatial axis of an increasing population of points at discrete time intervals. More formally, let us define our population of points as $\{X_k^n\}_k$ where n indicates the interval of time and k represents the number of particles at time interval n . For convenience the Real axis is set as the spatial axis and the initial conditions for the process at $n = 0$ are $k = 1$, resulting in an initial population set of $\{X_1^0\}$, starting at the origin. At each successive time interval, from n to $n + 1$, each parent particle X_i^n in the population is assigned a number of offspring particles Z_i in the $n + 1$ generation, Z_i being a random variable, independent and identically distributed for each parent, with each offspring assigned also a displacement, Ξ_j , which is also an i.i.d. random variable. For a given parent particle, the set of new offspring particles are then located at $\{X_{i,j}\}_j = X_i + \Xi_j$. In the transition from time interval n to $n + 1$, the population of generation n 'dies', leaving only the new offspring in generation $n + 1$. The property of this process highlighted by the authors to be

of significance is the progression of the point on the axis with the minimum location in the large n limit.

More specifically, the minimum point of each generation can be labelled; $\hat{K}^n = \min_i[X_i^n]$. After each time interval, by definition of \hat{K}^n , it must obey the recursive distributional equation:

$$(49) \quad \hat{K}^{n+1} \sim \min_{1 \leq i \leq Z} [\Xi_i - \hat{H}_i]$$

This type of process has been studied before, allowing the authors to draw on previous knowledge. As a result it is known that, provided the process does not reach extinction, $\hat{K}^n = -\infty$ in finite time, $\frac{\hat{K}^n}{n} \rightarrow v$ as $n \rightarrow \infty$ with almost sure convergence. v is a constant 'velocity' of the process and can be calculated as:

$$(50) \quad v = -\inf_{\theta > 0} \frac{1}{\theta} \ln (\mathbb{E}[Z] \mathbb{E}[e^{-\theta \Xi}])$$

Another important property of the random variable \hat{K}^n , is it follows the following limit:

$$(51) \quad \hat{K}^n - nv - C \log n \rightarrow L, \quad n \rightarrow \infty$$

where C is a constant; it is noted this behaviour is satisfied only under specific 'technical conditions'. The equation converges to the finite random variable L , the solution satisfying:

$$(52) \quad L \sim -v + \min_{1 \leq i \leq Z} [\Xi_i - L_i]$$

An important note is made at this point explaining that the solution L is translation invariant. This implies if L is a solution of this distributional equation, so too is $L + \alpha$ for any constant α . It is then argued, if the problem is considered such that $v = 0$, we see $\hat{K}^n - C \log n \rightarrow L \sim \min_{1 \leq i \leq Z} [\Xi_i - L_i]$. In this specific case, not only does L become a solution of the BRW recursive distributional equation for \hat{K}^n , we also see as $n \rightarrow \infty$, $\hat{K}^n \rightarrow C \log n + L$. As it is made clear L is translation invariant, this implies $\hat{K}^n \rightarrow C \log n + L \sim L$.

Returning to the planted matching problem at hand, it is outlined in the paper it is possible to prove the new random variable \hat{K}^n defined above is stochastically smaller than the previously defined \hat{H} , which describes the finite cavity field distribution. As both \hat{K}^n and \hat{H} are distributionally defined using equivalent parameters, we can rewrite the velocity as:

$$(53) \quad v = -\inf_{\theta > 0} \frac{1}{\theta} \ln [I(\theta)I(1-\theta)], \quad I(\theta) = \sqrt{\gamma} \int_{\Gamma} \hat{P}(w)^{\theta} P(w)^{1-\theta} dw$$

As explained above, in the large generation limit of the process \hat{K}^n , when $v = 0$ we see \hat{K}^n tend to a nontrivial solution. This is the condition the authors state must be satisfied at the boundary between partial and full recovery.

The infimum of $\ln [I(\theta)I(1-\theta)]$ is found to be the point $I(\frac{1}{2}) = 1$, resulting in the final form:

$$(54) \quad \int_{\Gamma} \Gamma \sqrt{\hat{p}(w)p(w)} dw = \frac{1}{\sqrt{c}}$$

where \hat{p}, p and c are the original parameters of the problem.

The paper goes on to make use of the results that have come before this and provide examples of a few planted matching problems as well as the results they were able to calculate along with some conclusive remarks.

CHAPTER 2

Inferring planted matchings on lattices

Influenced by the results of [12], the aim of my project expands on the papers previous work, focusing on more explicit examples and their properties. *We look to study the behaviour of the transition between partial and full recovery of a planted perfect matching in different graph topologies subject to various constraints.*

1. Establishing inference in Graphs

We will consider three undirected graph types in the following analysis. They are as follows:

- $\mathcal{G}_C(n)$ = Complete graph of n vertices
- $\mathcal{G}_R(n, c)$ = Random Regular graph of n vertices. Each vertex has coordination c or equivalently degree $c + 1$, connected uniformly at random to the other $n - 1$ vertices.
- $\mathcal{G}_S(\sqrt{n})$ = Finite square lattice of size $\sqrt{n} \times \sqrt{n}$

To outline the inference problem, let us start with an instance of an undirected, unweighted graph \mathcal{G}_0 - the use of the non specific graph notation \mathcal{G} indicates the same procedure is carried out regardless of which graph structure is initially chosen. The procedure is as follows:

- In \mathcal{G}_0 , let us define a perfect matching \mathcal{M} . Of all possible perfect matchings \mathcal{M} is arbitrarily chosen and selected as our planted matching.
- With \mathcal{M} chosen, $\hat{\rho}, \rho \in [0, 1]$ are defined to be the fraction of planted matching edges and non-planted matching edges 'removed' respectively. One can imagine this to be similar to loss of information during signal transfer. Removed edges are given an uninformative symbol \star .
- The remaining edges, those in the matching \mathcal{M} and those not in \mathcal{M} are assigned separate random variable weights. The weight of edge \hat{e} in the planted matching, $w_{\hat{e}} \sim \hat{W}$, is exponentially distributed with rate parameter λ . The weight of edge e not in the planted matching, $w_e \sim W$, is exponentially distributed with rate parameter $\frac{1}{c}$, c being the coordination. One should note all edges are independently distributed. More formally, $\hat{W} \sim \text{Exp}(\lambda)$ with probability distribution \hat{P} and $W \sim \text{Exp}(\frac{1}{c})$ with distribution P . The weighted graph is now labelled \mathcal{G} .

In the language of statistical inference, more specifically the Student-Teacher scenario, what we have done by employing this procedure is create a planted configuration, the planted matching \mathcal{M} , and generate a data set, that being the set of weighted edges of the graph \mathcal{G} , along with information about the distributions of edge weights, upon which inference can be conducted.

Our objective is to study how the success of recovery of \mathcal{M} behaves on average for each graph structure. We shall indicate the recovered matching as $\hat{\mathcal{M}}$ and our statistical inference will be performed using only the restricted knowledge of the weighted graph \mathcal{G} , the distribution of planted and non-planted edges \hat{P} and P and the probability of edge removal $\hat{\rho}$ and ρ . Success of our statistical inference can be quantified by considering the proportion of edges incorrectly inferred to be in the planted matching. More formally, we look to calculate the symmetric difference of $\hat{\mathcal{M}}$ and \mathcal{M} , scaled by a factor $\frac{1}{N}$ to allow for comparison of error between different

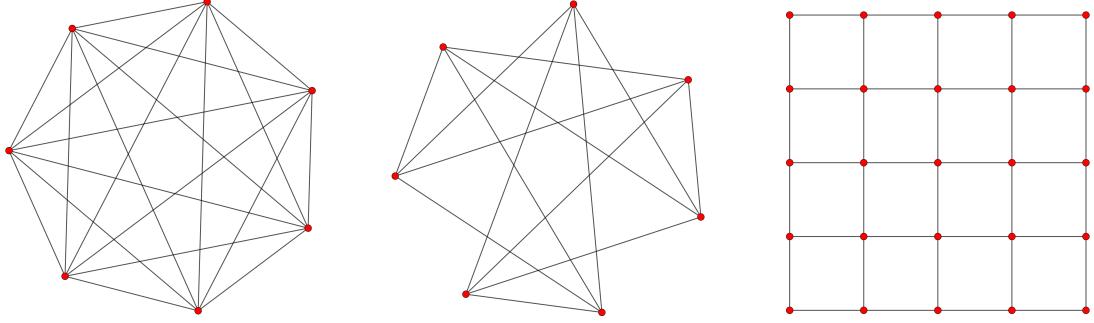


FIGURE 1. From left to right: Complete graph of $N = 7$, Random Regular graph of $N = 7, c = 4$, Square lattice of $N = 5 \times 5 = 25$

size graphs. Thus, the error in our recovery can be calculated as:

$$(55) \quad E = |\mathcal{M} \Delta \hat{\mathcal{M}}|$$

2. Boundary of phase transition

There are two states the system could exist in as a result of initial parameter choices: a state in which the error, E , is non-zero and a state in which the error is zero. We shall call these two phases, partial recovery and full recovery. It is possible, for the random regular graph, to calculate the transition point as a function of $\hat{\rho}, \rho, c$. To do this we follow similar steps as in [12]. First we must define the probability of a matching over the random regular graph. We define matchings, \underline{m} to be distributed with the following function:

$$(56) \quad P(\underline{m}) = \frac{1}{Z} e^{-\beta \sum_{e \in \mathcal{E}} m_e \omega_e} \prod_{i \in \mathcal{V}} \mathbb{I} \left(\sum_{e \in \partial i} m_e = 1 \right)$$

where we have included the effective weight $\omega_e = \omega(w_e) = -\ln \frac{\hat{P}(w_e)}{P(w_e)}$. From this we can use the formulation of belief propagation to calculate the BP messages. We know this is possible as the structure of the problem - the random regular graph \mathcal{G} , can be formulated into a factor graph \mathcal{F} . If we set variable nodes in \mathcal{F} to be the edges of \mathcal{G} and factor nodes of \mathcal{F} to be the vertices of \mathcal{G} we form a suitable bipartite graph - in any graph, edges only connect to vertices, and visa versa. For vertex i and edge e the messages have the form:

$$(57) \quad v_{i \rightarrow e}(m_e) \propto \sum_{\{m_{e'}\}_{e' \in \partial i \setminus e}} \mathbb{I} \left(m_e + \sum_{e' \in \partial i \setminus e} m_{e'} = 1 \right) \prod_{\substack{e'=(r,i) \\ e' \in \partial i \setminus e}} v_{r \rightarrow e'}(m_{e'}) e^{-\beta m_{e'} \omega_{e'}}$$

We look to write these message equations in a more versatile form, prompting the parameterisation of the marginal message $v_{i \rightarrow e}(m)$ in terms of cavity fields $h_{i \rightarrow e}$ which are related to

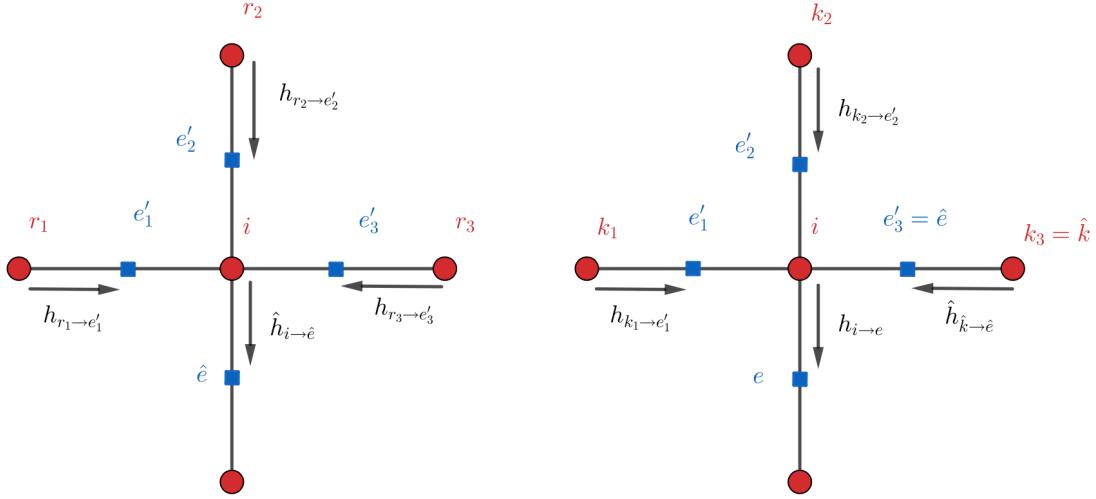


FIGURE 2. Decomposition of $\hat{h}_{i \rightarrow \hat{e}}$ and $h_{i \rightarrow e}$ in terms of other cavity fields for $c = 3$. We see only non planted matching cavity fields contribute to the cavity field for the planted matching $\hat{h}_{i \rightarrow \hat{e}}$. However a single planted matching cavity field contributes, along with the $c - 1$ non-planted matching cavity fields, to the non-planted cavity field $h_{i \rightarrow e}$

the message marginals via the normalised distribution for $m_e \in \{0, 1\}$:

$$(58) \quad v_{i \rightarrow e}(m_e) := \frac{e^{\beta m_e h_{i \rightarrow e}}}{1 + e^{\beta h_{i \rightarrow e}}} \Rightarrow h_{i \rightarrow e} = -\frac{1}{\beta} \ln \left(\sum_{\substack{e'=(r,i) \\ e' \in \partial i \setminus e}} e^{-\beta(\omega_{e'} - h_{r \rightarrow e'})} \right)$$

We can refine our definition of $h_{i \rightarrow e}$ by considering the case when e is a planted edge and when it is not a planted edge. Let us define $\hat{h}_{i \rightarrow \hat{e}}$ to be the cavity field of a planted edge \hat{e} and $h_{i \rightarrow e}$, the cavity field of a non-planted edge e . By using the topology of the random regular graph, we see $\hat{h}_{i \rightarrow \hat{e}}$, $h_{i \rightarrow e}$ take the values:

$$(59) \quad \hat{h}_{i \rightarrow \hat{e}} = -\frac{1}{\beta} \ln \left(\sum_{j=1}^c e^{-\beta(\omega_{e'_j} - h_{r_j \rightarrow e'_j})} \right)$$

$$(60) \quad h_{i \rightarrow e} = -\frac{1}{\beta} \ln \left(e^{-\beta(\omega_{\hat{e}} - h_{\hat{k} \rightarrow \hat{e}})} + \sum_{k=1, k \neq \hat{k}}^c e^{-\beta(\omega_{e'_k} - h_{r_k \rightarrow e'_k})} \right)$$

where we have assumed $e'_j = (r_j, i) \in \partial i \setminus \hat{e}$, $e'_k = (r_k, i) \in \partial i \setminus e$ for all j, k and $e'_{\hat{k}} = (\hat{k}, i) = \hat{e}$.

We can now define the inclusion rule, that being the condition edges must satisfy in order to be considered part of the recovered matching. This rule is simply whether the probability the edge is in the matching is higher than the probability it is not. This can be written as:

$$(61) \quad \hat{\mathcal{M}} = \{e \in \mathcal{E} \mid v_e(m_e = 1) > \frac{1}{2}\}$$

We have defined v_e to be the marginal of edge e connecting vertex i to j , and in a similar way as before, we can define v_e using the convenient parameterisation:

$$(62) \quad v_e(m_e) := \frac{e^{\beta m_e(h_{i \rightarrow e} + h_{j \rightarrow e} - \omega_e)}}{1 + e^{\beta(h_{i \rightarrow e} + h_{j \rightarrow e} - \omega_e)}}$$

It then follows, the inclusion rule must be defined as:

$$(63) \quad \hat{\mathcal{M}} = \{e \in \mathcal{E} \mid h_{i \rightarrow e} + h_{j \rightarrow e} > \omega_e\}$$

These equations work only for the single instance of a generated graph. To study the behaviour over infinite instances, we can define new random variables, \hat{H}, H which correspond to the planted and non-planted edges respectively and describe the cavity fields. They are governed by the cavity field equations themselves and have the form:

$$(64) \quad \hat{H} \sim -\frac{1}{\beta} \ln \left(\sum_{i=1}^c e^{-\beta(\Omega_i - H_i)} \right)$$

$$(65) \quad H \sim -\frac{1}{\beta} \ln \left(e^{-\beta(\hat{\Omega}_i - \hat{H}_i)} + \sum_{i=1}^{c-1} e^{-\beta(\Omega_i - H_i)} \right)$$

We need to consider the inclusion rule to progress further with the derivation. Recall the error was the proportion of incorrectly classified edges against the size of the matching. In other words, this is the fraction of incorrect edges in \mathcal{M} plus the fraction of correct edges not in \mathcal{M} . Using the inclusion rule, we can calculate the error E to be:

$$(66) \quad \mathbb{E}[\xi] = \frac{1}{2} \mathbb{P}[\hat{H} + \hat{H}' \leq \hat{\Omega}] + \frac{c}{2} \mathbb{P}[H + H' > \Omega]$$

As we are looking for the MAP estimator, we need to consider these equations in the limit as $\beta \rightarrow \infty$. In this limit, we see \hat{H} concentrate on the minimal exponent of $\Omega_i - H_i$ and H concentrates on the minimal exponent of $\Omega_i - H_i$ compared with $\hat{\Omega}_i - \hat{H}_i$. More formally, what happens is:

$$(67) \quad \hat{H} \sim \min_{1 \leq i \leq c} (\Omega_i - H_i), \quad H \sim \min(\hat{\Omega} - \hat{H}, \min_{1 \leq i \leq c-1} (\Omega_i - H_i))$$

The inclusion rule can also be simplified in the large β limit, and we are left with the resulting condition:

$$(68) \quad \mathbb{E}[\xi] = \mathbb{P}[\hat{H} + \hat{H}' \leq \hat{\Omega}]$$

What we see at this point is that for $\mathbb{E}[\xi] = 0 \Rightarrow \mathbb{P}[\hat{H} + \hat{H}' \leq \hat{\Omega}] = 0$, it must be the case that $\hat{H} + \hat{H}' > \hat{\Omega}$. This implies $\hat{H} \rightarrow \infty$ at the transition to zero error. This affects the distributions of \hat{H} and H , resulting in the new distributions:

$$(69) \quad \hat{H} \sim \min_{1 \leq i \leq c} (\Omega_i - H_i), \quad H \sim \hat{\Omega} - \hat{H}$$

The same application of the BRW can be made at this point, to these equations, resulting in the condition:

$$(70) \quad \int_{\Gamma} \sqrt{\hat{p}(w)p(w)} dw = \frac{1}{\sqrt{c}}$$

For distributions \hat{p} and p we are considering the case when there exists potential edge removal from the graph, edge weights being replaced by an uninformative symbol \star , with probability \hat{p}

and ρ respectively. Otherwise the distributions are exponential with separate variables. They can be written in the following way:

$$(71) \quad \hat{p}(w) = (1 - \hat{\rho})\hat{P}(w) + \hat{\rho}\delta(w - \star) = (1 - \hat{\rho})\lambda e^{-\lambda w} \mathbb{I}(w \in \mathbb{R}_{\geq 0}) + \hat{\rho}\delta(w - \star)$$

$$(72) \quad p(w) = (1 - \rho)P(w) + \rho\delta(w - \star) = (1 - \rho)\frac{1}{c}e^{-\frac{1}{c}w} \mathbb{I}(w \in \mathbb{R}_{\geq 0}) + \rho\delta(w - \star)$$

The support of each probability distribution is the same, meaning $\Gamma = \{\star\} \cup \mathbb{R}_{\geq 0}$. If we substitute just the range of integration into the given condition we find:

$$(73) \quad \frac{1}{\sqrt{c}} = \int_{\Gamma} \sqrt{\hat{p}(w)p(w)} dw$$

$$(74) \quad = \int_{\{\star\} \cup \mathbb{R}_{\geq 0}} \sqrt{\hat{p}(w)p(w)} dw$$

$$(75) \quad = \int_{\{\star\}} \sqrt{\hat{p}(w)p(w)} dw + \int_{\mathbb{R}_{\geq 0}} \sqrt{\hat{p}(w)p(w)} dw$$

$$(76) \quad = I_{\{\star\}} + I_{\mathbb{R}_{\geq 0}}$$

We can calculate each integral separately, taking the sum of the two to find the solution. Let us consider the first integral, $I_{\{\star\}}$. Over just the set $\{\star\}$, we find both distributions, \hat{p} and p , simplify. Without loss of generality, by considering \hat{p} over the domain $\{\star\}$, the $(1 - \rho)\frac{1}{c}e^{-\frac{1}{c}w} \mathbb{I}(w \in \mathbb{R}_{\geq 0})$ term equals 0 as $\mathbb{I}(\star \in \mathbb{R}_{\geq 0}) = 0$. So the result is just the delta term $\hat{p}(w) = \hat{\rho}\delta(w - \star)$. The same can be said for the distribution p , leaving the integral in the simplified form:

$$(77) \quad I_{\{\star\}} = \int_{\{\star\}} \sqrt{\hat{\rho}\delta(w - \star) \cdot \rho\delta(w - \star)} dw$$

$$(78) \quad = \int_{\{\star\}} \sqrt{\hat{\rho}\rho} \sqrt{\delta(w - \star) \cdot \delta(w - \star)} dw = \sqrt{\hat{\rho}\rho} \int_{\{\star\}} \sqrt{\delta(w - \star) \cdot \delta(w - \star)} dw$$

$$(79) \quad = \sqrt{\hat{\rho}\rho} \int_{\{\star\}} \delta(w - \star) dw$$

$$(80) \quad = \sqrt{\hat{\rho}\rho}$$

The second integral, $I_{\mathbb{R}_{\geq 0}}$, can be solved in a similar manner. For all $w \in \mathbb{R}_{\geq 0}$, $\delta(w - \star) = 0$ and $\mathbb{I}(w \in \mathbb{R}_{\geq 0}) = 1$ so both distributions \hat{p} and p are reduced to $(1 - \hat{\rho})\lambda e^{-\lambda w}$ and $(1 - \rho)\frac{1}{c}e^{-\frac{1}{c}w}$ respectively. The integral can then be solved by first substituting:

$$(81) \quad I_{\mathbb{R}_{\geq 0}} = \int_{\mathbb{R}_{\geq 0}} \sqrt{(1 - \hat{\rho})\lambda e^{-\lambda w} \cdot (1 - \rho)\frac{1}{c}e^{-\frac{1}{c}w}} dw$$

$$(82) \quad = \int_{\mathbb{R}_{\geq 0}} \sqrt{(1 - \hat{\rho})(1 - \rho)} \sqrt{\frac{\lambda}{c}} \sqrt{e^{-\lambda w} \cdot e^{-\frac{1}{c}w}} dw$$

$$(83) \quad = \sqrt{\frac{\lambda}{c}} \sqrt{(1 - \hat{\rho})(1 - \rho)} \int_{\mathbb{R}_{\geq 0}} e^{-\frac{\lambda+c-1}{2}w} dw$$

$$(84) \quad = \sqrt{\frac{\lambda}{c}} \sqrt{(1 - \hat{\rho})(1 - \rho)} \left[\frac{e^{-\frac{\lambda+c-1}{2}w}}{-\frac{\lambda+c-1}{2}} \right]_0^\infty$$

$$(85) \quad = \sqrt{\frac{\lambda}{c}} \sqrt{(1 - \hat{\rho})(1 - \rho)} \left(\frac{2}{\lambda + \frac{1}{c}} \right)$$

Combining the two integrals, we see the boundary is satisfied by the curve:

$$(86) \quad \frac{1}{\sqrt{c}} = \sqrt{\hat{\rho}\rho} + 2\sqrt{c(1-\hat{\rho})(1-\rho)} \frac{\sqrt{\lambda}}{\lambda c + 1}$$

What we can initially note about this function is that there is no distinction made between the values $\hat{\rho}$ and ρ . This implies, regardless of which edge set is suppressed, the transition will occur at the same point. Our statistical inference relies prior knowledge of $\hat{\rho}$, ρ and c . This allows us to estimate the regions of partial and full recovery before they are numerically investigated. Our numerical investigations will be limited to estimations as there may be noticeable finite size error in the results.

3. Numerical Investigation

Given some instance of matching recovery on graph \mathcal{G} , as established above, it is possible to calculate the success of the statistical inference in terms of the error E . This calculation though is limited to the success of recovery given the specific realisation of the random variables that were present in the given instance. In order to study the behaviour exhibited by the structure of graph \mathcal{G} , it is necessary to understand how the error behaves over multiple iterations of the process. With no prior knowledge of the planted matching \mathcal{M} , a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ is given along with the distributions \hat{P} and P of planted and non planted edge weights; the aim being to identify the planted matching \mathcal{M} . The numerical analysis was conducted by solving instances of the graph for given initial parameters and averaging the error of the matching over all iterations.

3.1. Numerical Discrepancy. In the numerical analysis we expect there to be a discrepancy in our calculation between the error E of the square lattice and the random regular lattice. There are three main causes of this. First is the finite size effect each recovery process feels. This has contribution of order $\frac{1}{N}$ and for small sized graphs such as $N = 64, 100$, it is more likely this affect has not become negligible. The second disparity is the boundary error on the square lattice. While the two graph topologies are similar, depending on the boundary conditions imposed, the square lattice is not always an instance of the random regular lattice. In this analysis, closed boundary conditions were implemented, resulting in the boundary vertices of the lattice having coordination $c \neq 3$. As a result of not all vertices having coordination 3, the graph structures were not exactly the same. The final difference would be as a result of the size of loops present in the graph. We know as the size of a random regular graph of coordination $c = 3$ increases, locally the graph behaves as a tree and cycles become longer. In a lattice this is not the case. For whatever N , there exist multiple cycles of length 4 for every node. The disparity between the curves in the E/λ plots will be mostly due to this.

3.2. Numerical analysis without suppression ($\hat{\rho} = \rho = 0$). We shall first consider recovery properties of the random regular graph $\mathcal{G}_R(n, c)$ with coordination c , along with the behaviour exhibited as $c \rightarrow n$, this limiting case corresponding to the complete graph $\mathcal{G}_C(n)$.

By plotting the error of recovery for random regular graphs of low values of coordination along with the limiting coordination case, we see the transition point between partial and full recovery is similar for all coordination c . This is to be expected and we can see this from the

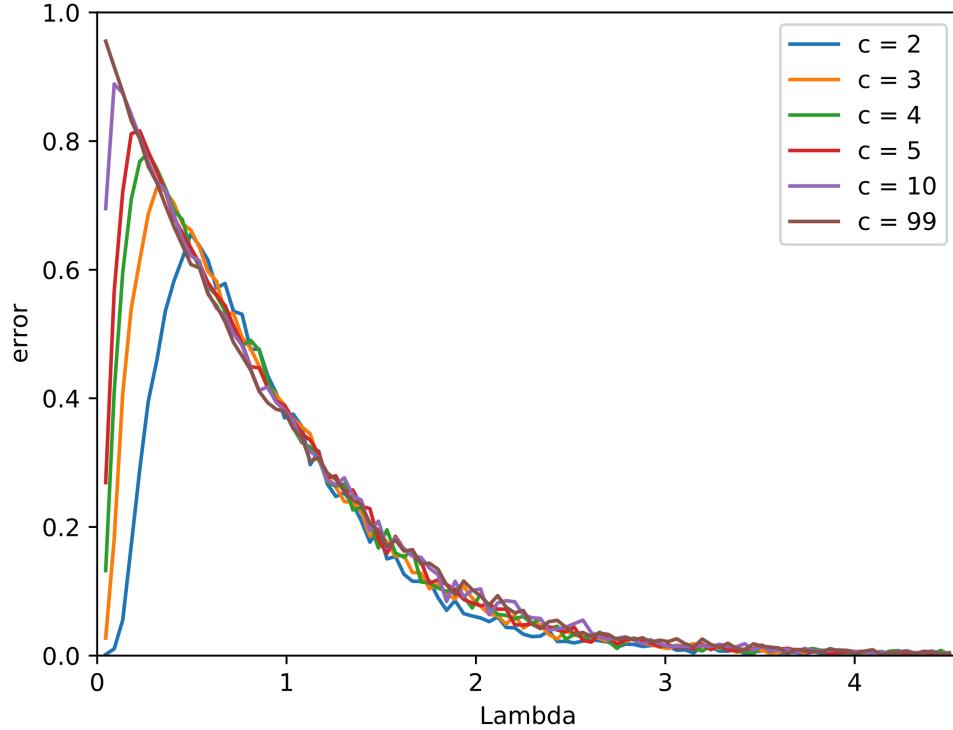


FIGURE 3. Reconstruction error for random regular graph of size $n = 100$ without suppression, $\hat{\rho} = \rho = 0$. Coordination varies: $c \in \{2, 3, 4, 5, 10, \infty\}$ such that $c \rightarrow \infty$ corresponds to the complete graph. Each curve represents a locus of points, each point being averaged over 100 instances.

boundary equation. At $\hat{\rho} = \rho = 0$ the boundary equation simplifies to give:

$$(87) \quad \frac{1}{\sqrt{c}} = 2\sqrt{c} \frac{\sqrt{\lambda}}{\lambda c + 1}$$

$$(88) \quad (\lambda c + 1)^2 = 4c^2 \lambda$$

$$(89) \quad 0 = c^2 \lambda^2 + (4c^2 - 2c)\lambda - 1$$

Considering this equation, we see there are 3 solution sets: 0, 1 or 2 solutions. At zero solutions, there is no partial recovery phase, so we expect the matching to always be recoverable in limiting size random regular graphs. For 2 solutions, (λ_1, λ_2) such that $\lambda_1 < \lambda_2$, we expect to see zero error for $\lambda < \lambda_1$ and $\lambda > \lambda_2$ along with non-zero error in the region $[\lambda_1, \lambda_2]$. Finally for the one solution case, there would be a single point which forces partial recovery of the matching, while elsewhere is a region of full recovery. Table 1 records the expected transition for each case plotted in figure 3. Considering the upper bound of the partial recovery phase first, the graph is representative of what we expect to see; the error transitions from a region where $E > 0$ into a region where $E \approx 0$. We note it is not exactly zero, this being most likely due to finite size errors. For the lower bound of the partial recovery phase a log scale is required to better understand the result.

c	λ_1	λ_2
2	0.0858	2.9142
3	0.0337	3.2997
4	0.0179	3.4821
5	0.0111	3.5889
10	0.00263	3.7974
∞	0	4

TABLE 1. Values of the transition point between partial recovery and full recovery of a hidden matching for a random regular graph of coordination c .

We can now consider the recovery error over the square lattice. Of the various graph topologies seen already, the square lattice has structure most similar to the random regular graph of coordination $c = 3$. As a result, the E/λ curve for the square lattice of size N will be plotted against the curve of the random regular graph of $c = 3$. This was tested for $N = 64, 100$ and in both cases, the error for the square lattice follows the shape of the error for the random regular graph closely, suggesting the transition points are close to one another. The similarity between the two curves could be indicative of a deeper connection between the two graph topologies. To understand further, we can next investigate how the suppression of edges affects the curve shapes.

3.3. Numerical analysis with suppression ($\hat{\rho}, \rho \neq 0$). Having investigated the graphs with no suppression, we can look to include suppression, remove random edges, and see what affect this has. For the random regular graph, to find the transition points, we need look no further than the boundary equation. Using this we can estimate the location of the transition points. For the performance of the square lattice, just as before, we can compare its error/ λ curve behaviour to that of the random regular graph of coordination $c = 3$ as these share a similar topology.

We have plotted the boundary equation for $c = 3$ and $\hat{\rho} = \rho$ in 5 to allow us to highlight whether the curve for the square lattice behaves as the curve for the random regular graph. An interesting note at this point is the asymptote, occurring at $\hat{\rho} = \rho = \frac{1}{\sqrt{c}}$, of partial recovery that exists on the graph. This suggests no matter what value of λ it is impossible for full recovery.

What we find in all cases, is the behaviour of the random regular curve is followed by the square lattice curve. With regards the transition points, we see the square lattice abides by the transition point calculated for the random regular graph. This is most likely due to the topologies of the graphs being so similar.

To consider a specific case however, we can investigate the set of instances of $\hat{\rho} = \rho = 1$. On the phase diagram, this point lies out of the partial recovery phase in 5. It also lies to the right of the asymptote at $\rho = \frac{1}{\sqrt{c}}$. At first glance, we expect to see a single region of full recovery for the $\hat{\rho} = \rho = 1$ graphs, but this is not the case. What is shown in 7 is a region of almost no recovery, instead of full recovery. To be more precise, we see recovery, but it is almost entirely incorrect. What is causing this could be the asymptote. It is suggested the asymptote acts as a barrier for the full recovery region, which from the evidence in 3, lies to the left of the asymptote. What remains on the right of the asymptote is a region of incorrect recovery, explaining the results we see for $\hat{\rho} = \rho = 1$ - at this high suppression, the graph can only guess the result of the matching. In the large graph limit, we expect this region to correspond to the graph instances with recovery error tending to 1.

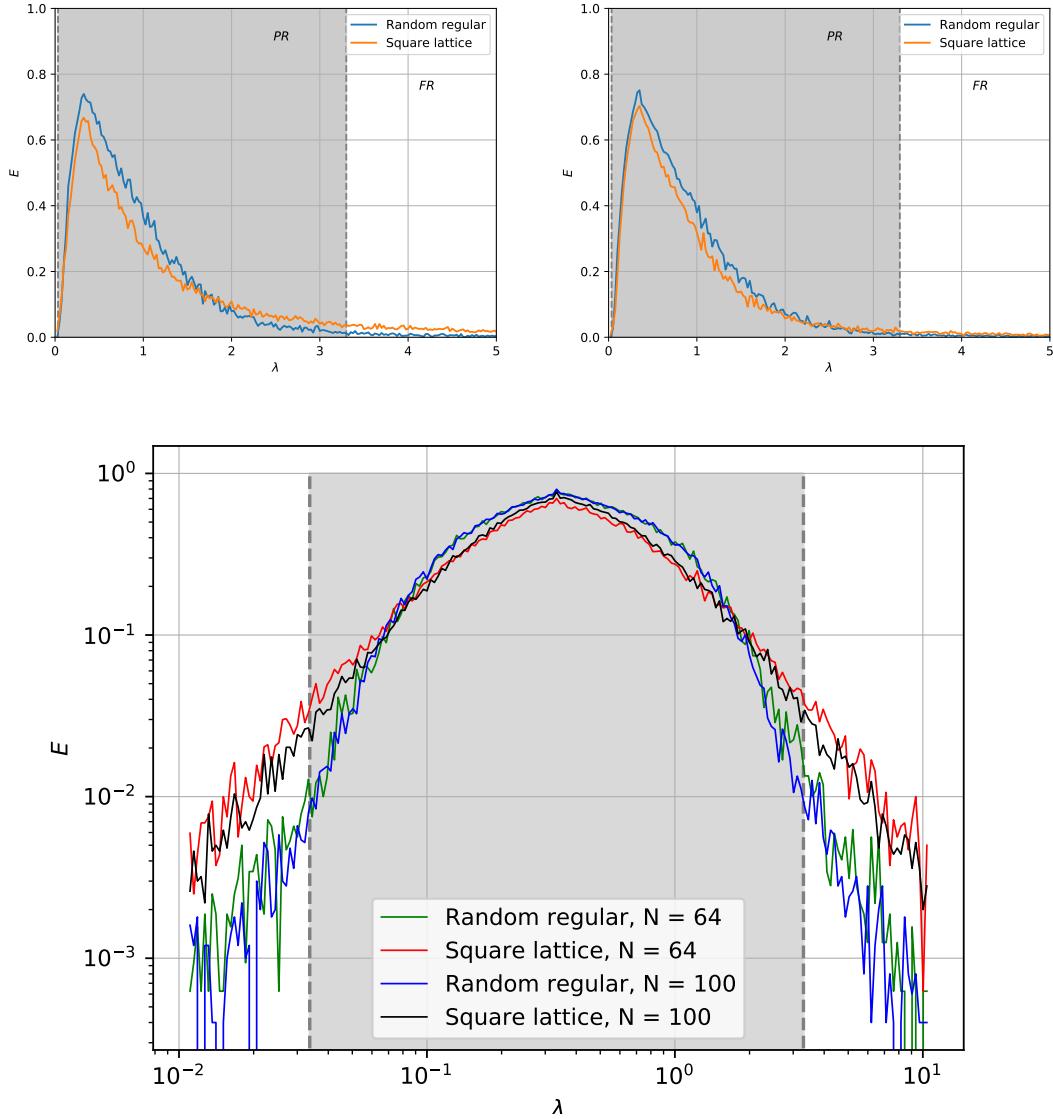


FIGURE 4. Top left and top right: Error E plotted against λ for $N = 64$ and $N = 100$, with $\hat{\rho} = \rho = 0$. Bottom: Log plot of error E plotted against λ . Shaded region represents partial recovery region, as calculated by the boundary equation.

3.4. Conclusion. We have shown there exists a phase transition for the square lattice between a region of partial recovery and full recovery of a planted matching. We also provide evidence the square lattice behaves similarly to the random regular lattice when subject to the same statistical inference of planted matchings. This is displayed through the similar behaviour both models exhibit as initial parameters are changed. Finally we provide evidence a third state exists, a ‘no’ recovery phase, in which the recovered matching error tends to 1 in the large graph limit.

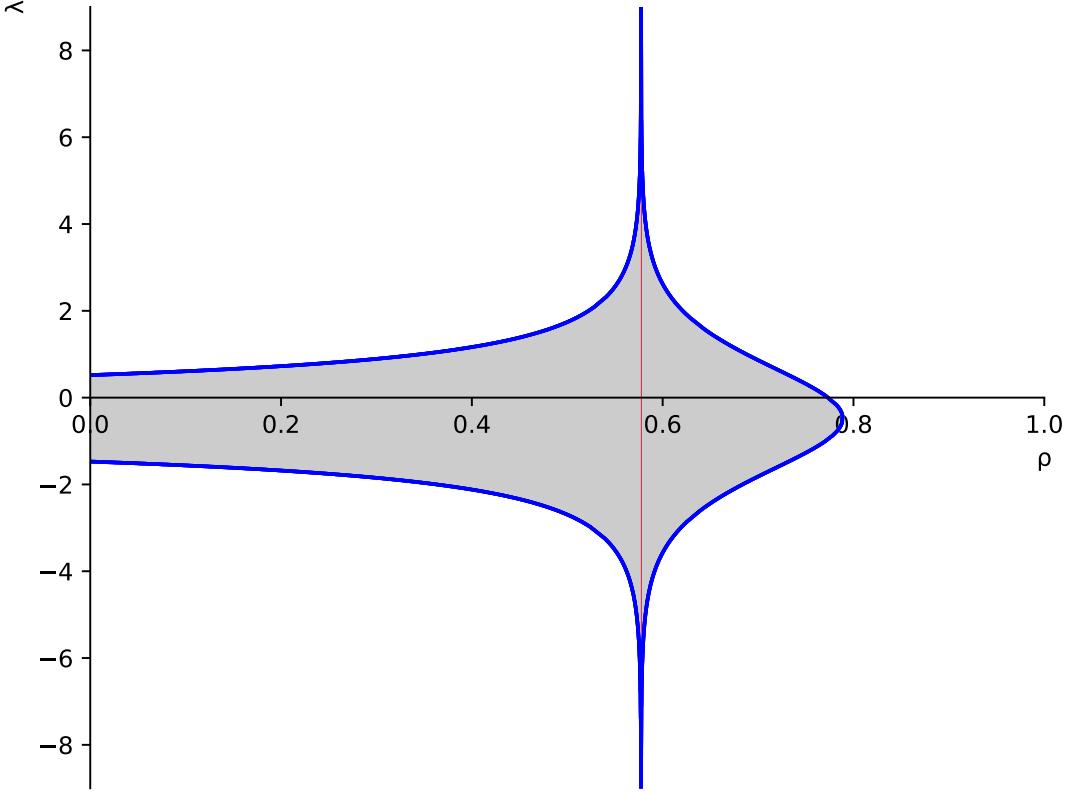


FIGURE 5. Phase diagram of planted matching problem over the random regular graph of coordination $c = 3$, with suppression $\hat{\rho} = \rho$. The λ axis intervals represent a logarithmic scale base 10. The grey area represents the region of partial recovery, while the white region represents full recovery. The blue line, bounding the partial recovery phase, is the vanishing velocity condition. The red line is the asymptote of the boundary in the $\lambda \rightarrow \pm\infty$ where $\rho \rightarrow \frac{1}{\sqrt{c}} = \frac{1}{\sqrt{3}}$

3.5. Future work. There is still plenty to investigate with regards the relationship between random regular graphs and lattice graphs in the context of this statistical inference. One obvious avenue of investigation would be to reduce error in the calculations. To make results more accurate one would have to reduce both finite size error and boundary effect error. Finite size error can be solved by increasing the size of the graphs. To expect BP to solve this challenge is unreasonable. Another way in which this can be approached is by using a population algorithm to iteratively replace elements using an update rule - in this case the recursive distributional equations - of large sets which represent random entries of the distributions \hat{H} and H . After many replacements of the distributions \hat{H} and H , over many instances of initial conditions, an average could be calculated and plotted over λ . This would be faster to complete for large size graphs.

Another point to consider would be to calculate the contribution causing the disparity between t error of the square lattice and the random regular graph. The boundary effect would be reduced by implementing the system over periodic boundary conditions - wrapping the graph

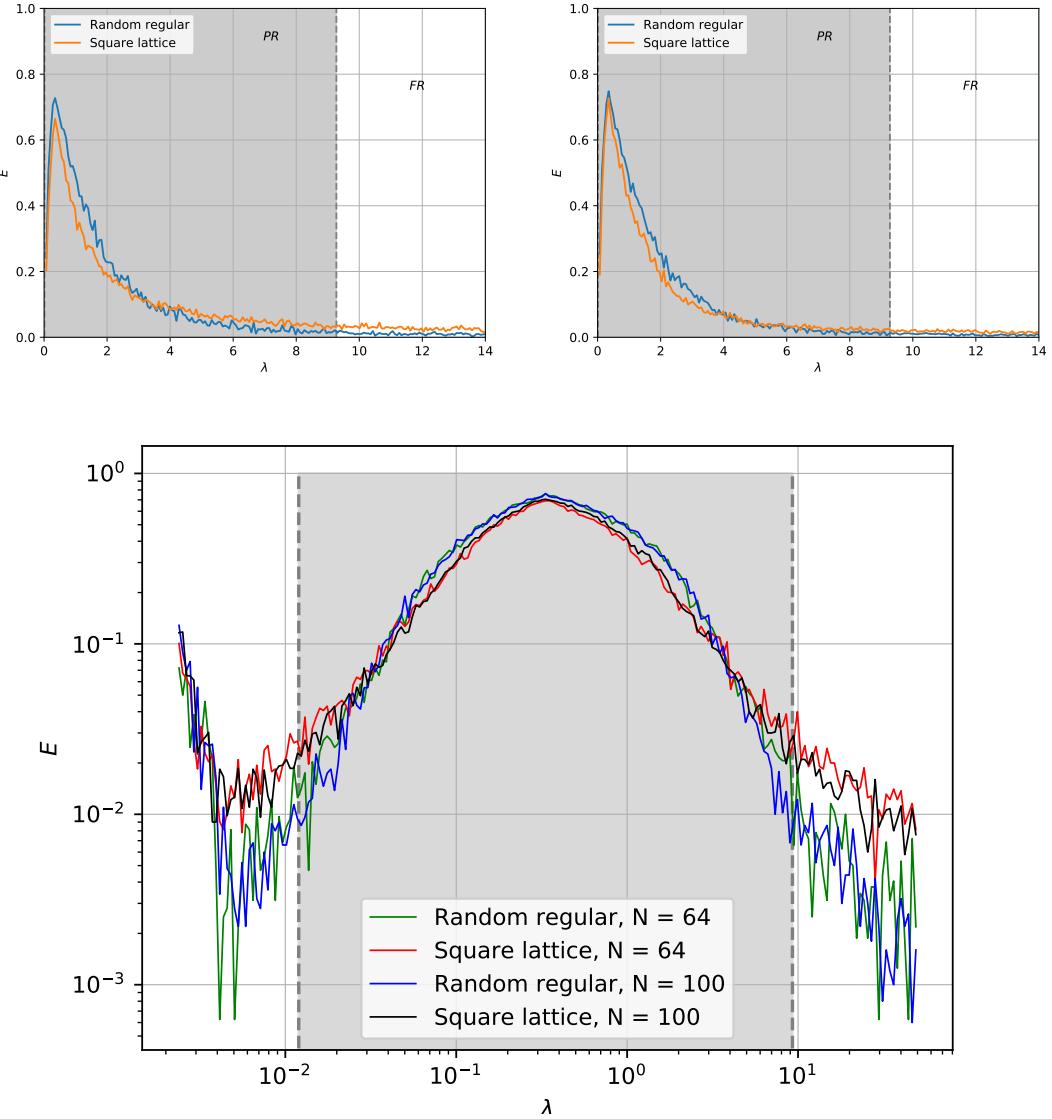


FIGURE 6. Top left and top right: Error E plotted against λ for $\rho = \hat{\rho} = \frac{1}{3}$ with $N = 64$ and $N = 100$. Bottom: Log plot of error E plotted against λ . Shaded region represents partial recovery region, as calculated by the boundary equation.

over a torus. More specifically, with boundary on the lattice graph, we notice that there exist vertices, the boundary vertices, with coordination $c \neq 3$. By applying periodic boundary conditions, all vertices would have coordination $c = 3$, allowing for a more accurate comparison between the square lattice and the random regular graph of coordination $c = 3$. By reducing this discrepancy, it allows the investigation of a more precise understanding of the difference between the recovered matching of the lattice and the random regular graph caused by the existence of short length cycles in the lattice, which are not present in the random regular graph. A way to

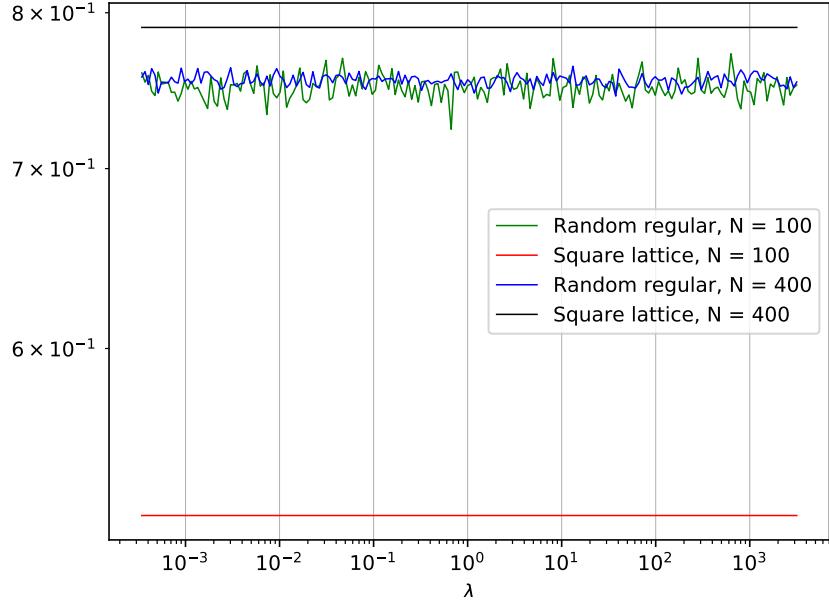


FIGURE 7. Error E plotted against λ for $\rho = \hat{\rho} = 1$ with $N = 100$ and $N = 400$. There exists no division of regions as these initial conditions lay out of the partial recovery phase.

graph this difference would be to find how the average recovery error changes as one transitions between a random regular graph and a lattice. One way to do this would be to start with a lattice and then alter that lattice by reconnecting an edge at random to another randomly chosen node in such a way that after enough iterations of this process, one would be left with a random regular graph. After each iteration of altering the lattice, the average reconstruction error would be calculated and the transition could be graphed.

The no recovery region found in 7 could also be investigated, providing more evidence and possibly an analytic argument for the incorrect recovery region.

A final route one could take is to alter the initial probability distributions. This most likely would result in different phase diagrams between partial recovery and full recovery along with transition points between models.

APPENDIX A

Supplementary material

1. Maximum Entropy Inference

The principle of Maximum Entropy Inference, or MaxEnt inference, calculates the probability distribution for a support, given certain constraints. This is achieved using Lagrange multipliers to maximise the entropy of the distribution. To begin, let us define entropy as $S[p] = -\sum_{\Omega} p(\sigma) \ln(p(\sigma))$ and identify the two constraints in question: $1 = \sum_{\Omega} p(\sigma)$ which ensures the distribution is normalised and $E = \langle E(\sigma) \rangle = \sum_{\Omega} E(\sigma)p(\sigma)$, where E is a dummy variable. From this, we can define our Langrangian, \mathcal{L} :

$$(90) \quad \mathcal{L} = -S[p] - \lambda_0 \left(1 - \sum_{\Omega} p(\sigma) \right) - \lambda_1 \left(E - \sum_{\Omega} E(\sigma)p(\sigma) \right)$$

where λ_0, λ_1 are Lagrange parameters for each constraint. By partially differentiating with respect to $p(\hat{\sigma})$, $\hat{\sigma} \in \Omega$, we find:

$$(91) \quad \frac{\partial \mathcal{L}}{\partial p(\hat{\sigma})} = \ln(p(\hat{\sigma})) + 1 + \lambda_0 + \lambda_1 E(\hat{\sigma}) = 0$$

$$(92) \quad \frac{\partial \mathcal{L}}{\partial \lambda_0} = \sum_{\Omega} p(\hat{\sigma}) - 1 = 0$$

$$(93) \quad \frac{\partial \mathcal{L}}{\partial \lambda_1} = \sum_{\Omega} E(\hat{\sigma})p(\hat{\sigma}) - E = 0$$

As $\hat{\sigma}$ is arbitrary, we can let $\hat{\sigma} = \sigma$. From this, we recover the constraints from 92 and 93 and from 91 we see:

$$\begin{aligned} 0 &= \ln(p(\sigma)) + 1 + \lambda_0 + \lambda_1 E(\sigma) \\ \ln(p(\sigma)) &= -1 - \lambda_0 - \lambda_1 E(\sigma) \\ p(\sigma) &= e^{-1-\lambda_0-\lambda_1 E(\sigma)} \\ p(\sigma) &= \frac{1}{Z} e^{-\lambda_1 E(\sigma)} \\ (94) \quad p(\sigma) &= \frac{1}{Z} e^{-\beta E(\sigma)} \end{aligned}$$

We have defined $Z = e^{1+\lambda_0}$ and $\beta = \lambda_1$. Z can then be calculated with respect to $p(\sigma)$ using the normalisation constraint. This results in 94 taking a form similar to a Boltzmann distribution, where Z is a normalisation constant and β is inverse temperature.

2. On the $\beta \rightarrow 0$ limit

We take the distribution for $p(\sigma)$ in the $\beta \rightarrow 0$ limit:

$$\begin{aligned}
(95) \quad \lim_{\beta \rightarrow 0} p(\sigma) &= \lim_{\beta \rightarrow 0} \frac{1}{Z} e^{-\beta E(\sigma)} \\
&= \lim_{\beta \rightarrow 0} \frac{1}{Z} \cdot \lim_{\beta \rightarrow 0} e^{-\beta E(\sigma)} \\
&= \lim_{\beta \rightarrow 0} \frac{1}{Z} \cdot 1 = \lim_{\beta \rightarrow 0} \frac{1}{Z} = \frac{1}{\lim_{\beta \rightarrow 0} Z}
\end{aligned}$$

By considering the limit of Z as $\beta \rightarrow 0$:

$$\begin{aligned}
(96) \quad \lim_{\beta \rightarrow 0} Z &= \lim_{\beta \rightarrow 0} \sum_{\Omega} e^{-\beta E(\sigma)} = \sum_{\Omega} \lim_{\beta \rightarrow 0} e^{-\beta E(\sigma)} \\
&= \sum_{\Omega} 1 \\
&= |\Omega|
\end{aligned}$$

Therefore, by combining 95 and 96, we find in the limit as $\beta \rightarrow 0$:

$$(97) \quad p(\sigma) = \frac{1}{|\Omega|}$$

3. On the $\beta \rightarrow \infty$ limit

We now take the distribution for $p(\sigma)$ in the $\beta \rightarrow \infty$ limit:

$$\begin{aligned}
(98) \quad \lim_{\beta \rightarrow \infty} p(\sigma) &= \lim_{\beta \rightarrow \infty} \frac{1}{Z} e^{-\beta E(\sigma)} \\
&= \lim_{\beta \rightarrow \infty} \frac{1}{Z} \cdot \lim_{\beta \rightarrow \infty} e^{-\beta E(\sigma)} \\
&= \frac{1}{\lim_{\beta \rightarrow \infty} Z} \cdot \lim_{\beta \rightarrow \infty} e^{-\beta E(\sigma)}
\end{aligned}$$

Before we proceed, let us define first $\Omega_0 \in \Omega$ to be the set of configurations σ that minimise the energy function; $\Omega_0 = \{\hat{\sigma} \in \Omega : E(\hat{\sigma}) = E_0\}$ where we have defined $E_0 = \min_{\Omega} E(\sigma)$. Now we turn our attention to the limit of Z as $\beta \rightarrow \infty$:

$$\begin{aligned}
(99) \quad \lim_{\beta \rightarrow \infty} Z &= \lim_{\beta \rightarrow \infty} \sum_{\Omega} e^{-\beta E(\sigma)} \\
&= \lim_{\beta \rightarrow \infty} \left(\sum_{\Omega_0} e^{-\beta E(\sigma)} + \sum_{\Omega \setminus \Omega_0} e^{-\beta E(\sigma)} \right) \\
&= \lim_{\beta \rightarrow \infty} \sum_{\Omega_0} e^{-\beta E(\sigma)} + \lim_{\beta \rightarrow \infty} \sum_{\Omega \setminus \Omega_0} e^{-\beta E(\sigma)} \\
&= \lim_{\beta \rightarrow \infty} |\Omega_0| \cdot e^{-\beta E_0} + 0 \\
&= |\Omega_0| \lim_{\beta \rightarrow \infty} e^{-\beta E_0}
\end{aligned}$$

Similarly we see for the limit of $e^{-\beta E(\sigma)}$ as $\beta \rightarrow \infty$:

$$(100) \quad \lim_{\beta \rightarrow \infty} e^{-\beta E(\sigma)} = \lim_{\beta \rightarrow \infty} e^{-\beta E(\sigma)} \mathbb{I}(\sigma \in \Omega_0) = \lim_{\beta \rightarrow \infty} e^{-\beta E_0} \mathbb{I}(\sigma \in \Omega_0)$$

where we have used the fact if $\sigma \notin \Omega_0$, in the limit as $\beta \rightarrow \infty$, $e^{-\beta E(\sigma)} \rightarrow 0$. By substituting 99 and 100 back into 98 we obtain:

$$\begin{aligned}
 \lim_{\beta \rightarrow \infty} p(\sigma) &= \frac{1}{|\Omega_0|} \lim_{\beta \rightarrow \infty} e^{-\beta E_0} \lim_{\beta \rightarrow \infty} e^{-\beta E_0} \mathbb{I}(\sigma \in \Omega_0) \\
 &= \frac{1}{|\Omega_0|} \lim_{\beta \rightarrow \infty} \frac{1}{e^{-\beta E_0}} \lim_{\beta \rightarrow \infty} e^{-\beta E_0} \mathbb{I}(\sigma \in \Omega_0) \\
 &= \frac{1}{|\Omega_0|} \lim_{\beta \rightarrow \infty} \frac{e^{-\beta E_0}}{e^{-\beta E_0}} \mathbb{I}(\sigma \in \Omega_0) = \frac{1}{|\Omega_0|} \lim_{\beta \rightarrow \infty} 1 \cdot \mathbb{I}(\sigma \in \Omega_0) \\
 (101) \quad &= \frac{1}{|\Omega_0|} \mathbb{I}(\sigma \in \Omega_0)
 \end{aligned}$$

4. Factor graph of probability

Suppose a group of 5 friends are trying to choose between 2 activities, A or B , to go to. 4 of the friends have a preference, with probability p_i to choose A and $1 - p_i$ to choose B . The 5th player, enjoying both activities equally, plays the role of an unbiased arbitrator. They decide to play a game of elimination to decide, which follows this structure:

- The 4 friends cast their votes, $v_i \in \{A, B\}$, for the activity.
- Votes from friend 1 and 2 are compared; if their choice of activity coincides, $v_1 = v_2$, that choice, we shall rename $v_{1\&2}$ will represent the two friends as they move to the next round. If not, the choice representing the two friends will be decided by the fair arbitrator friend. This same process is carried out for friends 3 and 4 to determine $v_{3\&4}$.
- With a representative choice for both pairs of friends, the same selection process is used for the final selection of activity, v_f , the group will partake in.

If we consider an arbitrary outcome, $\underline{v} = (v_1, v_2, v_3, v_4, v_{1\&2}, v_{3\&4}, v_f)$, from all possible outcomes, $\Omega = \{A, B\}^7$ we see the probability of \underline{v} can be calculated as:

$$(102) \quad P(\underline{v}) = P(v_{1\&2}) \cdot P(v_{3\&4}) \cdot P(v_f) \cdot \prod_i P(v_i)$$

$$(103) \quad = f_1(v_{1\&2}, v_1, v_2) \cdot f_2(v_{3\&4}, v_3, v_4) \cdot f_3(v_f, v_{1\&2}, v_{3\&4}) \cdot \prod_i f_{3+i}(v_i)$$

(104) where for example $P(v_{1\&2}) = P(V_{1\&2} = v_{1\&2})$

$$(105) \quad = \left(p_1 p_2 + \frac{p_1(1-p_2) + p_2(1-p_1)}{2} \right) \delta_{v_{1\&2}, A}$$

$$(106) \quad + \left((1-p_1)(1-p_2) + \frac{p_1(1-p_2) + p_2(1-p_1)}{2} \right) \delta_{v_{1\&2}, b}$$

$$(107) \quad = f(v_{1\&2}, p_1, p_2)$$

$$(108) \quad \Rightarrow P(v_{1\&2}) = f_1(v_{1\&2}, v_1, v_2)$$

We have used the fact p_1 and p_2 are functions of v_1 and v_2 respectively. Similar equations hold for $P(v_{3\&4})$ and $P(v_f)$. Finally the indices of the functions f_i are used to represent the different factors of $P(\underline{v})$, not uniqueness of the functions. This shows we can write this simple probabilistic system in a convenient factorised form.

Bibliography

- [1] Maria Chiara Angelini, Carlo Lucibello, Giorgio Parisi, Gianmarco Perrupato, Federico Ricci-Tersenghi, and Tommaso Rizzo. The loop expansion around the bethe solution at zero temperature predicts an upper critical dimension equal to 8 for spin glass models in a field, 2021.
- [2] S. Caracciolo, R. Fabbricatore, M. Gherardi, R. Marino, G. Parisi, and G. Sicuro. Criticality and conformality in the random dimer model. *Physical Review E*, 103(4), Apr 2021.
- [3] P. G. de Gennes. Exponents for the excluded volume problem as derived by the Wilson method. *Phys. Lett. A*, 38:339–340, 1972.
- [4] William Jockusch, J. Propp, and P. Shor. Random domino tilings and the arctic circle theorem. *arXiv: Combinatorics*, 1998.
- [5] R. Kenyon. An introduction to the dimer model. *arXiv: Combinatorics*, 2003.
- [6] Marc Mezard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, Inc., USA, 2009.
- [7] Marc Mézard and Giorgio Parisi. The euclidean matching problem. <http://dx.doi.org/10.1051/jphys:0198800490120201900>, 49, 12 1988.
- [8] Marc Mézard and Giorgio Parisi. The cavity method at zero temperature. *Journal of Statistical Physics*, 111(1):1–34, 2003.
- [9] Marc Mézard and Giorgio Parisi. Mean-field equations for the matching and the travelling salesman problems. *EPL (Europhysics Letters)*, 2:913, 07 2007.
- [10] Heidelberg Colloquium on Glassy Dynamics, J. L. van Hemmen, and I. Morgenstern. *Heidelberg Colloquium on Glassy Dynamics : proceedings of a colloquium on spin glasses, optimization and neural networks held at the University of Heidelberg, June 9-13, 1986 / edited by J.L. van Hemmen and I. Morgenstern*. Springer-Verlag Berlin ; New York, 1987.
- [11] A. Pagnani, G. Parisi, and M. Ratiéville. Near-optimal configurations in mean-field disordered systems. *Physical Review E*, 68(4), Oct 2003.
- [12] Guilhem Semerjian, Gabriele Sicuro, and Lenka Zdeborová. Recovery thresholds in the sparse planted matching problem. *Physical Review E*, 102(2), Aug 2020.
- [13] Alok Bhushan Shukla. A short proof of cayley’s tree formula, 2016.
- [14] Gordon Slade. Self-avoiding walk, spin systems and renormalization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2221):20180549, Jan 2019.
- [15] Lenka Zdeborová and Florent Krzakala. Statistical physics of inference: thresholds and algorithms. *Advances in Physics*, 65(5):453–552, Aug 2016.

APPENDIX B

Self Assessment

Overall, I am pleased with the project. In preparation for the write up, I read plenty of material around the subject. Not all of it was relevant for the write up, but it certainly gave me a better understanding of the material I decided to add to the report. The direction of the project was not established early on, allowing me to find interesting papers and consider various routes into which I could investigate. The numerical aspect of the project was very rewarding. The code was time consuming to run; taking hours for a single plot. Optimisation of computational algorithms is certainly something I appreciate now. On the numerical portion of the project, I am extremely pleased with my understanding of the material presented in the paper. Meeting online with my supervisor was not an environment conducive of conveying mathematical information, so I spent a considerable amount of time trying to fully understand the content and I feel proud of the level of understanding I reached. I made some insightful comments while analysing the numerical data, however my inexperience with the field prevented me from seeing certain subtle aspects.

- Student initiative and amount of guidance required: 65
- Scientific Quality: 75
- Breadth: 65
- Originality: 65
- Presentation and logical structure of the report: 80