# Community detection on random graphs using Potts models

## How random graphs with Potts models behave as random graphs with embedded community structure

Miltenburg, Steven J.G.

*Award date:*
2022

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science (MCS) & Applied Physics (AP)
Master track (AP) Bio/Nanoscience and Technology
Research Groups Statistics, probability and operations research (SPOR) & Soft matter and biological physics (SMB)

# Community detection on random graphs using Potts models

*How random graphs with Potts models behave as random graphs with embedded community structure*

SJG Miltenburg
Student number 0959623
60 EC

Supervisors:
Prof. Dr. Remco van der Hofstad (SPOR, MCS)
Dr. Wouter Ellenbroek (SMB, AP)
Dr. Pim van der Hoorn (SPOR, MCS)

Committee memebers:
Prof. Dr. Remco van der Hofstad
Dr. Wouter Ellenbroek
Dr. Pim van der Hoorn
Prof. Dr. Kees Storm

This is a public thesis which follows the TU/e code of scientific integrity

Eindhoven, March 2022

# Abstract

In this paper, we investigate the stochastic block model on $n$ vertices, which first assigns a community from $\{1, ..., q\} = [q]$ to every vertex and then connects two vertices based on their communities. Two vertices in the same community are connected with probability $p_{\text{in}}$ while two vertices of different communities are connected with probability $p_{\text{out}}$. The community detection problem is to reconstruct the original community assignment based on the edge structure of the resulting graph and the connection probabilities, better than a random guess.

In particular, we investigate the sparse stochastic block model, where $p_{in} = a/n$ and $p_{\text{out}} = b/n$, for $a, b \in \mathbb{R}^+$. This way, when $n \to \infty$, the average degree remains finite. When $n \to \infty$, the stochastic block model can be investigated using the local limit, which represents the neighbourhood of a random vertex $\rho$. For the stochastic block model, this is a random tree.

Physicists have tackled the community detection problem using belief-propagation algorithms. Using these algorithms and non-rigorous arguments, the authors of [1] have established a threshold for when this detection can be done in polynomial time.

We aim to connect these different fields of community detection and show why the assumptions made by the physicists are reasonable and why their algorithm should detect communities.

We observe the marked local limit of a sparse stochastic block model, where community $i$ is assigned to vertices with probability $\frac{1}{q}$. We show that this object is equal to a Potts model on a specific random tree. This tree is the local limit of the Erdös-Rényi random graph, which connects two vertices with probability $\lambda/n$. The constraints for which these models are equal are $\lambda = \sum_{j\in[q]} c_{ij} n_j \forall i \in [q]$, the magnetic field $\vec{B} = \mathbf{0}$ and the inverse temperature $\beta = -\frac{1}{2}\log(\frac{(q-1)b}{a})$. We do this by showing the equivalence of both models to a broadcasting process. A broadcasting process is defined on trees by the authors of [2]. We also show that both equivalences with the broadcasting process can be generalized further.

We show that using these constraints of the parameters, the community detection threshold on graphs and that of a broadcasting process on trees are the same. Using these constraints, the threshold for community detection on random graphs and the Potts model on trees do not match, but we think this is caused by our misinterpretation of the results in the literature.

We extend all of these results to degree-corrected graphs, where each vertex is also assigned a weight $\phi$. Thus, two vertices $u$, $v$ are connected with probability $\phi(u)\phi(v)p_{\text{in}}$, if they have the same community, and $\phi(u)\phi(v)p_{\text{out}}$ otherwise.

We then discuss the origin of belief propagation algorithms. We explain in what ways approximations are taken when applying these algorithms to random graphs whose local limit is a tree. We explain why intuitively the algorithm of the authors of [1] should detect communities.

Finally, we discuss a more mathematical description of these algorithms, which have been shown to calculate many physical properties exact on trees by the authors of [3]. One of these properties, the free energy density, can be used to determine the quality of a guess of the original communities, when the algorithm of the authors of [1] gives multiple results.

# Contents

# Chapter 1

# Introduction

Suppose we are given a graph of all people in Belgium, who are connected if they have ever called each other. Can we now figure out who lives in french-speaking Wallonia and who in dutch-speaking Flemish region? This example has been studied and is called community-detection. We say a graph has a community structure if it divides into groups which have more connections within their groups than outside of it. In the Belgian calling graph we expect such a structure, as the assumption is that people are more likely to call someone with the same native language. There is also another type of community structure, where there are more connections between different groups than within one. One example of this is a graph containing all marriages, where we want to guess the sex. We call the Belgian community structure assortative and the marriage community structure disassortative. We dedicate most of our work to the assortative case.

Now in this problem the question is usually when a community structure can be detected. For this the original communities need to be known, so that a guess of the communities can be compared to it. One random graph model that fulfills this requirement is the stochastic block model. It creates a graph by first dividing nodes into their original community structure by giving them a type. Then it connects edges between nodes based on their type. We will look at assortative graphs where the probability of connecting to the same type $p_{\text{in}}$ is larger than the probability of connecting to other types $p_{\text{out}}$. It has been defined that community detection is successful if, based on the structure of the resulting graph, the original communities can be guessed better than with a random guess.

In most practical cases we are interested in large networks. We will therefore investigate the limit case where the amount of vertices $n$ goes to infinity and investigate if this limit behaviour is present for large finite $n$. In order for such a limit to be nicely behaved, we will need sparse graphs, where $p_{\text{in}} = a/n$ and $p_{\text{out}} = b/n$ with $a, b$ constant, such that the average degree is finite as $n \to \infty$.

In this limit we can quantify the graph models we are interested in, by looking at the neighbourhood of a random vertex. We call this object a local limit. These local limits help us, not only rigorously, but also intuitively better understand these graph models.

Mathematicians have mainly focused on theoretically determining when a the structure of a graph does not give enough information to successfully detect communities. It has been proven by the authors of [4] that community detection in a model of $q$ communities is impossible for $(a-b)^2 < q(a+b)$.

Also physicists have tried to tackle this problem of community detection on sparse graphs. They mainly use algorithms which try to guess the original communities. These algorithms are called belief-propagation algorithms, and are based on properties of the Boltzmann distribution of statistical physics models, which can be calculated iteratively on trees. They then take these iterative calculation as the basis for an iterative algorithm on locally treelike graphs. Using these algorithms they empirically find that they can detect communities above a threshold. We show that this threshold corresponds to $(a-b)^2 > q(a+(q-1)b)$.

In this thesis we investigate when community detection is successful on sparse random graphs. We make a distinction between when successful community community detection is theoretically possible, and when there is an algorithm which can detect it easily, in polynomial time of $n$. Both mathematicians and physicists have tackled this problem of community detection. We therefore try to couple the parameters of the models used by mathematicians to those used by physicists, and couple results for community detection gained in one field to the other field. We also try to explain why the algorithms created by physicists should indeed detect communities.

We shall now give an overview of what we discuss in what chapter of this thesis. In chapter 2 we will give the background knowledge required for both the mathematical and physical arguments. We define the graph models used. We show that these models can for $n \to \infty$ be described as the neighbourhood of a random vertex, the local limit. Next we describe the kind of object this local limit is, a Galton Watson Tree. This will be important, as our physical models will have a much simpler local description on trees. Next, we talk about when our random graph models have a nonzero fraction of nodes in the largest connected component, a so-called Giant. This is called percolation in physics. This is important as the physical models will only have a phase transition, when the graph they are applied to has nonzero percolation, a Giant. This is important as the phase transition separates the regime where community detection is impossible and the regime where it is successful.

We then explain the Ising and Potts models and their properties on aforementioned graph models in chapter 3. We show that physics has developed many quantities to quantify these systems. In particular the free energy, which in minimized at constant temperature, can give us information. Furthermore we show when there is phase transition, the moment when the type of one node in the system tells us something about the entire system. This is important because we later try to relate this with the threshold of community detection.

We need all this in order to understand the significance of the equivalences we prove in chapter 4. We show that the local limit of the stochastic block model and using a Potts model on the local limit of a random graph without pre-defined types, where each vertex has equal probability $\lambda$ of connecting to another node, the Erdös-Renyi random graph, are equivalent. We do this by first showing the equivalence on trees and then using the fact that the local limit is a tree. This allows us to link the temperature of a Potts model and the average degree of the random graph it is defined on, to $p_{\text{in}}$ and $p_{\text{out}}$ in a stochastic block model. This means that we can express the results found by physicists and mathematicians in the same notation, so that we can compare them.

We then show why using our method, we cannot use the same method to extend this equivalence from the stochastic block model to a model where the probability of connecting can be dependent of the incoming and outgoing type, the inhomogeneous graph model.

In chapter 5 we look at algorithms that have been developed for community detection on finite but large graphs. We look in particular at the algorithm of [1] and give a detailed explanation of how this method was developed on trees. We then show how it can be applied to locally treelike graphs. We explain that this method uses only approximations of $\mathcal{O}(1/n)$ and therefore should give accurate results for large $n$.

In chapter 6 we use factor graphs, which originally come from problems in computer science, to have a more rigorous way of describing the belief propagation algorithms explained in the previous chapter. This way we are also able to iteratively calculate the free energy of a Boltzmann distribution. As the free energy is minimized in this distribution for constant temperature, this allows us to compare the quality of multiple different guesses of the communities without needing the original communities.

In chapter 7 we summarize our results and their limitations. We also show a few directions where new research for community detection can be done.

# Chapter 2

# Random graph background

In order to understand this thesis, some probability theory knowledge will be required. First we discuss random variables and types of convergence in section 2.1. In section 2.2 we define the random graph models usually used in community detection. We will see that as $n \to \infty$, the typical neighbourhood of a random vertex in a random graph can be described. We call this description a local limit, which we discuss in section 2.3. For the considered random graph models, these local limits will usually be a Galton-Watson tree. These are random trees from a population model where each vertex is seen as a parent which either dies out or has an offspring. The extinction of these typical neighbourhoods can be related to the size of the largest connected component of a graph. When this largest component contains a positive fraction we call it a giant, which is the subject of section 2.4. The existence of the giant is necessary because it impacts when there is a phase transition of the statistical physics models that we will discuss in the next chapter. The phase transition in turn will impact the threshold for community detection. In section 2.5 we define when community detection is successful both on graphs and on trees. We give community detection results of the mentioned graph models. In section 2.6 we describe a process that imposes a community structure on a tree, and when successful community detection is possible in this setting.

## 2.1 Probability theory notation

In this thesis we will work with discrete random variables. As these definitions can get quite technical, we will use an example.

Suppose we want to model the throwing of a balanced 6-sided dice. In this case we can take our random variable $X$ to be able to take the integer values 1 through 6. We call the set of states which $X$ can have the state space $\Omega = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$. In particular, as the dice is balanced, we want the probability of being equal to any one of these values to be 1/6. For example, the probability of $X = 1$ is 1/6, which we denote as $\mathbb{P}(X = 1) = \frac{1}{6}$. When we roll the die, the probability of rolling any result is 1. We can also ask ourselves what the chances are of rolling a number $\leq 4$, which we denote as $F_X(4) = \mathbb{P}(X \leq 4) = \frac{4}{6}$. Another probability of interest is probability of rolling a 2 or a 5 is. Let $A = \{\{2\}, \{5\}\}$, then we denote this probability as $\mathbb{P}(X \in A) = \mathbb{P}(X = 2) + \mathbb{P}(X = 5) = \frac{2}{6}$. Also a very important property is the expected value of a roll. In this case we say that the expected value $\mathbb{E}[X] = \frac{1}{6} + \frac{2}{6} + \frac{3}{6} + \frac{4}{6} + \frac{5}{6} + \frac{6}{6} = 3\frac{1}{2}$. We also introduce a useful tool, the probability-generating function, $G(z) = \mathbb{E}(\mathbb{z}^{\mathbb{X}})$, which can be used to calculate the moments $\mathbb{E}[\mathbb{X}^{\mathbb{k}}]$. Given this intuitive example, we can now introduce the formal definitions.

**Definition 2.1.1** (State space, (discrete) random variable, probability measure, cumulative distribution function, expected value, probability-generating function)**.** *Let $\Omega$ be a set of states, the state space. $X$ is a random variable with respect to a measure $\mathbb{P}$, if $X$ takes values of $\Omega$ with probability $\mathbb{P}(X = x) \geq 0$ for $x \in \Omega$. Also for any countable set of disjoint events $\{x_i\}$ with $x_i \in \Omega$ we have*

$$\mathbb{P}(X \in \cup_i (x_i)) = \sum_i \mathbb{P}(X = x_i). \tag{2.1}$$

*$\mathbb{P}$ is called a probability measure if*

$$\mathbb{P}(X \in \Omega) = \sum_{x \in \Omega} \mathbb{P}(X = x) = 1. \tag{2.2}$$

*$F_X$ is called the cumulative distribution function. When $\Omega \subseteq \mathbb{R}$ , there is a natural ordering of the states and we can define*

$$F_X(x) = \mathbb{P}(X \leq x). \tag{2.3}$$

*Otherwise we must first create an ordering, using a mapping $\phi : \Omega \to \mathbb{R}$. Then one can define*

$$F_X(x) = \mathbb{P}(X \in A), \qquad \textit{with } A = \{\omega \in \Omega | \phi(\omega) \leq x\}. \tag{2.4}$$

*The expected value of a random variable $X$ is then*

$$\mathbb{E}[X] = \langle X \rangle = \sum_{x \in \Omega} \phi(x)\mathbb{P}(X = x). \tag{2.5}$$

*The probability-generating function is defined as*

$$G(z) = \mathbb{E}(\mathbb{z}^{\mathbb{X}}) = \sum_{x=0}^{\infty} \mathbb{P}(X = x)z^x. \tag{2.6}$$

*Two random variables with equal probability-generating function have the same distribution.*

Throughout this thesis measures are called $\mu$, though $\mathbb{P}$ may occur occasionally. Mostly we look at graphs of $n$ vertices, where each vertex has a type in $[q] \equiv \{\{1\}, ..., \{q\}\}$. $\Omega$ will therefore usually be $[q]^n$, denoting all different $q^n$ states the graph can have.

We are interested in what happens to graphs when the number of vertices grows very large, $n \to \infty$. In order to nuance results in this limit, we need to distinguish different strengths of convergence. We start with the weakest convergence, convergence in distribution.

**Definition 2.1.2.** *Let $\{X_n\}_{n \geq 1}$ be a sequence of random variables. Let $F_n$ denote the cumulative distribution function of $X_n$. Then we say $\{X_n\}_{n \geq 1}$ converges in distribution to $X$, $\{X_n\} \xrightarrow{d} X$, when*

$$\lim_{n \to \infty} F_n(x) = F(x) \tag{2.7}$$

*for every point $x \in \mathbb{R}$ at which $F(x)$ is continuous.*

This definition is a weaker form of convergence, thus the limit can behave unusually. For example, for $n > 0$ let $X_n$ be uniform in $(0, \frac{1}{n})$, then this sequence converges in distribution to a degenerate random variable $X = 0$. Checking (2.7), $F_n(x) = 0$ for all $n$ when $x \leq 0$, and $F_n(x) = 1$ for all $x \geq \frac{1}{n}$. However, for this limiting random variable we have $F(0) = 1$, even though $F_n(0) = 0$ for all $n$. Thus the convergence of the cumulative distribution function fails at the point $x = 0$ where F is discontinuous. This is no problem for convergence in distribution, as (2.7) is only required on continuous point of $F(X)$.

Now we take a look at convergence in probability.

**Definition 2.1.3.** *Let $\{X_n\}_{n \geq 1}$ be a sequence of random variables. $\{X_n\}$ converges in probability towards the random variable $X$, $\{X_n\} \xrightarrow{\mathbb{P}} X$ if for all $\epsilon > 0$*

$$\lim_{n \to \infty} \mathbb{P}(|X_n - X| > \epsilon) = 0. \tag{2.8}$$

We can see that the previous example, where $X_n$ is uniform in $(0, \frac{1}{n})$ still works, as $X_n < \frac{1}{n}$, so for $n > \frac{1}{\epsilon}$ we have $\mathbb{P}(|X_n - X| > \epsilon) = 0$.

## 2.2 Random graph models

In this subsection we define several random graph models on which community detection is interesting and look at their properties. In order to be able to test community detection algorithms, we want random graph models which have a community structure built into them, so that we can compare the results of the algorithm with the 'true' community structure. However, we start by introducing the underlying random graph model without community structure, the Erdős–Rényi (ER) random graph. Even these random graphs without community structure can be interesting to us, as we later show that we can use statistical physics methods to introduce a community structure.

After introducing the ER random graph, we show a random graph which has a community structure by itself, the Inhomogeneous random graph (IRG). We mention one specific case of the inhomogeneous random graph, on which much research has been done and many results are known, the stochastic block model (SBM).

As we are mostly interested in large graphs, where $n \to \infty$, we want this limit to be well-behaved. Therefore we introduce the sparse case of random graphs. Intuitively this means that the degrees are of constant, $\mathcal{O}(1/n)$, size even as $n \to \infty$. For the rest of this thesis, we shall denote $\partial u$ for the set of all vertices connected to $u$. Thus $|\partial u|$ denotes the degree of $u$.

**Definition 2.2.1** (sparse graphs)**.** *Let $G_n = (V_n, E_n)$, then we say $G_n$ is (uniformly) sparse when*

$$\lim_{l\to\infty} \limsup_{n\to\infty} \frac{1}{n} \sum_{u\in V_n} |\partial u| \mathbb{1}(|\partial u| \geq l) = 0. \tag{2.9}$$

*Here $\mathbb{1}$ is the indicator function which is 1 when its argument is true and 0 otherwise.*

Or in words, when the amount of edges connected to a node with degree $\geq l$ scales at most linear with $n$. We shall discuss only graphs where the degree of nodes is distributed using a poisson distributed random variable with parameter $\lambda$, $\text{Poi}(\lambda)$, defined as

$$\mathbb{P}(\text{Poi}(\lambda) = x) = \frac{\lambda^x e^{-\lambda}}{x!}. \tag{2.10}$$

For this distribution we will have $G_n$ sparse when $\lambda = \mathcal{O}(1)$.

### The Erdős–Rényi model (ER)

An Erdős–Rényi random graph on $n$ vertices $\text{ER}_n(p)$ is created by creating an edge between 2 vertices with probability $p$ for every pair of vertices independently. Thus the creation of edges is independent of any a priori defined community structure. Thus even if a set of original communities would be put onto the vertices of $\text{ER}_n(p)$, they could never in any way be deduced from the edges of the graph, as these edges are completely independent of said communities. Therefore the ER graph is said to not have any community structure.

We will only investigate the sparse case, where even in the limit $n \to \infty$ the average degree is finite. Thus $p = \frac{\lambda}{n}$. We shall therefore denote the model with $\text{ER}_n(\lambda/n)$, where we are mostly interested in the limit $n \to \infty$. The subscript $n$ will often be omitted. We would like to quantify these graphs for large $n$. One way to do this is to look at a random vertex in $\text{ER}_n(\lambda/n)$. Let $\text{Bin}(n, p)$ denote the binomial distribution simulating $n$ experiments with success probability $p$, defined as

$$\mathbb{P}(\text{Bin}(n, p) = x) = \binom{n}{x} p^x (1-p)^{n-x}. \tag{2.11}$$

For the degree $D_n$ of a random vertex in $\text{ER}_n(\lambda/n)$ we obtain

$$\mathbb{P}(D_n = k) = \binom{n-1}{k} (\frac{\lambda}{n})^k (1 - \frac{\lambda}{n})^k = \mathbb{P}\left(\text{Bin}(n-1, \frac{\lambda}{n}) = k\right). \tag{2.12}$$

As $n \to \infty$, we know that

$$D_n \xrightarrow{d} \text{Poi}(\lambda). \tag{2.13}$$

As poisson random variables have neat properties, this helps us in our description of $\text{ER}_n(\lambda/n)$ as $n \to \infty$. We are interested in the behaviour of the graph as $n \to \infty$, in particular the neighbourhood of a random vertex $\rho$. A more complete picture of this will be given in section 2.3.

Now we would like to introduce graphs with a natural community structure. We can do this by by extending the ER random graph model such that it has different types of nodes. For this we introduce the Inhomogeneous random graph model.

### Inhomogeneous random graph model

We look at the inhomogeneous random graph model on $n$ vertices. Here first each node is assigned a type, which represents its community. Then each edge is formed independently with a probability that is based on the types of the nodes it would connect. For community detection we would sometimes like to keep track of the type of each node that was used to create the graph. When the original types are stored in the graph we call a graph *marked*. More precisely,

**Definition 2.2.2** ((Marked) inhomogeneous random graph)**.** *Let $\mathcal{S} = [q] = \{1, ..., q\}$ be the type space. Let $c$ be a $q \times q$ symmetric non-negative matrix, called a kernel. Let $\mu_\mathcal{S}(i) \equiv n_i$ be the probability of a node having type $i$. Then $\mathrm{IRG}_{n,\mu_\mathcal{S}}(c)$ is a random graph $G = (V(G), E(G))$ with $n = |V(G)|$ vertices created in the following way. Each node $u$ is assigned a type $t_u$ with probability $n_{t_u}$ independently. Then, an edge $(u, v) \in E(G)$ is created with probability $\frac{c_{t_u,t_v}}{n}$ independently for each of the $\binom{n}{2}$ possible edges.*

*We call this graph the marked $\mathrm{IRG}_{n,\mu_\mathcal{S}}(c)$ if the type $t_u$ of each vertex $u$ is stored in $\vec{t} \in \mathcal{S}^n \equiv \Omega$ at the $u$'th index, so $G = (V(G), E(G), \vec{t}(G))$. We usually omit $\mu_\mathcal{S}$ and mention what is known about $\vec{n} = \{n_1, ..., n_q\}$ instead.*

Note that if we set $\mathcal{S} = \{1\}$ and $c = \lambda$, then $\mathrm{IRG}_n(c) = \mathrm{ER}_n(\lambda/n)$. Also, when $c$ contains only values of order $\mathcal{O}(1)$ then $\mathrm{IRG}_n(c)$ is sparse.

### The Stochastic block model (SBM)

The stochastic block model $\mathrm{SBM}_n(q, p_{\mathrm{in}}, p_{\mathrm{out}})$ is a specific case of the IRG model where $n$ vertices are partitioned into $q$ clusters, where the probability of a node having type $i$, $n_i = \frac{1}{q}$. The random graph is created by connecting two vertices of the same type with probability $p_{\mathrm{in}}$ and vertices of a different type with $p_{\mathrm{out}}$. In the sparse case that we are interested in, $p_{\mathrm{in}} = a/n$ and $p_{\mathrm{out}} = b/n$, with constants $a, b > 0$. More precisely, we find $\mathrm{SBM}_n(q, a, b) = \mathrm{IRG}_n(c)$ with

$$c = \begin{pmatrix} \ddots & & \mathbf{b} \\ & a & \\ \mathbf{b} & & \ddots \end{pmatrix} \tag{2.14}$$

a $q \times q$ matrix with $a$ on each diagonal entry and $b$ everywhere else.

### Degree-corrected SBM (DSBM)

Furthermore there is the degree-corrected SBM (DSBM). Degree-corrected means that the degree of the vertices is changed (corrected) by adding a weight to vertices. This way a the degree distribution of the model can be made to fit the degree distribution of a real-world network. In this version, in addition to the properties of $\mathrm{SBM}_n(q, a, b)$, each vertex $u$ is assigned a weight $\phi_u$ governed by a distribution $\nu$ (with a minimum weight $\phi_{min} > 0$) and moments $\Phi^{(k)} = \int_W x^k d\nu(x)$. The probability of connecting two vertices becomes $\frac{\phi_u \phi_v a}{n}$ if $u$ and $v$ share the same type and $\frac{\phi_u \phi_v b}{n}$ otherwise. We write this model as $\mathrm{DSBM}_n(q, a, b, \nu)$. When $q = 1$ and $a = \lambda$ this reduces to the degree-corrected ER model, called $\mathrm{DER}_n(\lambda/n, \nu)$.

The IRG can also be extended by having arbitrary connection probabilities $p_{ij}$ between types, with in the sparse case $c_{ij} = p_{ij}/n$. As our results unfortunately don't apply to such a general model we won't further work it out here.

## 2.3 Local limits

We are often interested in graphs $G$ in the limit $n \to \infty$, the thermodynamic limit. In this limit, it is important to know how the neighbourhood looks from "typical vertex" $\rho$ (i.e. $\rho$ chosen uniformly at random ). It turns out that for the sparse graphs we have defined in the previous section 2.2, there are no cycles up to distance $R \xrightarrow[n\to\infty]{} \infty$. Thus these neighbourhoods can be described as trees.

In this section we introduce the 'local limit' as the convergence of the neighbourhood of a local root $\rho$ chosen uniformly at randomfrom $V(G_n)$ to a graph $G$. In order to understand this local limit we will use the definitions and an introduction to local limits from [5]. The version here will be shorter, so if anything is still unclear, look at [5].

We start by introducing rooted graphs, which allow us to better study the neighbourhood of the root.

**Definition 2.3.1** (Locally finite and rooted graphs)**.** *A rooted graph is a pair $(G; \rho)$, where $G = (V(G); E(G))$ is a graph with vertex set $V(G)$ and edge set $E(G)$, and $\rho \in V(G)$ is a vertex. Further, a rooted or non-rooted graph is called locally finite when each of its vertices has finite degree.*

In definition 2.3.1, graphs can have finitely or infinitely many vertices, but we will have graphs that are locally finite in mind.

**Definition 2.3.2** (Neighborhoods as rooted graphs)**.** *For a rooted graph $(G;\rho)$, we let $B_r(\rho) = B_r^{(G)}(\rho)$ denote the (rooted) subgraph of $(G;\rho)$ of all vertices at graph distance at most $r$ away from $\rho$. Formally, this means that $B_r(\rho) = ((V(B_r^{(G)}(o)), E(B_r^{(G)}(\rho)), \rho)$, where*

$$V(B_r^{(G)}(\rho)) = \{u : d_G(\rho, u) \leq r\}$$
$$E(B_r^{(G)}(\rho)) = \{(u,v) \in E(G) : d_G(\rho, u), d_G(\rho, v) \leq r\} \tag{2.15}$$

.

We continue by introducing the notion of isomorphism between graphs, which basically describes that graphs 'can be drawn to look the same', when the precise indexing of vertices is ignored. Here is the formal definition:

**Definition 2.3.3** (rooted Graph isomorphism)**.** *Two rooted (finite or infinite) graphs $(G_1, \rho_1)$ and $(G_2; \rho_2)$ where $G_i = (V(G_i), E(G_i))$ for $i \in \{1,2\}$ are called isomorphic, which we write as $(G_1; \rho_1) \simeq (G_2; \rho_2)$, when there exists a bijection $\phi : V(G_1) \to V(G_2)$ such that $\phi(\rho_1) = \rho_2$ and $\{u,v\} \in E(G_1)$ precisely when $\{\phi(u), \phi(v)\} \in E(G_2)$.*

We let $\mathcal{G}_*$ denote the space of rooted graphs modulo isomorphisms, introduced by Aldous and Steele [6]. This means that $\mathcal{G}_*$ contains the set of equivalence classes of rooted connected graphs. We will omit the equivalence classes, and write $(G, o) \in \mathcal{G}_*$, as all $(G', \rho')$ such that $(G'; \rho') \equiv (G; o)$ are considered to be the same. Thus, formally we deal with equivalence classes of rooted graphs. In the literature, the equivalence class containing $(G, \rho)$ is sometimes denoted as $[G, \rho]$.

Now we finally have all the ingredients needed to define a local limit. Intuitively this local limit should be the object such that building up the graph $G$ from the root $\rho$ has the distribution that looking at a random root $\rho_n$ in the finite graph $G_n$ converges to. In this case the distribution is up to graph isomorphisms and the convergence is in probability.

**Definition 2.3.4** (local convergence of random graphs)**.** *we say that $G_n$ converges locally in probability to $(G, \rho)$ having (possibly random) distribution $\mu$ when*

$$\mathbb{E}[h(G_n, \rho_n)|G_n] \xrightarrow[n\to\infty]{\mathbb{P}} \mathbb{E}_\mu[h(G, \rho)] \tag{2.16}$$

*for every bounded and continuous function $h : \mathcal{G}_* \to \mathbb{R}$. On the left-hand side the expectation $\mathbb{E}$ is w.r.t. the random vertex $\rho_n$, as the sequence of graphs $G_n$ is given for all $n$. Whilst on the right-hand side we consider a random graph (usually a tree), generated from some starting root $\rho$. The expectation is taken over the randomness of this graph, $\mathbb{E}_\mu[X] = \int_\Omega X(\omega) d\mu(\omega)$.*

*We write $G_n \xrightarrow{\mathbb{P}-loc} (G, \rho)$ and say that $G_n$ has local limit $(G, \rho)$ or that $G_n$ converges locally in probability to $(G, \rho)$ .*

**Random trees from population dynamics** We have already foreshadowed that the local limit of many of these graphs will be random trees. Here we describe properly what random trees are, in the terminology of Galton-Watson trees (GWT). These were originally invented for population dynamics. Therefore we refer to nodes $u$ as having a parent node $v$ which is the node closer to root node $\rho$. We also say $u$ has or generates children nodes, which are the nodes connected to them which are further from root $\rho$.

A Galton Watson process is a stochastic process $\{X_h\}$ where the starting population is one, $X_0 = 1$ and the population on time $h+1$, $X_{h+1}$, is calculated using

$$X_{h+1} = \sum_{i=1}^{X_h} Y_i^{(h)}. \tag{2.17}$$

Where $Y_i^{(h)}$ are i.i.d. random variables, independent of the time step $h$, which will be omitted. In other words, in one time step each individual gives birth to a $Y$ distributed amount of children and dies itself. Throughout this thesis the random variable $Y$ will be Poisson distributed with parameter $\lambda$. This is denoted as $Y \sim \text{Poi}(\lambda)$, for which we have

$$\mathbb{P}(Y = x) = \frac{\lambda^x e^{-\lambda}}{x!}. \tag{2.18}$$

This model can very naturally be extended to trees by drawing edges between each individual and their children.

**Definition 2.3.5** (Galton Watson Tree (GWT))**.** *Let $(T, \rho)$ be a distribution of rooted trees. Then $(T, \rho)$ is called a Galton-Watson tree with offspring $Y$, denoted $\mathrm{GWT}(Y)$, when a realization of $(T, \rho)$, is generated by taking a root $\rho$ and letting every node generate a $Y$ distributed number of children, which are connected with their parent.*

This definition is connected to the Galton Watson process, as the number of nodes at height $h$ of the tree is equal to $X_h$.

The extinction probability of the Galton-Watson process is

$$ex(X) = \lim_{n\to\infty} \mathbb{P}[X_n = 0]. \tag{2.19}$$

This is precisely the probability that the GWT is a finite tree. It greatly impacts the behaviour of community detection when $ex(X) < 1$, as this means that there are realizations which are infinite trees. We are interested in such graphs, as we shall see that statistical physics model have a phase transition which can be related to community detection on such trees 3.3. For $\mathbb{E}[Y] > 1$ we find that $ex(X) < 1$. When $\mathbb{E}[Y] \leq 1$ and $Y \neq 1$ we find that this probability is $ex(X) = 1$. In physics this survival probability can be identified as the probability of percolation.

GWT's turn out to be the local limit for the ER graph as we show in the next section. However, for the graphs with community structure the local limit also has this community structure. Therefore we introduce multi-type GWT's, where each vertex $u$ is assigned a type $t_u \in [q]$ and the distribution of children $Y_{t_u,i}$ is dependent on both the type of vertex $u$, $t_u$, as the type of the produced children, $i$.

As we want to be able to express the local limit of marked objects, we also introduce the marked multi-type GWT, where each vertex keeps track of their type.

**Definition 2.3.6** ((marked) multitype GWT's)**.** *Let $(T, \rho, \vec{t_\rho})$ be a distribution of rooted trees. Then $(T, \rho, \vec{t_\rho})$ is called a multitype Galton-Watson tree with $q \times q$ offspring matrix $Y$, denoted $\mathrm{GWT}(Y, \vec{t_\rho})$, when a realization of $(T, \rho, \vec{t_\rho})$, is generated by taking a root $\rho$ and giving it type $i$ with probability $t_{\rho,i}$. Then let every node generate children. A node of type $i$ generates a $Y_{i,j}$ distributed number of children of type $j$, which are connected with their parent. We call this multi-type Galton Watson tree marked if the type of every vertex is stored.*

We are interested in the extinction and survival of multitype GWT's, as statistical physics models have interesting properties once there is a survival probability $> 0$. Van der Hofstad [5] proved the following about the survival probability.

**Theorem 2.3.7.** *Let $\mathrm{GWT}(Y, \vec{n})$ be a multitype GWT with $n_i$ the fraction of nodes of type $i$. Let $c_{i,j}$ denote the expected offspring of type $j$ of a single individual of type $i$, divided by $n_i$, such that $K$ is the matrix with expected offsprings,*

$$K_{ij} = \mathbb{E}[Y_{i,j}] = c_{i,j} n_j. \tag{2.20}$$

*Let $\zeta$ be the survival probability of $\mathrm{GWT}(Y, \vec{n})$. Then the largest eigenvalue of $K$, $||K||$ can be computed and we have $\zeta > 0$ precisely when $||K|| > 1$.*

**The local limit of the IRG** We have almost all the ingredients to be able to give the local limit of the IRG. We continue with the definition of a graphical and irreducible kernel, before showing this local limit. These definitions are as introduced by [5], but simplified for our case where the set of types is $\mathcal{S} = [q] = \{1, ..., q\}$. Reducibility of $c$ in the $\mathrm{IRG}_n(c)$ roughly means that $c$ can be split into two smaller matrices $c_1$ and $c_2$, as there are guaranteed to be no connections between these two sets of types. Therefore we could have equally well started with two separate random graphs with two kernels of smaller dimension. As we can always create reducible random graphs by combining multiple irreducible ones, we will look at irreducible random graphs only. We discuss graphicality after its definition.

**Definition 2.3.8** (Graphical and irreducible kernels)**.** *A kernel $c$ (a $q \times q$ matrix defining the connecting probabilities of the IRG from section 2.2.2) is graphical if the following conditions hold:*

1. *(a)*

$$\sum_{i,j\in[q]} c_{i,j} n_i n_j < \infty \tag{2.21}$$

*(b)*

$$\frac{1}{n}\mathbb{E}[|E(\mathrm{IRG}_n(c))|] \to \frac{1}{2}\sum_{i,j\in[q]} c_{i,j}n_i n_j \tag{2.22}$$

*And a sequence $(c_n)_{n\geq 1}$ of kernels is called graphical with limit $c$ when $c_n(i,j) \to c(i,j)$, where $c$ satisfies $(a)$ as above, and*

$$\frac{1}{n}\mathbb{E}[|E(\mathrm{IRG}_n(c_n))|] \to \frac{1}{2}\sum_{i,j\in[q]} c_{i,j}n_i n_j \tag{2.23}$$

2. *A kernel $c$ is called reducible if $\exists A \subseteq S$ with $0 < \mu_{\mathcal{S}}(A) < 1$ such that $c = 0$ almost everywhere on $A \times (S \setminus A)$, otherwise $c$ is irreducible.*

As discussed before irreducibly simply means that $c$ is really a $q \times q$ martix and cannot be written as two smaller matrices $c_1$ and $c_2$. We now discuss the graphicality. The assumptions in (2.21) and (2.22) imply that the expected number of edges is proportional to $n$, and that the proportionality constant is precisely equal to $\sum_{i,j\in[q]} c_{i,j}n_i n_j$. In the case of a constant kernel $c$, assumption (2.22) is satisfied. In that case every node of type $n_i$ generates precisely $\sum_j \in [q]\frac{c_{i,j}}{n}n_j(n-1)$ edges, but every edge can be generated from 2 nodes but will only be generated once, so this is divided by 2. Note that if $c$ is graphical then we also have that $\mathrm{IRG}_n(c)$ is sparse.

We extend the setting to n-dependent sequences $(c_n)_{n\geq 1}$ of kernels in (2.23), as in some natural cases the kernels do indeed depend on $n$, which is to say that only when $n \to \infty$ do the group probabilities stop fluctuating.

All these definitions allow us to understand the following theorem from [5] about the local limit of the IRG, which we rewrite slightly for our purposes.

**Theorem 2.3.9** (Locally tree-like limit $\mathrm{IRG}_n(c_n)$)**.** *Assume that $c_n$ is an irreducible graphical kernel converging to some limiting kernel $c$. Then $\mathrm{IRG}_n(c_n)$ converges in probability in the local sense to the multi-type Galton-Watson tree, where*

- *the root has type distribution*

$$\vec{t_\rho} = \{n_1, ..., n_q\}. \tag{2.24}$$

- *a vertex of type $i$ has offspring distribution $\mathrm{Poi}(\lambda(i))$, with*

$$\lambda(i) = \sum_{j\in[q]} c_{i,j}n_j. \tag{2.25}$$

*Each of its offspring receives an independent type with distribution $Q(i)$ given by*

$$\mathbb{P}(Q(i) = j) = \frac{c_{i,j}n_j}{\sum_{j\in[q]} c_{i,j}n_j}. \tag{2.26}$$

**Arithmetic with Poisson distributed random variables** We would like to be able to write this in terms of an offspring matrix $Y$. We can express $Y_{ij}$, the offspring an individual of type $i$ generates whose children have type $j$, as an experiment which is done $\mathrm{Poi}(\lambda(i))$ times independently and has success probability $\mathbb{P}(Q(i) = j)$. A binomial distribution, with $n$ trials and success probability $p$, denoted as $\mathrm{Bin}(n,p)$, calculates precisely that distribution. Remember it is defined as

$$\mathbb{P}(\mathrm{Bin}(n,p) = x) = \binom{n}{x}p^x(1-p)^{n-x}. \tag{2.27}$$

Thus we find

$$Y_{ij} = \mathrm{Bin}(\mathrm{Poi}(\lambda(i)), \mathbb{P}(Q(i) = j)). \tag{2.28}$$

We will rewrite this formula as a Poisson distributed variable. For this we prove the following lemma.

**Lemma 2.3.10.** $\mathrm{Bin}(\mathrm{Poi}(a+b), \frac{a}{a+b})$ *and* $\mathrm{Poi}(a)$ *have the same distribution.*

*Proof.* The probability generating function (defined in (2.6)) of $\mathrm{Bin}(n, p)$ is $(pz + (1 - p))^n$. The probability generating function of $\mathrm{Poi}(\lambda) = \exp\{\lambda(z - 1)$. The probability generating function of $\mathrm{Bin}(\mathrm{Poi}(a + b), \frac{a}{a+b})$, is

$$\begin{aligned} G_X(z) &= \sum_{k=0}^{\infty}(\frac{a}{a+b}z + \frac{b}{a+b})^k \frac{(a+b)^k \exp\{-(a+b)\}}{k!} \\ &= \exp\{-(a+b)\}\sum_{k=0}^{\infty}\frac{((\frac{a}{a+b}z + \frac{b}{a+b})(a+b))^k}{k!} \\ &= \exp\{-(a+b)\}\exp\{az+b\} = \exp\{az-a\} = \exp\{a(z-1)\} \end{aligned} \tag{2.29}$$

which is the probability generating function of $\mathrm{Poi}(a)$. □

Now we can use lemma 2.3.10 with $a = c_{i,j}n_j$ and $b = \sum_{j \neq i \in [q]} c_{ij}n_j$ to show that equations (2.25) and (2.26) correspond to offspring matrix $Y$ with

$$Y_{ij} = \mathrm{Bin}(\mathrm{Poi}(\lambda(i)), \mathbb{P}(Q(i) = j)) = \mathrm{Bin}(\mathrm{Poi}(a+b), \frac{a}{a+b}) = \mathrm{Poi}(a) = \mathrm{Poi}(c_{i,j}n_j). \tag{2.30}$$

In the following paragraph we show what this implies for the local limit of $\mathrm{ER}_n(\lambda/n)$, $\mathrm{SBM}_n(q, a, b)$ and $\mathrm{DSBM}_n(q, a, b, \nu)$.

**The Local limit of the ER, the SBM and the DSBM** In $\mathrm{ER}_n(\lambda/n)$ we have the situation as described in the $IRC_n(c)$, with $\mathcal{S} = \{1\}$ and $c(1, 1) = \lambda$. Thus we find a that $\mathrm{ER}_n(\lambda/n)$ has as local limit a (single type) GWT where every vertex has offspring $\mathrm{Poi}(\lambda)$,

$$GWT(\mathrm{Poi}(\lambda)). \tag{2.31}$$

For the SBM we have $\mathrm{SBM}_n(q, a, b) = \mathrm{IRG}_n(c)$ with kernel $c$ as in (2.14). Thus we find for a SMB with $n_i = \frac{1}{q}$ for $i \in [q]$, that $\lambda(i) = \frac{a+(q-1)b}{q} = d$ for all $i$. Then for the local limit of a SBM we then find a multi-type GWT with

1. root $\rho$ has type distribution $\vec{t}_\rho = (\frac{1}{q}, ..., \frac{1}{q})$ and offspring distribution $\mathrm{Poi}(d)$.

2. a vertex has offspring $\mathrm{Poi}(d)$ with type distribution $Q(i)$

$$\mathbb{P}(Q(i) = j) = \begin{cases} \frac{a}{a+(q-1)b} & \text{for } i = j \\ \frac{b}{a+(q-1)b} & \text{for } i \neq j. \end{cases} \tag{2.32}$$

Which corresponds to

$$Y_{i,j} = \begin{cases} \mathrm{Poi}(\frac{ad}{a+(q-1)b}) = \mathrm{Poi}(\frac{a}{q}) & \text{for } i = j \\ \mathrm{Poi}(\frac{bd}{a+(q-1)b}) = \mathrm{Poi}(\frac{b}{q}) & \text{for } i \neq j. \end{cases} \tag{2.33}$$

Where a marked SMB has a local limit the same multi-type GWT, but then marked.

Gulikers et al. [4] show what the local limit is of the degree-corrected SBM (defined in section 2.2).

**Theorem 2.3.11** (local limit DSBM). $\mathrm{DSBM}_n(q, a, b, \nu)$ *converges in the local sense in probability to the multitype GWT, where*

1. *root $\rho$ has type distribution $\vec{t}_\rho = (\frac{1}{q}, ..., \frac{1}{q})$ and weight $\phi_\rho$ distributed as $\nu$.*

2. *every vertex $u$ has $\mathrm{Poi}(\frac{a+(q-1)b}{q}\Phi^{(1)}\phi_u)$ offspring with type distribution $Q_i$ as in (2.32) and size-biased weight distribution*

$$\nu^*([0, x]) = \frac{1}{\Phi^{(1)}}\int_{\phi_{\min}}^{x} y d\nu(y). \tag{2.34}$$

## 2.4 Giant component

An interesting property of graphs is their connectivity. In particular for the limit $n \to \infty$ we look at when there is a nonzero fraction of the nodes $\zeta$ in the largest connected component. When there is, we call this largest connected component the *giant*. In physics the existence of a giant is called percolation and *zeta* is often plotted as the probability of a node being inside the giant to find the threshold for percolation. For us it is necessary to quantify this behaviour, as the statistical physics models that we use behave differently when $\zeta > 0$, portraying a phase transition. More information on this will be given in section 3.3.

We will see that this is related to the extinction of the local limit. [5] prove the following theorem, which couples the fraction of nodes inside the giant to the survival probability of the local limit of the graph, also denoted $\zeta$.

**Theorem 2.4.1** (Relation giant and local limit)**.** *Let $G_n = (V(G_n), E(G_n))$ denote a finite (possibly disconnected) random graph. Let $u, v \in V(G_n)$ vertices and let $\mathscr{C}(u)$ denote the connected component of $u$. Let $u \nleftrightarrow v$ denote that there is no possible path using edges $e \in E(G_n)$ from $u$ to $v$, i.e. $u$ and $v$ are not in the same connected component. Assume that $G_n \xrightarrow[\mathbb{P}-loc]{} (G, \rho)$ having distribution $\mu_G$. Assume that*

$$\lim_{k\to\infty} \limsup_{n\to\infty} \frac{1}{n^2} \mathbb{E}[\#\{(u,v) \in V(G_n) : |\mathscr{C}(u)|, |\mathscr{C}(v)| \geq k, u \nleftrightarrow v\}] = 0. \tag{2.35}$$

*This means that there are no multiple separate giants.*

*Then, with $\mathscr{C}_{\max}$ and $\mathscr{C}_{(2)}$ denoting the largest and second largest connected components (with ties broken arbitrarily),*

$$\frac{|\mathscr{C}_{\max}|}{n} \xrightarrow{\mathbb{P}} \zeta = \mu_G(|\mathscr{C}_\rho| = \infty), \qquad\qquad \frac{|\mathscr{C}_{(2)}|}{n} \xrightarrow{\mathbb{P}} 0. \tag{2.36}$$

This means that when we can prove that vertices are likely to be connected, the size of the giant and the survival of the local limit are connected. In other words, we can identify the survival probability of the local limit $\zeta$ 2.3.7, with the probability of the random root $\rho$ being chosen inside the giant. Therefore a giant exists when $\zeta > 0$.

Thus we look back at the survival probabilities of the local limits of our described model, to see when they have a giant component.

**Lemma 2.4.2** (Existence of giant for ER, SBM, DSBM, IRG)**.** *Let us exclude the case every individual has precisely expected offspring of 1. Then the following holds.*

1. *For the* ER$(\lambda/n)$ *graph we use that $\zeta > 0 \Leftrightarrow \mathbb{E}[Y] > 1$ and find*

$$\zeta > 0 \Leftrightarrow \mathbb{E}[\mathrm{Poi}(\lambda)] > 1 \Leftrightarrow \lambda > 1. \tag{2.37}$$

2. *For the* SBM$(q, a, b)$ *we have $\lambda(i) = \frac{a+(q-1)b}{q}$ for all types $i$, hence*

$$\zeta > 0 \Leftrightarrow \mathbb{E}\left[\mathrm{Poi}\left(\frac{a+(q-1)b}{q}\right)\right] > 1 \Leftrightarrow a + (q-1)b > q. \tag{2.38}$$

3. *For the* DSBM$(q, a, b, \nu)$ *similarly we find*

$$\zeta > 0 \Leftrightarrow \mathbb{E}\left[\mathrm{Poi}\left(\frac{a+(q-1)b}{q}\Phi^{(1)}\phi_u\right)\right] > 1 \Leftrightarrow (a + (q-1)b)\,\Phi^{(2)} > q. \tag{2.39}$$

4. *Finally for the* IRG$_{\{n_1,\dots,n_q\}}(c_{ij})$ *we use theorem 2.3.7 and find that*

$$\zeta > 0 \Leftrightarrow ||K|| > 1 \tag{2.40}$$

*Where $||K||$ is the largest eigenvalue of $K_{ij} = c_{i,j} n_j$.*

## 2.5 Successful community detection

In this section we define what it means to successfully detect communities in different contexts. Successful community detection is in literature also referred to as successful reconstruction of communities. We look at community detection on random graphs, where the goal is to correctly reconstruct the original type assignment. Also on rooted trees community detection is useful, as these are the local limits of random graphs, On trees the goal will be to reconstruct the type of the root.

We say communities have not been detected (successfully), when the graph yields no information about the communities. This can still be dependent on the type of field. On an IRG this is when guessing that every node belongs to the largest type $i$ has the highest expected overlap with the original types. On a rooted tree this is when knowing the types of all the leaves, the best guessing method of the type of the root is guessing the largest type $i$. We will say communities have been (successfully) detected when there is a method which is strictly better than simply guessing the largest type. This may seem like a weak definition, as the guess could be only very slightly better than random. However, even when it is only slightly better than random, the fact that some configuration has higher probability than simply guessing the largest type means that that configuration contains information about the true communities. Hence we say that communities have been successfully detected. In practice we shall see that the threshold for community detection to be successful and the threshold for it being successful in linear (polynomial) time shall differ 7.2. As for large graphs exponential calculation takes too much time, there is thus a difference between successful community detection being possible and it being feasible, as we will see in Section 7.2.

**Definition 2.5.1** ((successful) community detection on random graphs)**.** *Let $G_n$ be a random graph which first assigns a type $i \in [q]$ to each vertex using probabilities $n_i$, resulting in original type assignment $\boldsymbol{t}^o$, before connecting edges based on these types. Let $\boldsymbol{t}^*$ be a guess of the assignment. In order to be able to check the quality of this guess, we introduce the overlap*

$$Q(\boldsymbol{t}^o, \boldsymbol{t}^*) = \max_{\pi} \frac{\frac{1}{n}\sum_u \delta_{t_u^o, \pi(t_u^*)} - max_i n_i}{1 - max_i n_i}, \tag{2.41}$$

*with $\pi$ a permutation over $[q]$, the types. For finite $n$, communities have been detected when for some guess $\boldsymbol{t}^*$ we have $\mathbb{E}[Q(\boldsymbol{t}^o, \boldsymbol{t}^*)] > 0$. For $n \to \infty$, we say communities have been detected if there is a sequence of guesses $\{\boldsymbol{t}^*\}_{n \geq 1}$ such that*

$$\mathbb{P}(Q(\boldsymbol{t}_n^o, \boldsymbol{t}_n^*) > 0) \to 1. \tag{2.42}$$

We show in chapter 6 that in the belief propagation algorithm of Decelle [1], the quality of a resulting guess can be checked without using $\mathbf{t}^o$. Thus this 'best guess' is uniquely defined.

**trees** In Section 2.3 we have seen that the graph models we investigate are locally treelike. On trees one can also define community detection. Here we call community detection successful if the root $\rho$ can be guessed better than simply guessing the largest type, based on the types at any distance from $\rho$.

**Definition 2.5.2** ((successful) community detection on random trees with equal group sizes (free boundary, + boundary))**.** *Let $(T, \rho)$ be a realization of $GWT(Y, \vec{t_\rho})$ with $\vec{t_\rho} = (\frac{1}{q}, ..., \frac{1}{q})$. Let $T_h$ denote the nodes at distance $h$ from $\rho$. Then we say communities have been detected successfully if there is a type $i$ with*

$$\inf_{h \geq 1} \mathbb{E}[\mathbb{P}(t_\rho = i | \boldsymbol{t}_{T_h})] > \frac{1}{q}. \tag{2.43}$$

*With no added constraints we call this the free boundary community detection. When one conditions on $\mathbf{t}_{T_h} = \mathbf{+}$ then it is called a + boundary condition. When no boundary is mentioned, we refer the free boundary case.*

This means that no matter at what distance of $\rho$ we are given the types of the nodes at this distance, we are always able to extract some information about the type of $\rho$, as there is a type for which it has probability higher than $\frac{1}{q}$.

## 2.6 Broadcasting process

In this section we look at community detection on trees. In particular we look at successful community detection in a broadcasting process on trees, described by Evans et al [2]. This process assigns communities to every node in the tree.

The broadcasting process can be applied to a rooted tree $(T, \rho)$. Then the broadcasting process first assigns a type $\sigma_\rho \in \mathcal{S} = \{-1, 1\}$ uniformly at random to the root. We use the symbol $\sigma$ as later these types will coincide with the spins from the Ising model, which are also denoted $\sigma$. The children of $\rho$ are given the same type as $\rho$ with probability $1 - \epsilon$ and the other type with probability $\epsilon$, with $0 < \epsilon < \frac{1}{2}$. An edge which connects nodes with different type is called a flip, and $\epsilon$ is called the flipping probability.

This assignment of types can also be done by labeling every edge $(u, v) \in E(T)$ with a random variable $\eta_{(u,v)}$. For each edge, let

$$\begin{aligned} \mathbb{P}(\eta_{(u,v)} = -1) &= \epsilon \\ \mathbb{P}(\eta_{(u,v)} = 1) &= 1 - \epsilon. \end{aligned} \tag{2.44}$$

Using this we can calculate the type of each vertex $u$ using

$$\sigma_u = \sigma_\rho \prod_{(u',v')} \eta_{(u',v')}, \tag{2.45}$$

where the product is over all edges $(u', v')$ on the path from $\rho$ to $u$. We let $\{\eta_e\}$ denote the set of random variables labeling all edges $e \in E(T)$. Then we define $\mathbb{P}_\eta$ as the probability measure that assigns a probability to each state according to

$$\mathbb{P}_\eta(\boldsymbol{\sigma}) = \mathbb{P}(\sigma_\rho)\mathbb{P}\left(\{\eta_e\} : \sigma_u = \sigma_\rho \prod_{(u',v')} \eta_{(u',v')} \forall u \in V(T)\right). \tag{2.46}$$

We call this a broadcasting process. As the distribution of $\eta$ (2.44) is independent of the type $\sigma_u$, we can express $\mathbb{P}_\eta$ of a state solely in terms of the number of flips in that state. Let $k$ be the number of flips in $\boldsymbol{\sigma}$ and let $n = |V(T)|$. An added bonus of the identification of a flip with $\eta_e = -1$ and no flip with $\eta_e = +1$, satisfying (2.45), is that we have

$$\sum_{e \in E(T)} \eta_e = (n - k) - k = n - 2k \tag{2.47}$$

$$k = \frac{1}{2}\left(n - \sum_{e \in E(T)} \eta_e\right). \tag{2.48}$$

Then $\mathbb{P}_\eta$ can be simplified as

$$\mathbb{P}_\eta(\boldsymbol{\sigma}) = \frac{1}{2}\epsilon^k(1 - \epsilon)^{n-k}. \tag{2.49}$$

The critical conditions for successful community detection are coupled with the branching factor $\bar{v}$. This branching factor can be thought of as the expected number of branches each vertex creates. On GWT($Y$) we can define the branching factor

$$\bar{v} \equiv \mathbb{E}[Y]. \tag{2.50}$$

Then Evans et al [2] show the following theorem.

**Theorem 2.6.1** (Succesful community detection on trees)**.** *Community detection is successful for* $1-2\epsilon > \bar{v}^{-1/2}$ *and unsuccessful for* $1 - 2\epsilon < \bar{v}^{-1/2}$.

Where successful community detection on trees was defined in 2.5.2. We later try to compare these results to the results of community detection on graphs.

**q types** For our own purposes we extend the broadcasting process on $(T, \rho)$ to $q$ types. In this case the broadcasting process first assigns a type $t_\rho \in [q]$ to the root uniformly at random. For $\{\eta_e\}$ we can define this in multiple ways. An intuitive way is to define a flipping matrix $A$ where $A_{ij}$ is the probability of an edge $(u, v)$ with $t_u = i$ flipping to type $j$. This matrix is then

$$A = \begin{pmatrix} \ddots & & \boldsymbol{\epsilon} \\ & 1 - (q-1)\epsilon & \\ \boldsymbol{\epsilon} & & \ddots \end{pmatrix}. \tag{2.51}$$

However, we wish to keep identification 2.45, where each $\eta_{(u,v)}$ can be seen as a multiplication. Thus we define

$$\mathbb{P}(\eta_{(u,v)} = \frac{i}{t_u}) = \begin{cases} 1-(q-1)\epsilon & \text{for } i = t_u \\ \epsilon & \text{for } i \neq t_u. \end{cases} \tag{2.52}$$

This has the same distribution as $A$, and the type of any node in $T$ can still be calculated using (2.45). We let $\mathbb{P}_\eta$ be defined as in (2.46) and call it a $q$-type broadcasting process. Even though the $\eta_{(u,v)}$ are now dependent on the $t_v$, we show in section 4.4 that $\mathbb{P}_\eta(\mathbf{t})$ can again be simplified in terms of $n = |V(T)|$ and the number of flips in $\mathbf{t}$, $k$, as

$$\mathbb{P}_\eta(\mathbf{t}) = \frac{1}{q}\epsilon^q(1-(q-1)\epsilon)^{n-k}. \tag{2.53}$$

This is less strong as in the 2-type case, as now the number of flips $k$ can no longer easily determined by a sum and $n$ as in (2.47). Instead it has to be counted manually as

$$k = \#\{\eta_e | \eta_e \neq 1\}. \tag{2.54}$$

We have now seen some random graph models, and how communities can be detected on them. We have seen that these models are locally treelike. We have therefore also looked at community detection on trees. We also described the giant component for these graph models, which is necessary for the behaviour of the statistical physics models of the next Chapter.

# Chapter 3

# Statistical physics background

Ising and Potts models are models created by physicists in order to better understand phenomena which include magnetism and spin in quantum mechanics. Magnetic systems are at the core lattices where every node has either spin $+1$ or spin $-1$. In ferromagnets, the fact that a node has spin $+1$ will increase the probability of other nodes to also have spin $+1$. However, the interaction strength of this influence rapidly decays with the distance. Therefore the Ising models approximates this interaction by only taking neighbours of the lattice into account. When the temperature $T$ is constant, the Ising model assigns a probability to every state of the entire system, using the Boltzmann distribution. The Potts model is a generalization of the Ising model where there are $\geq 3$ different types of spin.

These models can easily be applied to graphs, where every vertex is seen as a node in the lattice, and an edge between two nodes signifies that they are neighbours. We can identify each spin type as a community. This allows us to relate these models to community detection, as the expected spin of a node is now the probability of it belonging to that community. In physics, there are three types of systems which can be described using the Ising model. In the ferromagnetic case the only interaction is that a particle is more likely to have the same type as their neighbours. This corresponds to the ferromagnetic community detection of the native language of people in the graph of the Belgian phone network. In the antiferromagnetic case a particle is less likely to have the same type as their neighbours, which corresponds to the community detection of sex in the graph of marriages. In a glassy system each edge has its own interaction strength which determines both the strength of the interaction, as well as if it is ferromagnetic or antiferromagnetic. Few exact results are known about the behaviour of glassy systems. The connection with community detection here is for example the political vote of people in the USA, where the graph is people who are connected when they know each other on Facebook, and the interaction strength is determined by the type of relation (i.e. friend, colleague, family, etc). We mainly investigate the ferromagnetic case.

In this chapter we start by introducing the Ising model on graphs. We then show what physical quantities are used for quantifying an Ising model in physical problems. We then look at when there is a phase transition, which is when the change of a single spin affects a positive fraction of all other spins. This phase transition is important for us, as one of the main points of this thesis is that it is exactly the threshold for community detection on graphs.

We then show the Potts model, which extends the Ising model to $\geq 3$ types. Much less is known about the phase transition in Potts models, but we present the results thus far. The reason behind the lack of knowledge will become clear in chapter 6.

## 3.1 Ising model

The Ising model is a model designed for magnetic states on grids. Every node has a spin, $\sigma_u \in \{+1, -1\}$. The model assigns probabilities of all states based on how many spins align with their neighbours. The type of system is determined by the coupling constant $J$. For $J > 0$ the system is ferromagnetic and for $J < 0$ it is antiferromagnetic and for edge-dependent $J_{uv}$ it is called a spin-glass. We only consider the ferromagnetic case and therefore look at positive $J$. The direct interaction of non-neighbours is neglected as these interactions decay very rapidly with distance. We look at this model at constant inverse temperature $\beta$. Furthermore there is a magnetic field which incorporates some external preference for one of the spins. This model can be defined identically on graphs. For this thesis we use the graph notation vertices and types indistinguishable with nodes and spins.

The model is based on the universal Boltzmann relation from thermodynamics of a system with constant temperature which can transfer energy with a large thermal reservoir (its surroundings). In this case the probabilities of each state can be described as

$$\mu_\beta(\boldsymbol{\sigma}) = \frac{1}{Z}\exp\{-\beta E(\boldsymbol{\sigma})\} \tag{3.1}$$

where $E$ is the energy of a state, and $Z$ is the called the partition sum, which normalizes $\mu$. In this setting $E$ is called the Hamiltonian, and is then usually denoted with $H$. We will also use the Hamiltonian notation to avoid confusion with the edge set $E(G)$. These systems have been studied thoroughly, and we will make use of quantities for this system in section 3.2. In the Ising model setting we define the energy

$$H(\boldsymbol{\sigma}) = -\left(\sum_{(u,v)\in E} J\sigma_u\sigma_v + B\sum_{u\in V}\sigma_u\right) \tag{3.2}$$

with $B \in \mathbb{R}$ the magnetic field and $J$ the the coupling constant. The magnetic field $B$ signifies a bias towards a type, where in this case a positive $B$ is a bias towards the type $\{+1\}$ and a negative $B$ a bias towards $\{-1\}$. The sign of the coupling constant $J$ determines if the behaviour is ferromagnetic or anti-ferromagnetic. We work with ferromagnetic $J > 0$, for which we can take $J = 1$ because we can always rescale $\beta$ and $B$ to Erdös–Rényi. For spin glasses the coupling constant is also dependent on the specific edge, so $J_{uv}$. This model we will slightly adjust and use in chapter 4.

An Ising model of two types on a finite graph $G$ is thus defined as follows. Let $u \in V$ be a vertex, then $\sigma_u \in \mathcal{S} = \{+1, -1\}$ is the type of $u$. Every state $\boldsymbol{\sigma} \in \Omega = \mathcal{S}^n$ containing the types of all $u \in V$ has a probability

$$\mu(\boldsymbol{\sigma}) = \mu_{\beta,B}(\boldsymbol{\sigma}) = \frac{1}{Z(\beta,B)}\exp\{\beta\left(\sum_{(u,v)\in E}\sigma_u\sigma_v + B\sum_{u\in V}\sigma_u\right)\}. \tag{3.3}$$

At $\beta = 0$, the $T \to \infty$ limit, every state has an equal probability. As $\beta$ grows, states where the type of a vertex aligns with the type of its neighbours have a growing probability.

We introduce an extension to the Ising model with a magnetic field at every vertex, $B_u$. We do this because later we can show that probabilities of this model can be calculated iteratively on trees. Let $\mathbf{B} = \{B_1, ..., B_n\}$. In this model each state $\boldsymbol{\sigma}$ has a probability

$$\mu(\boldsymbol{\sigma}) = \mu_{\beta,\mathbf{B}}(\boldsymbol{\sigma}) = \frac{1}{Z(\beta,\mathbf{B})}\exp\{\beta\left(\sum_{(u,v)\in E}\sigma_u\sigma_v + \sum_{u\in V}B_u\sigma_u\right)\}. \tag{3.4}$$

The partition function $Z$ has many practical uses in statistical physics because it contains all necessary information to derive some thermodynamic quantities. We show these quantities in the next section.

## 3.2 Statistical Physics notation

Models with a Boltzmann measure have been thoroughly studied for physical applications. We can use the quantities and notation devised for these studies on graphs as well, and will introduce them here.

One quantity that helps us quantify a system in statistical physics is the *internal energy* $U$. This is nothing more than the expected energy of the system. In the Ising model this corresponds to the expected probability of a state

$$U \equiv \mathbb{E}[E(\mathbf{t})] = \sum_{\mathbf{t}}\mu(\mathbf{t})E(\mathbf{t}). \tag{3.5}$$

Another quantity that can help us is the entropy of a measure,

$$S_\mu \equiv -\sum_{t\in\mu}\mu(t)\log(\mu(t)). \tag{3.6}$$

Entropy is a measure of uncertainty. It tells us how certain we are about which state we will encounter. The higher the entropy, the less information we have on which state a random variable will get. A very intuitive example is the Boltzmann distribution (3.4) we have just seen. As $\beta$ grows, states with more clustering of types have a much higher probability, thus increasing the knowledge about which states we are likely to encounter, decreasing the entropy.

The third quantity we discuss is the *free energy* of the system. In Boltzmann distributions (3.1) we get

$$\begin{aligned} F_\mu \equiv -\frac{1}{\beta}(S_\mu - \beta U_\mu) &= -\frac{1}{\beta}\sum_{\mathbf{t}} \mu(\mathbf{t})(-\log(\mu(\mathbf{t})) - \beta E(\boldsymbol{\sigma})) \\ &= -\frac{1}{\beta}\sum_{\mathbf{t}} \mu(\mathbf{t})(\log(Z) + \beta E(\boldsymbol{\sigma}) - \beta E(\boldsymbol{\sigma})) \\ &= -\frac{1}{\beta}\sum_{\mathbf{t}} \mu(\mathbf{t})(\log(Z)) = -\frac{1}{\beta}\log(Z). \end{aligned} \tag{3.7}$$

Calculating the partition function can be computationally hard. We will see that we can calculate the internal energy $U$ and the entropy $S$ computationally easy using iterative calculation on trees. We can extend this to an approximation using the same methods on graphs, allowing us to approximate $\log(Z)$.

The free energy is so important because physical systems will minimize this quantity at constant temperature (i.e. constant $\beta$). Similarly for our purposes, we will be able to distinguish which guess of communities is thermodynamically favoured by having the lower free energy, thus being the better guess.

## 3.3 Phase transition

### Phase transition

A phase transition has this name because it stems from a transition of one phase to the other, for example solid to liquid. In physical systems, the reason for this is that the free energy of one state surpasses the other for some critical inverse temperature $\beta_c$, therefore the system will choose the state with the lowest free energy. Usually this transition is accompanied by a discontinuity in the $i$'th derivative of the free energy. We then call this an $i$'th order phase transition.

The free energy (3.7) is an extensive quantity i.e. it scales with the size of the system. We therefore introduce the *free energy density*

$$\phi_n(\beta, B) = \frac{1}{n}\log(Z_n(\beta, B)). \tag{3.8}$$

We see that for finite $n$ the free energy density is the logarithm of a sum of exponentials, which is analytic and therefore will have no singularities, thus no phase transition. In order to obtain a phase transition we therefore need to apply the Ising model to a graph which has a giant component as $n \to \infty$. As we have seen in section 2.4, this is true when $|\mathscr{C}_{\max}|/n \to \zeta > 0$. Note that if $\zeta = 0$ we only have finite connected components a.s. and therefore the Ising model on the entire graph can be reduced to the multiplication of the Ising models on these finite components, which have no phase transition.

Thus when a phase transition exists it will appear as a discontinuity of the limit of the free energy density of $n \to \infty$. We thus look at this limit

$$\lim_{n\to\infty} \phi_n(\beta, B) = \lim_{n\to\infty} \frac{1}{n}\log(Z_n(\beta, B)) = \phi^*(\beta, B). \tag{3.9}$$

For models with a Boltzmann measure, this limit exists, as the free energy is a so-called extensive quantity, meaning that it scales with the size of the system $n$.

First we look at phase transitions of the Ising model (3.4) on trees. According to the statistical physics derivation [7], the model (3.3) has a line of first-order phase transitions for $B = 0$ and $\beta > \beta_c$, (where specifically the continuous function $B \to \phi(\beta, B)$ has a discontinuous derivative). The critical temperature $\beta_c$ depends on the graph only through the average branching factor $\bar{v}$. This branching factor can be thought of as the expected number of branches each vertex creates, which is 1 lower than the degree of such a vertex. Thus let $v_k = \mathbb{P}(deg(u) = k)$ be the degree distribution of $(T, \rho)$, then we can define the branching factor

$$\bar{v} \equiv \sum_{k=1}^{\infty}(k-1)v_k. \tag{3.10}$$

Then the authors of [7] show that the critical temperature is determined by the condition

$$\bar{v}\tanh(\beta_c) = 1. \tag{3.11}$$

In the case of trees, $\zeta > 0$ corresponds to $\bar{v} > 1$. Thus we need $\bar{v} > 1$ as an additional condition for phase transition. Using this condition, we can see that $\beta_c$ is positive and well-defined.

We now look at the slightly more general case of locally treelike graphs, where the authors of [8] prove that the free energy density converges and we have the same phase transition. The main takeaway of the theorem below is that for the treelike graphs we look at, the free energy density exists even in the limit $n \to \infty$. Therefore we can look for a phase transition in this regime. However, the resulting limit has a very convoluted expression, even for the Ising model.

**Theorem 3.3.1** (convergence free energy on locally treelike graphs)**.** *Let $G_n$ be uniformly sparse (2.9) and $G_n \xrightarrow{\mathbb{P}-loc} (G, \rho)$ as in (2.16) with offspring distribution of the root $P = \{P_k : k \geq 0\}$. Let*

$$v_k = \frac{kP_k}{\sum_{l=1}^{\infty} lP_l} \tag{3.12}$$

*its size-biased version with finite first moment, with averages*

$$\bar{v} \equiv \sum_{k=1}^{\infty} (k-1) v_k \qquad\qquad \bar{P} = \sum_{k=0}^{\infty} kP_k. \tag{3.13}$$

*Let further the Ising model be given by equation (3.3).*

*Then for any $B \in \mathbb{R}$ and $\beta \geq 0$, the following limit exists.*

$$\lim_{n\to\infty} \frac{1}{n} \log[Z_n(\beta, B)] = \phi(\beta, B), \tag{3.14}$$

*where for $B > 0$ the limit is given by*

$$\phi(\beta, B) \equiv \frac{\bar{P}}{2} \log\cosh(\beta) - \frac{\bar{P}}{2} \mathbb{E}[\log(1 + \tanh(\beta)\tanh(h_1)\tanh(h_2))] \tag{3.15}$$

$$+ \mathbb{E}[\log\left(e^B \prod_{i=1}^{L} [1 + \tanh(\beta)\tanh(h_i)] + e^{-B} \prod_{i=1}^{L} [1 - \tanh(\beta)\tanh(h_i)]\right)] \tag{3.16}$$

*where $L$ has distribution $P_l$ and is independent of the "cavity fields" $h_i$ that are i.i.d. copies of the fixed point $h^*$ of*

$$h^{(t+1)} \overset{d}{=} B + \sum_{i=1}^{K-1} \operatorname{arctanh}[\tanh(\beta)\tanh(h_i^{(t)})] \tag{3.17}$$

*with $K$ distributed as $\bar{v}$. Also, $\phi(\beta, B) = \phi(\beta, -B)$ and $\phi(\beta, 0)$ is the limit of $\phi(\beta, B)$ as $B \to 0$.*

For general graphs no direct coupling to $\beta_c$ has been made and the only criterion we know is that we need $\zeta > 0$ for a phase transition.

## 3.4 Potts model

The Potts model is the Ising model extended to $q \geq 3$ types, so $\mathcal{S} = [q]$. A magnetic preference for each type $i$ is needed, so there are $q$ magnetic fields $B_i$. Then we define the energy of each state $\mathbf{t} \in \Omega$ as

$$H(\mathbf{t}) = -2\{\sum_{(u,v)\in E} J\mathbb{1}(t_u = t_v) + \sum_{u\in V}\sum_{i=1}^{q} B_i \mathbb{1}(t_u = i)\} \tag{3.18}$$

We multiply this by 2 in order to maintain the property that the Ising measure is the Potts measure with $q = 2$. Again we look at ferromagnetic systems and set $J = 1$ because we can rescale $\beta, B$. This leads to the probability measure

$$\mu(\mathbf{t}) = \mu_{\beta,\vec{B}}(\mathbf{t}) = \frac{1}{Z} \exp\{2\beta \left(\sum_{(u,v)\in E} \mathbb{1}(t_u = t_v) + \sum_{u\in V}\sum_{i=1}^{q} B_i \mathbb{1}(t_u = i)\right)\}. \tag{3.19}$$

As we did for the Ising model in (3.4), this model can be extended to have a magnetic field $B$ at every vertex. Then the probability of each state $\mathbf{t}$ becomes

$$\mu(\mathbf{t}) = \mu_{\beta,\vec{B}}(\mathbf{t}) = \frac{1}{Z(\beta,\vec{B})} \exp\{2\beta \left( \sum_{(u,v)\in E} \mathbb{1}(t_u = t_v) + \sum_{u\in V} \sum_{i=1}^{q} B_{ui}\mathbb{1}(t_u = i) \right) \}. \tag{3.20}$$

For the Potts model we also show some results for a slightly adjusted spin glass model. In the spin glass the interaction strength $J_{uv}$ is dependent on the edge $(u, v)$, so

$$\mu(\mathbf{t}) = \frac{1}{Z} \exp\{2\beta \left( \sum_{(u,v)\in E} J_{uv}\mathbb{1}(t_u = t_v) + \sum_{u\in V} \sum_{i=1}^{q} B_i\mathbb{1}(t_u = i) \right) \}. \tag{3.21}$$

Much less is known about spin glasses, but we show in section 4.6 that we can identify an adjusted spin glass Potts model on trees as a broadcasting process.

### Phase transition

For the Potts model there are also some exact results of phase transitions on trees. In [7] is shown that (3.19) also has a line of first-order phase transitions for $B = 0$ and $\beta > \beta_c$ The critical temperature depends on the graph only through the average branching factor $\bar{v}$ (3.10) and is determined by the condition

$$\bar{v}\frac{e^{\beta_c} - e^{-\beta_c}}{e^{\beta_c} + (q-1)e^{-\beta_c}} = 1 \tag{3.22}$$

with $\bar{v} > 1$ s.t. $\zeta > 0$. Notice that for $q = 2$ this reduces to the critical temperature for the Ising model (3.11).

In the Potts model much less is known about the exact phase transitions on more general graphs. However, we know that when there is a giant, so $\zeta > 0$, such phase transitions can arise.

In the next chapter we shall show that the Potts model introduces a community structure of $q$ types on the local limit of an $\mathrm{ER}_n(\lambda/n)$. This can be described the same way as a $q$-type broadcasting process on trees. This broadcasting process on the local limit of an $\mathrm{ER}_n(\lambda/n)$ can then be described as the local limit of a marked $\mathrm{SBM}_n(q, a, b)$. This way we will be able to couple the parameter of the Potts model, $\beta$ and the average degree $\lambda$ to the parameters $a$ and $b$.

## Chapter 4

# How Potts models impose a community structure on random graphs

We want to be able to compare statistical physics results of community detection using Ising and Potts models on graphs with community detection results on mathematical graph models. This could lead to new insights in either field that have been discovered in the other. It can also help to understand the models even better.

In this chapter we show that for very large graphs, in the limit $n \to \infty$, a comparison can be made. First we start with some intuition on the mapping of Ising models to trees and a broadcasting process in 4.1. In section 4.2 we show that there is an equivalence between the local limit of a 2-type marked SBM and an Ising model on the local limit of an ER graph. After proving this equivalence we couple the community detection results of the critical parameters found for successful reconstruction of communities on a tree found in both statistical physics and mathematics in section 4.3. We extend this equivalence to the local limit of a marked SBM with the local limit of an ER graph with a Potts model in section 4.4. In section 4.5 we show how far we can generalize the identification between the broadcasting process on the local limit of an ER graph with the marked local limit of an IRG. We shall see that this is only possible if the average offspring degree of the IRG is equal to $\lambda$ for all types. In section 4.6 we show how far we can generalize the identification of the Potts model as a broadcasting process on trees. We further show that introducing a magnetic field complicates matters and does not lead to the same equivalence using our method. We further extend all our previous results to degree-corrected graphs in section 4.7, before giving a small conclusion of all the results of this chapter in section 4.8.

## 4.1 Intuition Ising on trees

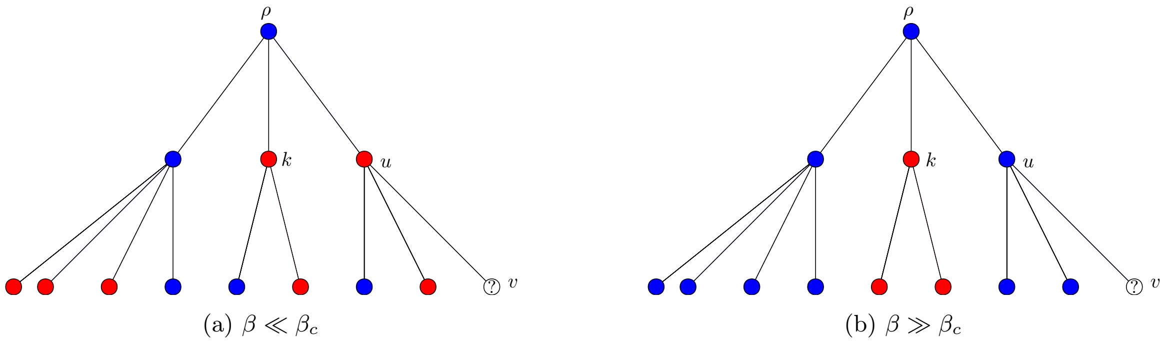
A mapping we will make is that an Ising model on a tree can be described as a broadcasting process. We will familiarize ourselves with an Ising model on a graph and show intuitively why and when such a mapping is possible. The rest of the chapter we make this intuition rigorous.

Let us apply an Ising model (3.3) with $B = 0$ to a small tree $T$, from now on referred to as our favourite small tree, in figure 4.1. Intuitively, for low $\beta$, all states have about equal probability and thus the system is dominated by states with low total magnetization $|m|$, $m = \sum_{u \in T} \sigma_u$, so about the same amount of both types. This is the case, as there are simply much more of such states than states with many of the same spin. As $\beta$ grows, the alignment of the spins becomes more important. For $\beta > \beta_c$ we have reached the point where alignment is so important, that knowing a single spin $\sigma_u$ gives us at least some information about all spins in $T$, no matter how far away. Thus the system is dominated by states where many spins align, for which the magnetization $|m|$ is high.

The calculation of the probability of a single state in the Ising model, equation (3.3), requires the computation of the partition function $Z$. For general graphs this requires the calculation of $2^n$ states. For small trees, such as our favourite $T$, this is completely fine. For trees with very large $n$ this computation becomes problematic. We therefore look for a way to calculate the probability of a state in linear time, using an iterative method.

When we look back at figure 4.1a, we can ask ourselves what the probability of the shown state is when $v$ is blue versus the probability that $v$ is red. Since $v$ only has one edge to $u$, which is blue, (3.3) shows that $\mu(\sigma_v = \text{blue}) = \exp(2\beta)\mu(\sigma_v = \text{red})$. This is only based on the type of $u$ and the fact that there is an edge $(u, v)$.

For non-leaves $k$ we can use a similar argument. Suppose we know $\sigma_\rho$ but nothing about the sub-tree of $k$



(a) $\beta \ll \beta_c$ (b) $\beta \gg \beta_c$

Figure 4.1: Our favourite small tree $T$ with a typical state from an Ising model on $T$ for $\beta \ll \beta_c$ (a) and $\beta \gg \beta_c$ (b). We discuss in the text ways to determine the type of $v$.

($k$ and all nodes connected to $k$ without $\rho$). Then for each configuration of the subtree of $k$ with $\sigma_k =$ blue we can flip all spins in the subtree of $k$ and find that the flipped state is precisely $\exp(2\beta)$ times less likely.

Knowing this, it feels intuitive that we can calculate the probability of a single state for $B = 0$. We can do this by simply taking $\mathbb{P}(\rho = \text{blue}) = \mathbb{P}(\rho = \text{red})$ and then working our way down every edge, multiplying with $\frac{1}{1+\exp(2\beta)}$ when the type flips and with $\frac{\exp 2\beta}{1+\exp(2\beta)}$ when the type stays the same. By using these fractions we indirectly compute $Z$. We shall see in Chapter 6 that on trees this calculation of $Z$ is exact and can be used to approximate $Z$ on treelike graphs.

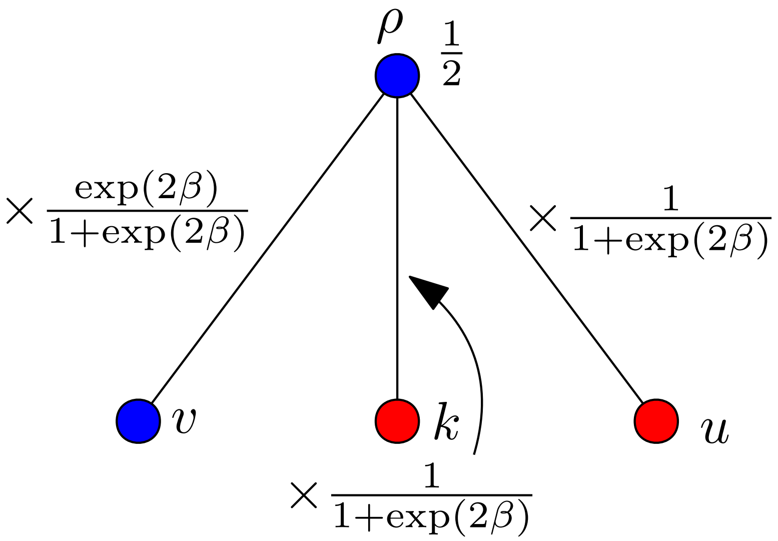

$\rho$
$\frac{1}{2}$
$\times \frac{\exp(2\beta)}{1+\exp(2\beta)}$
$\times \frac{1}{1+\exp(2\beta)}$
$v$
$k$
$u$
$\times \frac{1}{1+\exp(2\beta)}$

Figure 4.2: The iterative calculation of the probability of a state in the Ising model on a very small tree $T$. The state shown has probability $\frac{\exp(2\beta)}{2(1+\exp(2\beta))^3}$.

As there are only $n$ vertices, and every edge one node is updated, this only takes $n$ calculations, which is a huge improvement compared with the $2^n$ we had before.

We have intuitively seen that on any tree we can write the Ising model as an iterative calculation of the probabilities of the edges. Thus we can also do this on random (Galton-Watson) trees 2.3.5. In the rest of this chapter we investigate the Ising (and Potts) model to the local limits from section 2.3. In particular we apply the Ising model to the local limit of $\text{ER}(\lambda/n)$, a graph without communities. We show under which parameters and conditions this is equivalent to the local limit of a marked SBM.

## 4.2 Equivalence local limit stochastic block model and Ising model on local limit Erdös–Rényi graph

In this section we will show that there is an equivalence between two graph models. The first is the Ising model $\mu_\beta$ as defined in (3.3) on the local limit of an ER random graph with parameter $\lambda$, $\mathrm{GWT}(\mathrm{Poi}(\lambda))$ (see Section 2.3). We show this is equivalent with the local limit of a two-type marked SBM with parameters $a, b$ as defined in section 2.3, $T_2$.

Let us first go over the intuition behind comparing these models. In the SBM, vertices have a probability of $a/n$ to connect with vertices of equal type and $b/n < a/n$ to connect with vertices of the other type. In an Ising model $\beta$ imposes a preference of neighbours having equal spin, which we associate as their type. Higher $\beta$ means more chance of being connected to the same type. Higher $a$ and $b$ means a higher degree, for which we need to compentsate in the Ising model with a higher $\lambda$. The intuition is thus that we can express connecting probabilities $a, b$ in terms of some average degree $\lambda$ and a preference $\beta$. For this "average degree graph" we choose the ER graph, as it handles connecting to a neighbour similar to the SBM, namely by having a probability to connect with each other node separately. The reason we use the limit of $n \to \infty$ is because then the local limit of these objects become Galton-Watson trees (GWT's), which are trees, on which Ising (and Potts) models are easier to describe.

**Theorem 4.2.1.** *Let* $\mathrm{GWT}(\mathrm{Poi}(\lambda))$ *be the local limit of* $\mathrm{ER}_n(\lambda/n)$ *and let* $\mu_\beta$ *be an Ising model with* $B = 0$ *imposed on* $\mathrm{GWT}(\mathrm{Poi}(\lambda))$*. Let the marked Galton-Watson tree* $\mathrm{GWT}(Y, (\frac{1}{2}, \frac{1}{2}))$ *(as defined in* (2.3.5)*) with* $Y$ *as in* (2.33) *be the local limit of a marked* $\mathrm{SBM}(2, a, b)$*. Let* $\lambda = \frac{a+b}{2}$ *and* $\beta = \frac{1}{2}\log(\frac{a}{b})$*, then these two objects have equal distribution.*

This equivalence will be done in two steps. First we will identify an Ising model on a rooted tree $(T, \rho)$ as a broadcasting process on $(T, \rho)$ which is described in section 2.6. Secondly we will show that using this identification we can prove equivalence of the distribution of a broadcasting process on $\mathrm{GWT}(\mathrm{Poi}(\lambda))$ with the marked $\mathrm{GWT}(Y, (\frac{1}{2}, \frac{1}{2}))$.

First [2] show that a broadcasting process and an Ising model with $B = 0$ on $(T, \rho)$ have equal distribution, which we show in the following lemma.

**Lemma 4.2.2** (Equivalence Ising measure with $B = 0$ and probability measure broadcasting process on trees)**.** *Let* $(T, \rho)$ *be a finite rooted tree with* $T = (V(T), E(T))$ *and* $|V(T)| = n$*. Let* $\mathcal{S} = \{-1, 1\}$*,* $\Omega = \mathcal{S}^n$*. Let* $\epsilon$*,* $\beta$ *satisfy*

$$\frac{\epsilon}{1-\epsilon} = \exp(-2\beta). \tag{4.1}$$

*Then using the probability measure* $\mu_\beta$ *of the Ising model* (3.3) *with* $B = 0$ *on* $T$ *and assigning probabilities using* $\{\eta_e\}$ *as in* (2.44) *and* (2.46) *on* $T$ *lead to the same type distribution a.e..*

*Proof.* First note that in both distributions, the probability of a state $\boldsymbol{\sigma}$ is dependent only on the distribution of $\sigma_\rho$ and the total amount of edges $(u, v)$ with $\sigma_u \neq \sigma_v$, called flips.

For both models we have $\mathbb{P}(\sigma_\rho = 1) = \mathbb{P}(\sigma_\rho = -1) = 1/2$. In the broadcasting process (2.46) this is by definition. In the Ising measure (3.3) with $B = 0$, only the amount of flips influence the probability of a state, which is not influenced by $\sigma_\rho$.

Let $(u, v) \in E(T)$ and let $T_v$ denote the subtree of $v$, $v$ and all its descendants. Let $\boldsymbol{\sigma} \in \Omega$ be a state with $\sigma_u = \sigma_v$ and consider what happens in both models when flipping all types in $T_v$, creating $\boldsymbol{\sigma}_{flip}$, as can be seen in figure 4.3. For the broadcasting process this means that in the realization of $\{\eta_e\}$, only $\eta_{(u,v)}$ is different, and for the probability of this flip we find using (2.44) that

$$\begin{aligned} \epsilon\mathbb{P}(\eta_{(u,v)} = 1) &= (1-\epsilon)\mathbb{P}(\eta_{(u,v)} = -1) \\ \epsilon\mathbb{P}_\eta(\boldsymbol{\sigma}) &= (1-\epsilon)\mathbb{P}_\eta(\boldsymbol{\sigma}_{\mathrm{flip}}). \end{aligned} \tag{4.2}$$

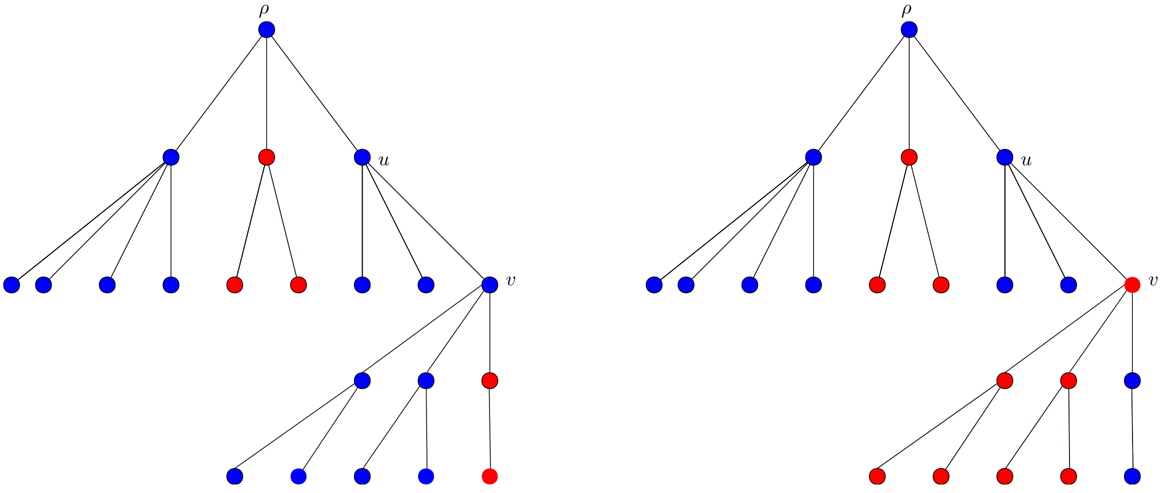
For the Ising model with $B = 0$ we can use the definition (3.3). Let $k$ be the total number of flips in $\boldsymbol{\sigma}$. We find

$$\begin{aligned} \mu_\beta(\boldsymbol{\sigma}) &= \frac{1}{Z}\exp\{\beta((n-k) - k)\} \\ \mu_\beta(\boldsymbol{\sigma}_{\mathrm{flip}}) &= \frac{1}{Z}\exp\{\beta[(n-(k+1)) - (k+1)]\} \\ \mu_\beta(\boldsymbol{\sigma}) &= \exp(2\beta)\mu_\beta(\boldsymbol{\sigma}_{flip}). \end{aligned} \tag{4.3}$$

Thus both measures assign probabilities in the same ratio when (4.1) is satisfied. Since both measures are probability measures ($\mathbb{P}(\Omega) = \mu_\beta(\Omega) = 1$) and therefore normalized, this implies that

$$\mathbb{P}_\eta(\boldsymbol{\sigma}) = \mu_\beta(\boldsymbol{\sigma}) \forall \boldsymbol{\sigma} \in \Omega \qquad \Leftrightarrow \qquad \frac{\epsilon}{1-\epsilon} = \exp(-2\beta). \tag{4.4}$$

□



(a) $\mathbf{t}$

(b) $\mathbf{t}_{\text{flip}}$

Figure 4.3: A tree $T$ with a state $\mathbf{t}$ with $T_u = t_v$ in (a) and the state in which all vertices of $T_v$ have opposite type with respect to $\mathbf{t}$, state $\mathbf{t}_{\text{flip}}$ in (b).

We believe this result can be extended to infinite trees, but this won't be done in this thesis.

Now that we described the Ising model on rooted trees as a broadcasting process, it is much easier to compare with local limits. Thus we can now prove the following result.

**Lemma 4.2.3** (equivalence local limit of a marked 2-type SBM and Ising broadcasting process on local limit ER)**.** *Let the marked* $\mathrm{GWT}(Y, (\frac{1}{2}, \frac{1}{2}))$ *with* $Y$ *as in* (2.33) *be the local limit of a marked* $\mathrm{SBM}(2, a, b)$*, denoted as* $(T_1, \rho)$*. Let* $\mathrm{GWT}(\mathrm{Poi}(\lambda))$ *be the local limit of* $\mathrm{ER}_n(\lambda/n)$ *and let* $\mathbb{P}_\eta$ *be a broadcasting process imposed on* $\mathrm{GWT}(\mathrm{Poi}(\lambda))$*. Call the resulting (marked) tree* $(T_2, \rho)$*. Then* $(T_1, \rho)$ *and* $(T_2, \rho)$ *have equal distribution when*

$$\lambda = \frac{a+b}{2} \tag{4.5}$$

*and*

$$\epsilon = \frac{b}{a+b}. \tag{4.6}$$

*Proof.* For the root we find that $\sigma_\rho$ is distributed uniformly at random for both trees by definition. For the offspring of every vertex of $T_1$ we find $\mathrm{Poi}(\frac{a+b}{2}) = \mathrm{Poi}(\lambda)$, the offspring of every vertex of $T_2$. For the type distribution of $T_2$, $Q_2(i)$, we find using (2.52) that

$$\mathbb{P}(Q_2(i) = j) = \mathbb{P}_\eta(\eta_{(u,v)} = \frac{j}{i}) = \begin{cases} 1 - \epsilon = \frac{a}{a+b} & \text{for } i = j \\ \epsilon = \frac{b}{a+b} & \text{for } i = -j \ (i \neq j), \end{cases}$$

which is equal to type distribution $Q(i)$ of $T_1$ from (2.32) with $q = 2$. □

Lemma 4.2.2 shows that the Ising model on $\mathrm{ER}_n(\lambda/n)$ is equivalent to a broadcasting process and lemma 4.2.3 proves that the local limit of $\mathrm{SBM}_n(2, a, b)$ and that broadcasting process are equivalent, when $\lambda = \frac{a+b}{2}$

and

$$\begin{aligned} \epsilon &= \frac{b}{a+b} \\ \exp(-2\beta) &= \frac{\epsilon}{1-\epsilon} = \frac{\frac{b}{a+b}}{1-\frac{b}{a+b}} = \frac{b}{a} \\ \beta &= \frac{1}{2}log(\frac{a}{b}), \end{aligned} \tag{4.7}$$

which proves theorem 4.2.1.

## 4.3 Comparing successful community detection results

Now that there is an equivalence between these models, we can compare results found for community detection found for SBM's, broadcasting processes and Ising models. We show that parameters which determine when community detection is successful in these different fields can be coupled to each other.

In the local limit of $\mathrm{ER}_n(\lambda/n)$, $\lambda$ is precisely the branching factor $\bar{v}$. Intuitively this may be strange, as the average degree is $\lambda$, so then shouldn't the branching factor be $\lambda - 1$? However, you can think about it like this. Being connected to a parent $v$ means that the expected number of edges to be connected to beside $v$, the branching factor, is $\lambda\frac{n-1}{n}$. As $n \to \infty$, this converges to $\lambda$.

**Corollary 4.3.1** (Equivalence reconstruction results)**.** *Let $G$ be a graph created from $\mathrm{SBM}_n(2, a, b)$, then as $n \to \infty$, reconstruction of original communities is successful if $(a-b)^2 > 2(a+b)$ [9] and unsuccessful if $(a-b)^2 < 2(a+b)$. Consider $GWT(Poi\lambda))$, equipped with a broadcasting process $\mathbb{P}_\eta$ with $\{eta_{(u,v)}\}$ satisfying equations (2.44) and (2.46). Then reconstructing $\sigma_\rho$ is successful when $n \to \infty$ if $1-2\epsilon > \bar{v}^{-1/2}$ and unsuccessful if $1-2\epsilon < \bar{v}^{-1/2}$ [2]. Now let $\lambda, \epsilon, \beta, a, b$ satisfy relations (4.5),(4.6) and (4.1), then these critical values coincide.*

*Proof.* We find

$$\begin{aligned} (a-b)^2 = \left(\frac{a-b}{a+b}\right)^2 (a+b)^2 = (1-2\epsilon)^2(a+b)^2 &= 2(a+b) \\ (1-2\epsilon)^2 &= \frac{2}{a+b} \\ 1-2\epsilon &= \sqrt{\frac{2}{a+b}} = \lambda^{-1/2} = \bar{v}^{-1/2} \end{aligned} \tag{4.8}$$

□

We can express this in terms of $\beta$ as

$$\tanh\beta_c = \frac{-1+e^{2\beta_c}}{1+e^{2\beta_c}} = \frac{\frac{1-\epsilon}{\epsilon}-1}{1+\frac{1-\epsilon}{\epsilon}} = \frac{\frac{1-2\epsilon}{\epsilon}}{\frac{1}{\epsilon}} = 1-2\epsilon = \sqrt{\frac{2}{a+b}} = \bar{v}^{-1/2}. \tag{4.9}$$

This unfortunately is still a sqrt factor of of the result of Pemantle [7] on the phase transition of Ising models on trees. This would be explained with a simple factor 2 missing in their definition of the ising model, which would be a misinterpretation on our side. Otherwise the difference could lie in the fact that they look at a robust phase transition, which is stronger than the usual phase transition.

## 4.4 Extension to q types

In this section we show that the result section 4.2 can be extended to $q$ types. Let us now look at the local limit of $\mathrm{SBM}(q, a, b)$ and a Potts model measure as in (3.19) on the local limit $\mathrm{ER}_n(\lambda/n)$. We show that these are again equivalent using the right parameters.

Now this equivalence will be done in three steps. First we show when a broadcasting process assigns probabilities independent of its root type $t_\rho$. Then we identify that a $q$-type broadcasting process satisfies this property and is equivalent to a Potts model on trees. Finally we show that a broadcasting process on the local limit of $\mathrm{ER}(\lambda/n)$ is equivalent to the local limit of a marked $\mathrm{SBM}(q, a, b)$.

In order to have exactly the same probability distribution starting from every type in a broadcasting process, we somehow need all the types to be able to be permuted onto any other type, without the probabilities of $\eta_e$ changing. In graph theory such a relation has been defined, and is called vertex transitive.

**Definition 4.4.1** (Automorphism, vertex transitive graph)**.** *Let $G = (V(G), E(G))$ be a graph where every edge has a weight $\phi$. Then permutation $\pi$ of the vertex set $V(G)$ is an automorphism if*

$$(\pi(u), \pi(v)) \in E(G) \Leftrightarrow (u, v) \in E(G) \tag{4.10}$$

*and*

$$\phi((\pi(u), \pi(v))) = \phi((u, v)) \forall (u, v) \in E(G). \tag{4.11}$$

*Now $G$ is called vertex transitive if for any two vertices $u, v$, there is an automorphism $\pi$ with $\pi(u) = v$.*

**Lemma 4.4.2** (type transitive broadcasting process)**.** *Let $\mathbb{P}_\eta$ be any broadcasting process satisfying (2.46) with root type distribution $\vec{t_\rho} = (\frac{1}{q}, ..., \frac{1}{q})$. Define $q \times q$ matrix $A$ as*

$$A_{ij} = \mathbb{P}(\eta_{(u,v)} = \frac{j}{i}) \tag{4.12}$$

*and assume that $\{\eta_e\}$ is distributed such that $A$ is symmetric. Let $G$ be a graph of $q$ vertices where each edge $(i, j)$ has weight $A_{ij} = A_{ji}$.*

*Then if $G$ is vertex transitive, then for any state $\boldsymbol{t}$ and any type $j$ there is permutation $\pi$ with $\pi(t_\rho) = j$, such that*

$$\mathbb{P}_\eta(\boldsymbol{t}) = \mathbb{P}_\eta(\pi(\boldsymbol{t})). \tag{4.13}$$

*We call $\mathbb{P}_\eta$ a type-transitive broadcasting process*

*Proof.* First note that the root type is chosen uniformly at random. $G$ is vertex transitive. Thus for any two vertices $i, j$, there is an automorphism $\pi$ with $\pi(i) = j$. We take $i = t_\rho$. Once the root type is chosen, the probability of a state is entirely dependent on the matrix $A$. Since $\pi$ is an automorphism, all edges in $\pi(G)$ have the exact same weight and thus its adjacency matrix $A_{\pi(G)}$ satisfies $A_{\pi(G)} = A$. Therefore (4.13) holds. □

Note that this constraint has still quite a large set of matrices containing it. An example of an adjacency matrix $A$ for which $G$ is still vertex transitive is

$$A_1 = \begin{pmatrix} 4 & 1 & 0 & 2 & 0 & 0 \\ 1 & 4 & 0 & 0 & 2 & 0 \\ 0 & 0 & 4 & 1 & 0 & 2 \\ 2 & 0 & 1 & 4 & 0 & 0 \\ 0 & 2 & 0 & 0 & 4 & 1 \\ 0 & 0 & 2 & 0 & 1 & 4 \end{pmatrix}. \tag{4.14}$$

This is more easily visible by drawing the corresponding graph $G_1$, which we do in figure 4.4. Suppose we have a permutation which has $\pi_1(1) = 2$, then we can turn this into an automorphism using $\pi_1(\{1, 2, 3, 4, 5, 6\}) = \{2, 1, 6, 5, 4, 3\}$. For $\pi_2(1) = 3$ we have automorphism $\pi_2(\{3, 4, 5, 6, 1, 2\})$ and for $\pi_3(1) = 4$ we have automorphism $\pi_3(\{4, 3, 2, 1, 6, 5\})$. We leave the rest of the automorphisms as an exercise to the reader as they can all be done using the symmetry present in $G$.

Note that for a type-transitive broadcasting processes, $A$ is a matrix composed of probabilities. Therefore it also needs to satisfy the property that it is doubly stochastic, which is that the sums of the rows and columns add up to 1. Note that $\frac{1}{7}A_1$ is such a matrix.

For the $q$-type broadcasting process we have

$$A = \begin{pmatrix} \ddots & & \boldsymbol{\epsilon} \\ & 1 - (q-1)\epsilon & \\ \boldsymbol{\epsilon} & & \ddots \end{pmatrix}, \tag{4.15}$$

as it was defined in (2.51). For this matrix $G$ is symmetric, which implies it is vertex transitive and therefore $\mathbb{P}_\eta$ is type-transitive.

With this result for the $q$-type broadcasting process we can expand theorem 4.2.2.

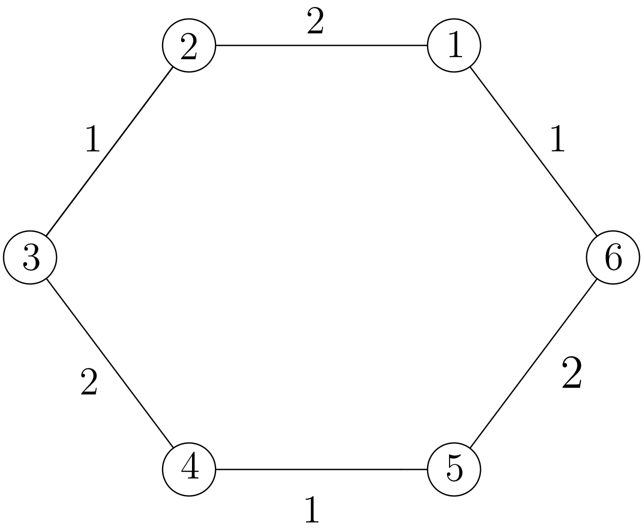


Figure 4.4: The vertex transitive graph $G_1$ with adjacency matrix $A_1$. We left out the edges from nodes to themselves as they are all equal to 4.

**Theorem 4.4.3** (equivalence Potts measure and probability measure broadcasting process)**.** *Let $(T, \rho)$ be a finite rooted tree with $T = (V(T), E(T))$ with $|V(T)| = n$. Let $\Omega = [q]^n$. Let $\epsilon$, $\beta$ satisfy*

$$\frac{(q-1)\epsilon}{1-(q-1)\epsilon} = \exp(-2\beta). \tag{4.16}$$

*Then using the probability measure $\mu_\beta$ of the Potts model (3.19) on $T$ and using the q-type broadcasting process $\mathbb{P}_\eta$ satisfying (2.46) and (2.52) on $T$ lead to the same type distribution a.s..*

*Proof.* First note that in both distributions, the probability of a state $\mathbf{t}$ is dependent only on the distribution of $t_\rho$ and the total amount of edges $(u, v)$ with $t_u \neq t_v$, called flips. For both models we have $\mathbb{P}(t_\rho = i) = 1/q$. In (2.46) this is by definition, whilst we reach this conclusion in (3.3) by noting that only the amount of flips influence the Potts measure, which is not influenced by $t_\rho$ a priori.

Let $(u, v) \in E(T)$. Let $\mathbf{t} \in \Omega$ be a state with $t_u = t_v$ and consider what happens in both models when all types on one side of the edge $(u, v)$, the subtree $T_v$, are permuted. For this a permutation $\pi : [q] \to [q]$ is used, which interchanges types $t_v$ with any other type $i$, with $t_v \neq i$. This creates $\mathbf{t}_{\text{flip}}$. Let $\mathbf{t}_{T_v}$ denote the state $\mathbf{t}$, restricted to subtree $T_v$. The key to this theorem will be that for the broadcasting process, we have that

$$\mathbb{P}_\eta(\mathbf{t}_{T_v}) = \mathbb{P}_\eta(\pi(\mathbf{t}_{T_v})) = \mathbb{P}_\eta(\mathbf{t}_{T_v,\text{flip}}). \tag{4.17}$$

Because we are only considering $T_v$, we can identify $v$ as the root of $T_v$. Since $\mathbb{P}_\eta$ is a type transitive broadcasting process, (4.17) holds.

Thus only the impact of $\eta_{(u,v)}$ is felt in the probability of $\mathbf{t}$ and we find using (2.52) that

$$\begin{aligned}
\mathbb{P}(\eta_{(u,v)} = 1) &= 1-(q-1)\epsilon \\
\mathbb{P}(\eta_{(u,v)} \neq 1) &= (q-1)\epsilon \\
(q-1)\epsilon\mathbb{P}(\eta_{(u,v)} = 1) &= (1-(q-1)\epsilon)\mathbb{P}(\eta_{(u,v)} \neq 1) \\
(q-1)\epsilon\mathbb{P}_\eta(\mathbf{t}) &= (1-(q-1)\epsilon)\mathbb{P}_\eta(\mathbf{t}_{\text{flip}}).
\end{aligned} \tag{4.18}$$

For the Potts model we again take the definition (3.19). Let $k$ be the total number of flips in $\mathbf{t}$ then we have

$$\begin{aligned}
\mu_\beta(\mathbf{t}) &= \frac{1}{Z}\exp(2\beta(n-k)) \\
\mu_\beta(\mathbf{t}_{\text{flip}}) &= \frac{1}{Z}\exp(2\beta(n-(k+1))) \\
\mu_\beta(\mathbf{t}) &= \exp(2\beta)\mu_\beta(\mathbf{t}_{\text{flip}}).
\end{aligned} \tag{4.19}$$

Thus both measures assign probabilities in the same ratio when (4.16) is satisfied. Since both measures are probability measures ($\mathbb{P}(\Omega) = \mu_\beta(\Omega) = 1$) and therefore normalized, this implies that

$$\mathbb{P}(\mathbf{t}) = \mu_\beta(\mathbf{t}) \forall \mathbf{t} \in \Omega \text{ iff } \qquad \frac{(q-1)\epsilon}{1-(q-1)\epsilon} = \exp(-2\beta). \tag{4.20}$$

□

**Theorem 4.4.4** (equivalence local limit of SBM and Potts broadcasting process on local limit ER)**.** *Let $\mathcal{S} = [q]$.*

*Let $(T_1, \rho)$ be the local limit of a marked* $\mathrm{SBM}_n(q, a, b)$*.*

*Let $GWT(Poi(\lambda))$ be the local limit of* $\mathrm{ER}_n(\lambda/n)$*. Now assign types to $GWT(Poi(\lambda))$ by a q-type broadcasting process $\mathbb{P}_\eta$ satisfying* (2.46)*, with $\{\eta_e\}$ satisfying* (2.52) *and giving the root type $t_\rho \in [q]$ uniformly at random. Denote the resulting (marked) tree as $(T_2, \rho)$.*

*Then $(T_1, \rho)$ and $(T_2, \rho)$ have the same distribution when*

$$\lambda = \frac{a + (q-1)b}{q} \tag{4.21}$$

*and*

$$\epsilon = \frac{b}{a + (q-1)b}. \tag{4.22}$$

*Proof.* For the root we find that $t_\rho$ uniformly at random in both models. For the offspring of every vertex we find $Poi(\lambda) = Poi(\frac{a+(q-1)b}{q})$. For the type distribution $Q_2(i)$ in $T_2$ we find that

$$\mathbb{P}(Q_2(i) = j) = \mathbb{P}_\eta(\eta_{(u,v)} = \frac{j}{i}) = \begin{cases} 1-(q-1)\epsilon = \frac{a}{a+(q-1)b} & \text{for } i = j \\ \epsilon = \frac{b}{a+(q-1)b} & \text{for } i \neq j \end{cases}$$

which is equal to type distribution $Q(i)$. □

Combining theorem 4.4.4 and 4.4.3 we find equivalence of the local limit of the $\mathrm{SBM}_n(q, a, b)$ and a Potts measure on the local limit of $\mathrm{ER}_n(\frac{a+(q-1)b}{n})$ when

$$\exp(-2\beta) = \frac{(q-1)\epsilon}{1-(q-1)\epsilon} = \frac{(q-1)b}{a} \qquad \beta = -\frac{1}{2}\log(\frac{(q-1)b}{a}), \tag{4.23}$$

which reduce to the relations we found between the parameters when $q = 2$ in (4.7).

**Corollary 4.4.5** (Equivalence successful community detection results)**.** *Let $G$ be a graph created from* $\mathrm{SBM}_n(q, a, b)$*, then as $n \to \infty$, reconstruction of original communities with positive correlation, i.e. better than random guessing, is possible iff $(a-(q-1)b)^2 > q(a+(q-1)b)$ [9]. Let $(T, \rho)$ be a GWT with $t_\rho$ uniformly at random and offspring distribution $Poi(\lambda)$, where the types are determined using $\mathbb{P}_\eta$, $\{\eta_e\}$ satisfying* (2.52) *and* (2.46)*. Then reconstructing $\sigma_\rho$ from the spins at the n'th level of $T$, $T_n$, is possible when $n \to \infty$ if $1-(q-1)\epsilon > \lambda^{-1/2}$ and not possible if $1-(q-1)\epsilon < \lambda^{-1/2}$. For the critical $\beta$ of the phase transition on the Potts models [7] find the relation $\frac{e^{\beta_c} - e^{-\beta_c}}{e^{\beta_c} + (q-1)e^{-\beta_c}} = \bar{v}^{-1}$.* (3.22)

*Now let $\lambda, \epsilon, \beta, a, b$ satisfy relations* (4.21)*,*(4.22) *and* (4.16)*, then these critical values coincide.*

*Proof.* We find

$$\begin{aligned} (a-(q-1)b)^2 = \left(\frac{a-(q-1)b}{a+(q-1)b}\right)^2 (a+(q-1)b)^2 = (1-2(q-1)\epsilon)^2(a+(q-1)b)^2 = q(a+(q-1)b) \\ (1-2(q-1)\epsilon)^2 = \frac{q}{a+(q-1)b} = \lambda^{-1} \\ 1-2(q-1)\epsilon = \lambda^{-1/2} \end{aligned} \tag{4.24}$$

This results in the critical $\beta_c$ with

$$\tanh(\beta_c) = \frac{-1+e^{2\beta_c}}{1+e^{2\beta_c}} = \frac{\frac{1-(q-1)\epsilon}{(q-1)\epsilon} - 1}{1 + \frac{1-(q-1)\epsilon}{(q-1)\epsilon}} = \frac{\frac{1-2(q-1)\epsilon}{(q-1)\epsilon}}{\frac{1}{(q-1)\epsilon}} = 1-2(q-1)\epsilon = \frac{a-b(q-1)}{a+b(q-1)} \tag{4.25}$$

$$\tanh(\beta_c) = \lambda^{-1/2} = \bar{v}^{-1/2} \tag{4.26}$$

This does not correspond to the result of Pemantle [7] of (3.22) on the robust phase transition of Potts models on trees. One can even redefine $\beta$ in terms of flipping away to a specific type,

$$\exp(-2\beta) = \frac{\epsilon}{1-(q-1)\epsilon}. \tag{4.27}$$

Then

$$\frac{e^{\beta_c} - e^{-\beta_c}}{e^{\beta_c} + (q-1)e^{-\beta_c}} = \frac{-1+e^{2\beta_c}}{(q-1)+e^{2\beta_c}} = \frac{\frac{1-(q-1)\epsilon}{\epsilon} - 1}{(q-1) + \frac{1-(q-1)\epsilon}{\epsilon}} = \frac{\frac{1-q\epsilon}{\epsilon}}{\frac{1}{\epsilon}} = 1 - q\epsilon \tag{4.28}$$

which also does not correspond to the branching factor. This difference could lie in the fact that they look at a robust phase transition, which is stronger than the usual phase transition. Else we may have misinterpreted their results.

□

## 4.5 General broadcasting process and inhomogeneous random graph

We would like to extend the results from the previous section to be as general as possible. From theorems 4.4.3 and 4.4.4 we conclude that there are two main components to the equivalence. The first is the equivalence between a type-transitive broadcasting process and a Potts-model, which we work out more in-depth in the next section 4.6. The second is the relation between a broadcasting process applied to the local limit of an ER graph with the local limit of the more general IRG, which we discuss in this section.

We show in this section that the local limit of the IRG and a broadcasting process on the local limit of an ER graph can be mapped onto each other, as long as the IRG has the same total expected number of connections for every type. This is a necessary assumption, as a broadcasting process only assigns types and cannot compensate for changes in the distribution of the offspring of the tree. These can only be coupled with an ER graph with parameter $\lambda$ if the expected offspring of every type is really $\lambda$.

Thus we look at $q \times q$ kernels for which

$$\lambda = \lambda(i) = \sum_{j\in[q]} c_{ij} n_j. \tag{4.29}$$

In other words, the expected number of connections of a node of type $i$, $\lambda(i)$, is equal to $\lambda$ for all the different types. Now as $c_{ij}$ can be different for specific $i, j$, we need to define separate probabilities of flipping from type $i$ to type $j$, $\epsilon_{ij}$ to contain this information. We define

$$\epsilon_{i,j} = \frac{c_{ij} n_j}{\lambda}. \tag{4.30}$$

We also redefine $\eta_{(u,v)}$ accordingly so that

$$\mathbb{P}(\eta_{(u,v)} = \frac{i}{t_u}) = \epsilon_{t_u i} = \frac{c_{t_u i} n_i}{\lambda} \tag{4.31}$$

We call the resulting $\mathbb{P}_\eta$ satisfying (2.46) an *inhomogenious broadcasting process.*

Using these definitions we prove the following lemma.

**Lemma 4.5.1** (Equivalence local limit IRG and inhomogenious broadcasting process on local limit ER)**.** *Let $(T_1, \rho)$ be the local limit of a marked* IRG($c$)*, where $c$ satisfies* (4.29) *and have type probabilities $\vec{n}$.*

*Let* GWT(Poi($\lambda$)) *be the local limit of* ER($\lambda/n$) *2.3. Equip* GWT(Poi($\lambda$)) *with an inhomogenious broadcasting process $\mathbb{P}_\eta$ with $\{\eta_e\}$ which satisfy* (2.46) *and* (4.31)*. Distribute the type of the root $t_\rho$ using type probabilities $\vec{t_\rho} = \vec{n}$. Denote this marked tree as $(T_2, \rho)$.*

*Then these objects have the same distribution.*

*Proof.*

$$\mathbb{P}(t_\rho = i) = n_i \tag{4.32}$$

for both models by definition.

The total offspring of $(T_1, \rho)$ of type $i$ is distributed with $\mathrm{Poi}(\lambda(i))$, where we copy this result from (2.25). We find

$$\lambda(i) = \sum_{j \in [q]} c_{ij} n_j = \lambda. \tag{4.33}$$

The offspring of $(T_2, \rho)$ is also $\mathrm{Poi}(\lambda)$ distributed, as the underlying tree is $\mathrm{GWT}(\mathrm{Poi}(\lambda))$.

For the type distribution of $(T_1, \rho)$ we copy the result from (2.26), which is that

$$\mathbb{P}(Q_1(i) = j) = \frac{c_{ij} n_j}{\sum_{j \in [q]} c_{ij} n_j} = \frac{c_{ij} n_j}{\lambda}. \tag{4.34}$$

For the broadcasting process we find

$$\mathbb{P}(Q_2(i) = j) = \mathbb{P}_\eta(\eta_{(u,v)} = \frac{j}{i}) = \epsilon_{i,j} = \frac{c_{ij} n_j}{\lambda}. \tag{4.35}$$

Thus the offsprings and types are distributed same and thus both trees $(T_1, \rho)$ and $(T_2, \rho)$ have the same distribution. □

## 4.6 Type-transitive broadcasting process and adjusted spin glass Potts models

In this section we generalize the coupling of a broadcasting process to a Potts model. Let $L$ denote the property that the probability of a state which is permuted on all types of sub-tree $T_v$ is independent of the sub-tree $T_v$. In section 4.4 we have seen that for broadcasting processes, $L$ holds as long as the broadcasting process is type-transitive. In this section we show how much we can generalize the Potts model such that $L$ still holds. When property $L$ holds for both models, we can prove the equivalence of their type distributions on trees analogous to theorem 4.4.3.

We investigate multiple extensions to the Potts model. First we show that applying a magnetic field $\vec{B} \neq \mathbf{0}$ (and not rescalable to $\mathbf{0}$), property $L$ does not hold. Afterwards we show that we can adjust the spin glass Potts model in such a way that it is more general than the Potts model and $L$ still holds.

For simplicity we look at the Ising model on a tree $T$ with a magnetic field $B \neq 0$, as described in (3.3). Let $(u, v) \in E(T)$. We investigate what happens to the probability of a state $\boldsymbol{\sigma}$ with $\sigma_u = \sigma_v$, when we flip all types of $T_v$, called $\boldsymbol{\sigma}_{\text{flip}}$. Without loss of generality, let $B > 0$. Let $k$ denote the number of flips in state $\boldsymbol{\sigma}$ and let $r$ denote the number of vertices $u$ with $\sigma_u = +1$ in $\sigma$. Let $|T_v|$ denote the number of vertices in $T_v$ and $s$ denote the number of vertices $u$ in $T_v$ with $\sigma_u = +1$. Using the definition of the Ising model we find

$$\mu_\beta(\boldsymbol{\sigma}) = \frac{1}{Z} \exp\{\beta\left((n-k) - k + B(r - (n-r))\right)\} \tag{4.36}$$

$$\mu_\beta\left(\boldsymbol{\sigma}_{\text{flip}}\right) = \frac{1}{Z} \exp\{\beta\left((n-(k+1)) - (k+1) + B(2r - n + (|T_v| - 2s))\right)\} \tag{4.37}$$

$$\mu_\beta(\boldsymbol{\sigma}) = \exp(2\beta + \beta B(2s - |T_v|)\mu_\beta(\boldsymbol{\sigma}_{\text{flip}}). \tag{4.38}$$

This is dependent on $s$ and $|T_v|$ therefore not independent of the subtree $T_v$. This argument also extends to $q$ types and thus Potts models with magnetic field $\vec{B} \neq 0$ do not satisfy property $L$.

Now we extend the Potts model such that it still satisfies property $L$. We can do this by adding an interaction strength $J_{uv}$ to every edge, as is done in the spin glass Potts model. We repeat here the definition of the spin glass Potts model with $\vec{B} = \mathbf{0}$ as we will need it to show that property $L$ still holds.

$$\mu_{\text{glass}}(\mathbf{t}) = \frac{1}{Z} \exp\{2\beta \left( \sum_{(u,v) \in E} J_{uv} \mathbb{1}(t_u = t_v) \right)\}. \tag{4.39}$$

Using the definition of $\mu_{\text{g}}$ (4.39) we find

$$\begin{aligned}\mu_{\text{glass}}(\mathbf{t}) = \frac{1}{Z} \exp\{2\beta \left( \sum_{(u',v') \in E} J_{u'v'} \mathbb{1}(t_{u'} = t_{v'}) \right)\} = \\ = \exp(2\beta J_{uv}) \frac{1}{Z} \exp\{2\beta \left( \sum_{(u',v') \in E \setminus \{(u,v)\}} J_{u'v'} \mathbb{1}(t_{u'} = t_{v'}) \right)\} = \exp(2\beta J_{uv}) \mu_{\text{glass}}(\pi_{T_v}(\mathbf{t})).\end{aligned} \tag{4.40}$$

This is independent of the specific tree $T_v$, therefore property $L$ holds.

However, in order to couple the Potts model to inhomogeneous type-transitive broadcasting models we need the interaction strength $J_{uv}$ to depend on the types of $t_u$ and $t_v$. Therefore we define

$$H(\mathbf{t}) = -2 \sum_{(u,v)\in E} J_{t_u,t_v}, \tag{4.41}$$

with $J_{t_u,t_v}$ the type-dependent interaction strength. This leads to an adjusted spin glass Potts model

$$\mu_{\mathrm{g}}(\mathbf{t}) = \frac{1}{Z} \exp\{2\beta( \sum_{(u,v)\in E} J_{t_u,t_v})\}. \tag{4.42}$$

Now we apply $\mu_{\mathrm{g}}$ to a tree $(T, \rho)$ with $(u, v) \in E(T)$. We investigate what happens to the probability of a state $\mathbf{t}$ with $t_u = t_v$ when permuting all types of subtree $T_v$ using a permutation $\pi_{T_v}$ with $\pi_{T_v}(t_u) = i \neq \pi_{T_v}(t_v)$, resulting in the state $\pi_{T_v}(\mathbf{t})$. As now these $J_{t_u,t_v}$ in principle impact the rest of the tree. We therefore also need type-transitivity for the adjusted Potts model. We say $\mu_{\mathrm{g}}$ is type-transitive when the graph $G$ with adjacency matrix $J$ is vertex-transitive. When this is satisfied we find

$$\mu_{\mathrm{g}}(\mathbf{t}) = \exp(2\beta(J_{t_u t_u} - J_{t_u i}))\mu_{\mathrm{g}}(\pi_{T_v}(\mathbf{t})). \tag{4.43}$$

Using this we can now show the following theorem.

**Theorem 4.6.1.** *Let $(T, \rho)$ be a finite tree. Let $\mu_\beta$ be a q-type spin glass Potts model as in (4.42) with $\vec{B} = \mathbf{0}$ and let $\mathbb{P}_\eta$ be a type-transitive broadcasting process as in (4.13), with $A_{ij} > 0$ for all $i, j$. The root is assigned a type from $[q]$ uniformly at random. Let $J_{uv}$ be dependent on the type $t_u$, such that*

$$J_{ij} = \log(\epsilon_{ij}). \tag{4.44}$$

*Then equipping $(T, \rho)$ with $\mu_{\mathrm{g}}$ and $\mathbb{P}_\eta$ lead to the same distribution.*

*Proof.* Note that the type of the root is distributed uniformly at random from $[q]$ in both models, in the broadcasting process by definition and in $\mu_{\mathrm{g}}$ by We now use the results of equation (4.43) for the ratio of $\mu_{\mathrm{g}}$. For the ratio of probabilities of a state $\mathbf{t}$ with $t_u = t_v$ and state $\pi_{T_v}(\mathbf{t})$ for which all types of $T_v$ are permuted using $\pi_{T_v}$ with $\pi_{T_v}(t_u) = i \neq t_u$, we find in the type-transitive broadcasting model that

$$\mathbb{P}(\eta_{(u,v)} = 1) = \epsilon_{t_u t_u}$$
$$\mathbb{P}(\eta_{(u,v)} = \frac{j}{i}) = \epsilon_{t_u i}$$
$$\mathbb{P}(\mathbf{t}) = \frac{\epsilon_{t_u,t_u}}{\epsilon_{t_u i}} \mathbb{P}(\pi_{T_v}(\mathbf{t})). \tag{4.45}$$

For this we which coincide when

$$\exp(2\beta(J_{ii} - J_{ij})) = \frac{\epsilon_{ii}}{\epsilon_{ij}}, \tag{4.46}$$

which is satisfied with constraint (4.44).

□

We realize that by making the coupling factor $Juv$ dependent on the types $t_u, t_v$, there is no longer much physical intuition of Potts models that applies. However, we merely wanted to show in what direction the Potts model has to be changed in order to match the broadcasting description even better.

## 4.7 Extension to degree corrected models

Here we extend the result of section 4.4 to degree-corrected models. This is possible, because degree-correcting only affects the underlying graph structure. Thus it is something we can do to both both the ER model and the SBM, which changes both models in precisely the same way such that the equivalences still hold. This result is very nice, as many real-world problems are degree-corrected.

We can use the same identification of $\beta, \lambda$ as (4.23), (4.21). We find that the local limit of a degree-corrected $\mathrm{SBM}_n(q, a, b)$ is equivalent to a Potts model $\mu_\beta$ as (4.23) on the local limit of a degree-corrected $\mathrm{ER}_n(\lambda/n)$.

We use the notation described in [4]. Every node $u$ has a weight $\phi_u$ given by a distribution $\nu$ with some minimal weight $\phi_{min} > 0$ and this weight has moments $\Phi^{(k)} = \int_W x^k d\nu(x)$.

The local limit of a DSBM was given in section 2.3, which we repeat here. It is a GWT, with root type $t_\rho \in \{1, ..., q\}$ taken uniformly at random and root weight $\phi_\rho$ taken random according to $\nu$. Now a vertex $u$ produces Poi($\frac{a\Phi^{(1)}\phi_u}{q}$) children of the same type $\sigma_u$ and Poi($\frac{b\Phi^{(1)}\phi_u}{q}$) of each of the other types. The weights of the children are i.i.d. distributed with size-biased distribution

$$\nu^*([0, x]) = \frac{1}{\Phi^{(1)}} \int_{\phi_{min}}^{x} y d\nu(y) \tag{4.47}$$

.

We define the degree-corrected ER as $\text{DER}_n(\lambda/n)$ as a DSBM with $q = 1$ and $a = \lambda$.

Notice that theorem 4.4.3 is true on any tree, so also degree-corrected trees. We can thus immediately continue with the equivalence of the local limits.

**Theorem 4.7.1** (equivalence local limit of DSBM and $q$-type broadcasting process on local limit degree-corrected ER)**.** *Let $(T_1, \rho)$ be the local limit of* $\text{DSBM}_n(q, a, b, \nu)$*.*

*Let $(T_3, \rho)$ be the local limit of $\text{DER}_n(\lambda/n, \nu)$, a GWT where each vertex $u$ has offspring distribution* Poi($\lambda\Phi^{(1)}\phi_u$) *with the offspring weight distributed as $\nu^*$. Now equip $(T_3, \rho)$ with a q-type broadcasting process, which assigns the root type $t_\rho$ uniformly at random from $[q]$ and distributes types of other vertices using $\mathbb{P}_\eta$ and $\{\eta_e\}$ satisfying (2.52) and (2.46). Call the resulting marked graph $(T_2, \rho)$.*

*Then $(T_1, \rho)$ and $(T_2, \rho)$ yield the same distribution when*

$$\lambda = \frac{a + (q-1)b}{q}$$

*and*

$$\epsilon = \frac{b}{a + (q-1)b}.$$

*Proof.* For the root we find that $t_\rho$ uniformly at random for both trees. For the offspring of every vertex in $(T_2, \rho)$ is

$$\text{Poi}(\lambda\Phi^{(1)}\phi_u) = \text{Poi}(\frac{a + (q-1)b}{q}\Phi^{(1)}\phi_u) = \text{Poi}(\frac{a}{q}\Phi^{(1)}\phi_u) + (q-1)\text{Poi}(\frac{\mathrm{b}}{\mathrm{q}}\Phi^{(1)}\phi_{\mathrm{u}}). \tag{4.48}$$

Where we use the special property of poisson variables that the sum of two poisson variables is again poisson distributed with the sum of the parameters. The right-hand side of (4.48) is precisely the offspring of $(T_1, \rho)$.

For the type distribution $Q_2(i)$ in $T_2$ we find that

$$\mathbb{P}(Q_2(i) = j) = \begin{cases} 1 - (q-1)\epsilon = \frac{a}{a+(q-1)b} & \text{for } i = j \\ \epsilon = \frac{b}{a+(q-1)b} & \text{for } i \neq j \end{cases}$$

which is equal to type distribution $Q_1(i)$ from (2.32). Also both weight distributions of the root and the size-biassed weight of each offspring are equal. □

## 4.8 Conclusions

Thus we have seen in this chapter that in the thermodynamic limit $n \to \infty$, describing a graph locally from a random root $\rho$, by taking a local limit of random graph models with communities, the SBM, is the same thing as applying a Potts model to the local limit of an ER with the same total degree distribution $\lambda = \frac{a+(q-1)b}{q}$, when $\beta = -\frac{1}{2}\log(\frac{(q-1)b}{a})$. This is even the case for degree-corrected models.

We have seen that applying a magnetic field complicates matters in the equivalence between the Potts model and the broadcasting process, and cannot be proven using our method. This does not mean that no equivalence exists. Further research could be done on this subject, trying to couple a skewed matrix $A$ with a magnetic field $\vec{B}$. We conjecture that this is possible only if it is possible to find root probabilities in the magnetic Potts model that only depend on the size of the tree and not its specific structure.

We also showed that some specially adjusted Potts model is equivalent to a broadcasting tree on a larger space of type flipping matrixes $A$.

# Chapter 5

# Detecting communities using belief propagation algorithms

In the previous chapter we have focused on coupling Potts models and random graphs in the limit $n \to \infty$. This way we could compare results which state when community detection is possible. In this chapter we focus on a more practical question, which is when communities can be detected succesfully in polynomial time using an algorithm. Community detection is in practice usually done using belief propagation (bp) algorithms. In this chapter we explain what a belief propagation algorithm is and how it works. We then intuitively show where the idea of such an algorithm comes from and give the intuition of why and when these algorithms can accurately detect communities.

To do this, in section 5.1 we show what a general belief propagation algorithm is and how such algorithms could help us detect communities. In section 5.2, we show a bp algorithm TreeBP which calculates exact conditional probabilities on trees. In section 5.3 we explain the bp-algorithm of Decelle et al.[1] line by line, which we abbreviate as BPAD. We explain how BPAD was created to make it suited for community detection on IRG's. We expand on this in section 5.4 by showing under which conditions communities will be detected using BPAD. We then finish in section 5.5 by coupling these conditions with the critical parameters of community detection which were encountered in chapter 4.

## 5.1 Belief propagation algorithms

When is an algorithm a belief propagation algorithm? We call it so when a vertex passes on its belief about itself and other vertices to their neighbours in a message. These messages are then iterated on until they converge, such that the belief of all the nodes is stable. The stable belief of a node is then taken as a guess of the true state. These algorithms can be used to extrapolate information about unobserved nodes using observed nodes, which is for example done in some neural networks. When all the messages have converged, we will convert these messages into the probability of a vertex belonging to a type, which we can use to see if communities have truly been detected.

Why would such an algorithm be helpful for detecting communities? What use are the probabilities of vertices belonging to a type for the detection of communities? When all the type distributions of each vertex are known, we can simply categorise a vertex as the type it has the highest probability of having. Furthermore, a vertex has a probability of having type $i$ which is unequal to the global expected probability of that type, we know that some information has been gained above guessing at random. Thus intuitively, ending in a fixed point where any vertex defers from this global expected distribution is the condition for detecting communities.

In community detection we attempt to use the message passing to tell us something about the structure of the graph itself, without knowing anything about types in advance. The idea here is that well-connected groups amplify their initially random belief inside their group, thereby splitting into communities. Thus we want to know how to create messages which we can convert into type probabilities. We start with the calculation of a single type distribution of a vertex. In a Potts model, the type distribution $\nu_u(\vec{q}) = (\nu_u(1), ..., \nu_u(q))$ of an unknown node $u$ can be calculated by calculating all the marginal probabilities $\nu_u(i)$. We define

$$\nu_u(i) = \sum_{\mathbf{t}|t_u=i} \mu(\mathbf{t}). \tag{5.1}$$

as the marginal probability of type $i$ of vertex $u$. Here $\mu(\mathbf{t})$ is defined as the Potts model (3.19). Note that in this case we have that $\nu_u(i)$ is precisely the probability that $t_u = i$. Therefore $\nu_u(\vec{q})$ is the exact type distribution of vertex $u$, so the best guess of the type of node $u$, $t_u^*$ we can make is

$$t_u^* = \arg\max_i \nu_u(i), \tag{5.2}$$

where $t_u^*$ is chosen uniformly at random between multiple types satisfying the maximum when the maximum of $\nu_u(i)$ is not unique.

Note that equation (5.1) can take much calculation, as there are $q^{n-1}$ different configurations $\mathbf{t}|t_u = i$ to consider. On a tree however, we shall see in section 5.2 that these marginals can be calculated iteratively from leaves to the unknown node, leaving only $\mathcal{O}(n)$ calculations.

A belief propagation algorithm starts from some initial position, where each vertex is given an initial type distribution. Then in the calculation of the set of marginal probabilities for the next time step, it uses the current marginal probabilities of a node (which are approximations) as if they are the exact type distribution of that node. Using this calculation it starts iterating the marginal probabilities until these converge to a fixed point. At first it may not be intuitive that this leads to good approximations of the exact type distributions. However, in the next section we prove that on a tree, in the Potts model, this assumption leads to convergence of the marginal probabilities to the correct type distribution!

As our graphs are locally treelike, it is to be expected that as $n \to \infty$, these algorithms will also give accurate type distributions on graphs, not just trees. We will give some non-rigorous arguments for this in section 5.4.

In order to intuitively understand where bp-algorithms come from, we will first look at a way to calculate $\nu_u$ iteratively on a tree in a Potts model. Let the root $\rho$ be our unknown node and let $(T, \rho)$ be our rooted tree. Now suppose we know the exact type distribution of all leaves $u \in T$.

For this calculation, messages can be used. In this calculation, every node $u$ sends a message $\psi^{u\to v}$ to each of its parents $v$. A visual representation of this can be seen in figure 5.1. Here a message will represent the type distribution of the node $u$ in the subtree rooted at $u$ without its parent $v$, $T_{u\setminus v}$. Initially, each message $\psi^{u\to v}$ will contain the known type distributions of $u$, which in the leaves is equal to our wanted representation. Then we update each message from leaves up to the root by calculating he type distribution of the node $u$ in the subtree rooted at $u$. Then in the end the marginals of $\rho$ can be calculated using only messages $\psi^{u\to\rho}$, which is explained in section 5.2.

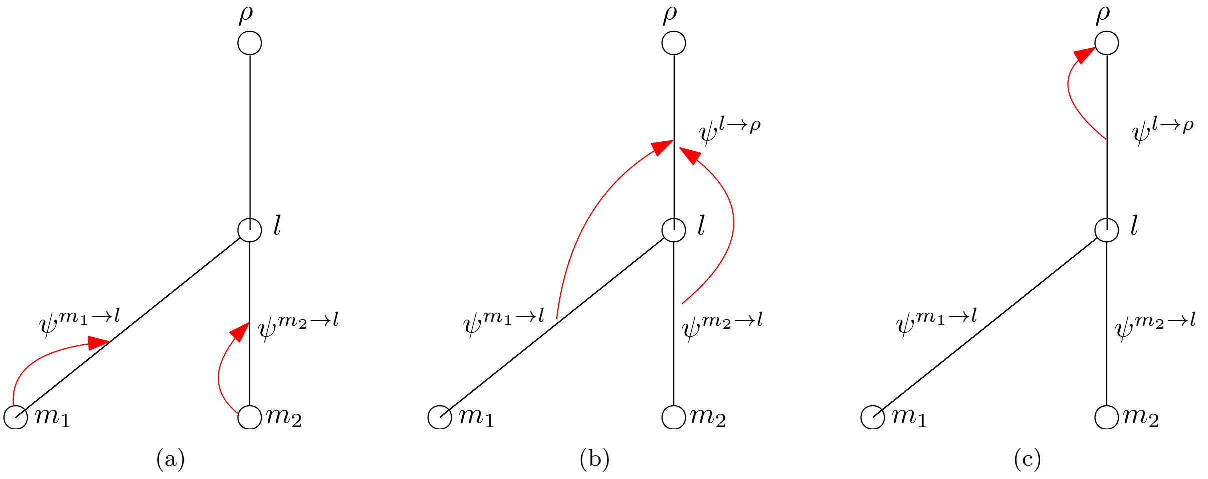


Figure 5.1: The iterative calculation of the marginal of the root $\rho$ when the type distribution of all other nodes is known, using messages. From left to right we see how every iteration step the messages one layer closer to $\rho$ are calculated, until $\nu_\rho$ can be calculated using the messages in the final iteration. In (a) we see the messages of the leaves $m_1$ and $m_2$ to their parent are simply their (known) type distributions. In (b) the marginal probability of $l$ in the subtree $T_l$ is calculated and stored in the message $\psi^{l\to\rho}$. In (c) the marginal of $\rho$ in $T$ can be calculated by calculating the marginal of $\rho$ in the smaller tree containing only $\rho$ and its neighbours $r$ where all neighbours have type distribution $\psi^{r\to\rho}$.

In general, one iteration a belief propagation algorithm is analogue to the previous calculation using messages, except that nodes $u$ send a message to every neighbour $v$. Note that this way, the message $\psi^{u\to v}$ is directed from $u$ to $v$ and works on a directed arc $\{u, v\}$. This way every edge $(u, v) \in E$ will contains 2 messages, one in each direction. When an iteration no longer significantly changes the messages, the algorithm is said to have converged. Important to note is that in this calculation three choices need to be made.

1. First we chose that the messages represented type distributions of node $u$ in $T_{u\backslash v}$. We could have also chosen a magnetisation or something else, as long as messages can be calculated using the messages of neighbours and the objective of the algorithm (in our case the marginals) can be calculated using the converged messages.

2. The second choice was to choose the initial messages. In the previous case they were intuitive, as what we wanted to represent was already known. This is not the case for all bp-algorithms we will see.

3. The third choice was the order in which the messages are updated. We will show that on trees the order of the messages heavily impacts the convergence time. On graphs the order is even more important, as it can impact the final solution of the algorithm.

As we have seen throughout this section, many of these arguments come from the exactness of bp-algorithms on trees. Thus we will spend the next section proving this and gaining more intuition for these algorithms.

## 5.2 Exact belief propagation on trees

In this section we show that all marginal probabilities of a Potts model can be calculated iteratively simultaneously on trees. We then show a specific belief propagation algorithm that calculates these marginals exact with finite convergence time on finite trees. We use an algorithm with a specific order to emphasize the importance of the order in which the messages are calculated with respect to the convergence time.

In order to show that marginals can be calculated iteratively, we use theorem 5.2.1, which comes directly from [8]. In an Ising model on trees $T$ where every node has its own magnectic field, as defined in equation (3.4), the marginal of a subtree $U$ in $T$ is defined as

$$\nu_U^T(\mathbf{t}_U^*) = \sum_{\mathbf{t}|\mathbf{t}_U=\mathbf{t}_U^*} \mu_T(\mathbf{t}). \tag{5.3}$$

Here $\mathbf{t}_U$ is the vector consisting of all types $t_u \in \mathbf{t}$ with $u \in U$. $\mu_T$ is the Ising model defined on tree $T$. Now the theorem shows that $\nu_U^T(\mathbf{t}_U^*)$ can be expressed as a separate Ising model on $U$, where each of the leaves of $U$ has an adjusted magnetic field. This result helps us show that we can calculate the marginal of a single node $u$ itereatively, because we can start with $U = T$ and take $U$ smaller and smaller around node $u$.

**Theorem 5.2.1.** *Let $U$ be a subtree of a finite tree $T$, where $\delta_* U$ are vertices of $U$ connected to $W = T\backslash U$. Let $\mu_T(\boldsymbol{t})$ denote the Ising model distribution with $\beta > 0$ on $T$, where each vertex has its own magnetic field $B_u$, as defined in (3.4). For $u \in \delta_* U$ look at $T_u$, the maximal subtree $T_u$ of $W \cup \{u\}$ rooted at $u$. A visual representation is given in figure 5.2. Let*

$$\langle t_u \rangle_W = \sum_{\boldsymbol{t}|t_u=+} \mu_{T_u}(\boldsymbol{t}) - \sum_{\boldsymbol{t}|t_u=-} \mu_{T_u}(\boldsymbol{t}) \tag{5.4}$$

*be the expected value of the type of this root $u$ in $T_u$.*

*Then the marginal of $U$ of the Ising measure on $T$, denoted $\nu_U^T$, is then also an Ising measure on $U$ with another magnetic field $B'_u = \frac{1}{\beta}\operatorname{arctanh}(\langle t_u \rangle_W)$ for $u \in \delta_* U$*

*Proof.* This proof was given by the authors of [8]. We give it here with slightly more explanation, as it illustrates the key point to why belief propagation is exact on trees. Furthermore it shows the way the magnetic field is connected to the type distribution.

This proof will have two main steps. In the first step we show that we can write the marginal using only vertices in $U$. In the second step we rewrite this marginal as the usual distribution of a full tree on an Ising model, with a different magnetic field.

We want to calculate the effects of $T_u$ on $u$, for all $u \in \delta_* U$. For this we calculate the marginal probabilities of $u$ in the subtree of $T_u$, defined as (5.1). Since $U$ is a subtree of $T$, all subtrees $T_u$ are disjoint. Therefore the vertices of different $T_u$ are not connected, thus their Ising measures are independent. Thus the joint probability of the type distribution of all $u \in \delta_* U$ in their subtree $T_u$ equals the product of their type distributions $\mu_{T_u}(t_u)$. Let $\mathbf{t}_U$ denote a state of all vertices $u \in U$, then

$$\nu_U^T(\mathbf{t}_U) = \frac{1}{\hat{Z}} \exp\{\beta \left( \sum_{uv \in U} t_u t_v + \sum_{u \in U \setminus \delta_* U} B_u t_u \right)\} \prod_{u \in \delta_* U} \mu_{T_u}(t_u). \tag{5.5}$$

Now we want to write this as a marginal of $U$ as the usual Ising distribution on a tree $U$. Thus we need to find a function $B'(u)$ such that

$$\prod_{u \in \delta_* U} \mu_{T_u}(t_u) = c_u \exp[\beta \sum_{u \in \delta_* U} B'(u) t_u] \tag{5.6}$$

with $c_u \in \mathbb{R}^+$ constants. This is possible since $\mu_{T_u}(t_u) > 0$.

For a single node $T_u = u$, the expected value $\langle t_u \rangle$ behaves as $\tanh(\beta B_u)$, so

$$\frac{1}{\beta}\operatorname{arctanh}(\langle t_u \rangle_{\{u\}}) = \frac{1}{\beta}\operatorname{arctanh}(\frac{\exp\{\beta(B_u)\} - \exp\{\beta(-B_u)\}}{\exp\{\beta(B_u)\} + \exp\{\beta(-B_u)\}}) = \frac{1}{\beta}\operatorname{arctanh}(\tanh(\beta B_u)) = B_u. \tag{5.7}$$

The magnetic field for bigger trees will no longer satisfy this relation, but using constants $c_u$ can be expressed in a similar way.

Since the $\operatorname{arctanh}(\langle t_u \rangle)$ is an odd function in $\langle t_u \rangle$, multiplying it with $t_u = \operatorname{sign}(\langle t_u \rangle)$ will turn it into an even function with range $[0, \infty)$. Since $\mu_{T_u}(t_u)$ is a probability it has range $[0, 1]$. As both functions are $\geq 0$ we can find a constant $c_u \in \mathbb{R}^+$ such that

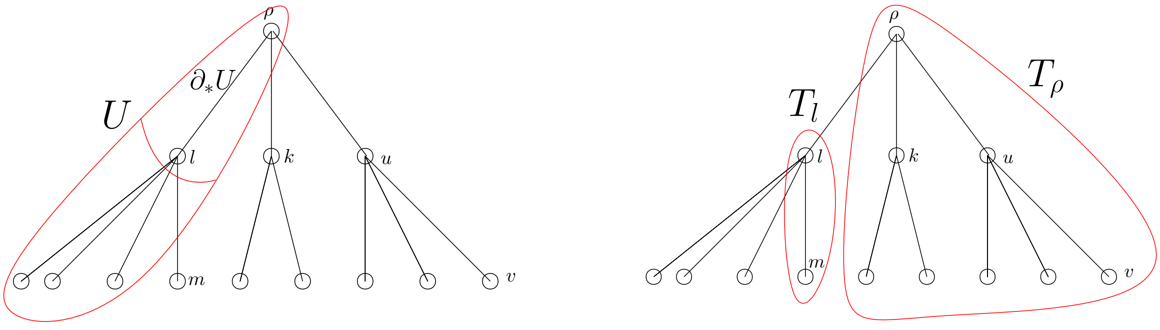
$$\mu_{T_u}(t_u) = c_u \exp(\beta \frac{1}{\beta} \operatorname{arctanh}(\langle t_u \rangle_W) t_u). \tag{5.8}$$

By incorporating the constants $c_u$ into the partition sum $Z$ such that $\hat{Z} = Z \prod_{u \in \partial_* U} c_u$ we find that $\beta B'_u = \operatorname{arctanh}(\langle t_u \rangle_W)$ and we can write

$$\nu_U^T(\mathbf{t}_U) = \frac{1}{\hat{Z}} \exp\{\beta \left( \sum_{uv \in U} t_u t_v + \sum_{u \in U \setminus \delta_* U} B_u t_u + \sum_{u \in \delta_* U} B'_u t_u \right)\} \tag{5.9}$$

which proves our claim. □

In statistical physics terms, the theorem tells us that if we consider an Ising measure on a tree, we can exactly construct the Ising measure on any subtree $U$ by integrating out the degrees of freedom not included in the subtree $U$. The netto effect of doing this amounts to adding a local magnetic field to the nodes of the subtree $U$ that connect it to the rest of the tree, $\partial_* U$.



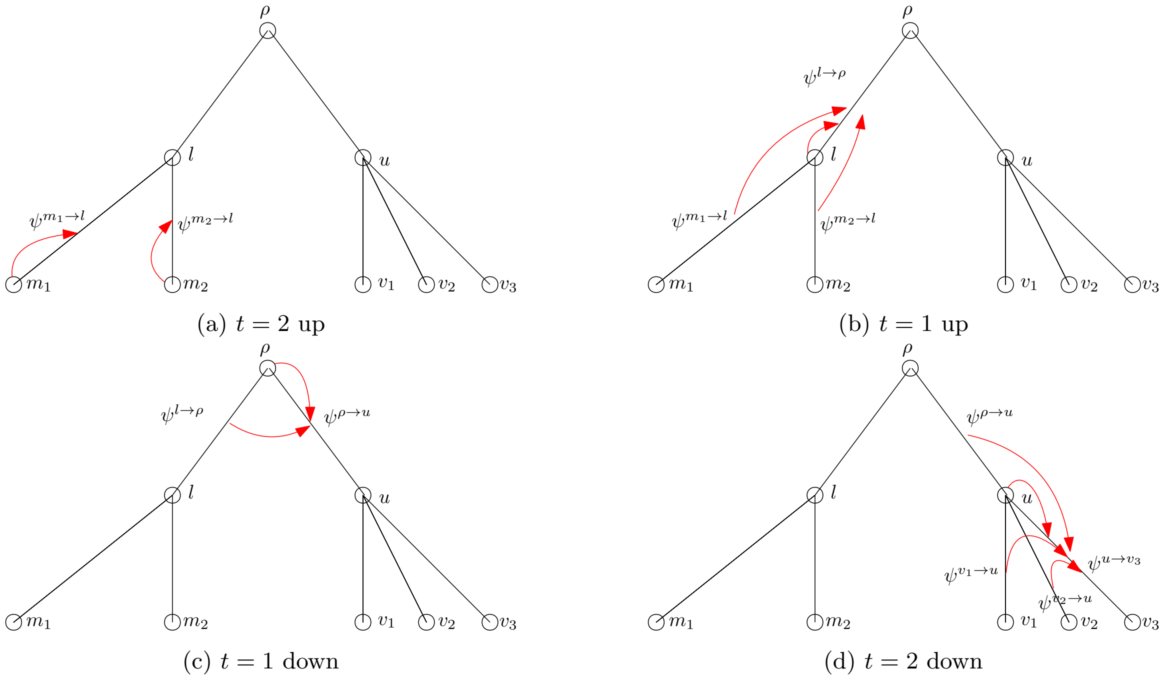
(a) $U$

(b) $T_\rho$ and $T_l$

Figure 5.2: Our favourite small tree $T$ with a subtree $U$, for which we can calculate the marginal probabilities of $U$ in the Ising model (3.4) using theorem 5.2.1

The marginal of a single node $u$ in the Ising model on $T$ can be reduced to the marginal of $u$ on an Ising model on the subgraph containing $u$ and its neighbours $U = \{u\} \cup \delta u$. This can also be done by starting with $U = T$ and then iteratively reducing $U$ to $\{u\} \cup \delta u$ vertex by vertex by removing a leaf every reduction. Essentially this is what a bp-algorithm does. We work out such a bp-algorithm in the rest of this section.

Let us now consider our first specific bp-algorithm on a rooted tree $(T, \rho)$, which we will denote as bp-ordered. We use $\phi$ instead of $\psi$ for messages to better distinguish which algorithm is used. Bp-ordered calculates the magnetic field $B'_v$ that a node $u$ receives from a subtree rooted at neighbour $v$ in a message $\phi^{v \longrightarrow u}$, by only considering neighbours $\neq u$. Then $B'_v \equiv \operatorname{arctanh}(\langle t_v \rangle_W)$ with $W = \delta v \setminus \{u\}$ and the magnetic field of the neighbours $k \in W$ considered is $\phi^{k \longrightarrow v}$. This can be compared to the reduction of $U = T$ to $U = \{u\} \cup \delta u$, when the messages of leaves are calculated first and then the messages are calculated inwards. Theorem 5.2.1 then shows that when calculating the messages in this order, this leads to the exact marginal of a specific node $u$.



(a) $t = 2$ up

(b) $t = 1$ up

(c) $t = 1$ down

(d) $t = 2$ down

Figure 5.3: Some schematic figures of the calculation of the messages at time step $t$. We did not draw all arrows to avoid a crowded figure, but each leaf at the same distance from $\rho$ is treated equally. In figure (a) we see that the initial messages are recalculated using only the magnetic field of the leaves. In the other figures we see that the calculation of the new message is done using only the incoming messages and the magnetic field of the node in between. Remember that during one time step $t$ all messages at depth $t$ are calculated.

Consider rooted tree $(T, \rho) = T_\rho = (V, E)$, with at each node $u \in V$ magnetic field $B_u$. Now let depth$(T_\rho)$ be the maximal distance from any vertex to the root $\rho$. Let $U_{u,v}$ denote the subtree $U = (V = \{u\} \cup \{\partial u \setminus \{v\}\}, E = \{(u,k) | k \in \{\partial u \setminus \{v\}\})$ consisting of $u$ and its neighbours except $v$. As initial messages we choose $\phi^{u \longrightarrow v} = B_u$. A message $\phi^{u \longrightarrow v}$ is only updated once all messages in $T_{u \setminus v}$ are updated. This is possible by, in a rooted tree, starting at all the leaves, updating upwards towards the root and then updating back down starting from the root, as visible in figure 5.3.

**Algorithm 1** ordered algorithm similar to bp exact on trees

```
procedure ORDERED-MARGINAL-CALCULATION($T_\rho, \beta, \vec{B}, \text{depth}(T_\rho)$)
    Start by setting $\phi^{u \longrightarrow v} = B_u$ for all $u \in V$
    $t = \text{depth}(T_\rho)$
    while $t > 0$ do
        For every node $u$ at distance $t$ with parent $v$, update $\phi^{u \longrightarrow v}$ using

        $$\begin{aligned} \phi^{u \longrightarrow v} &= \frac{1}{\beta} \operatorname{arctanh}(\langle t_u \rangle_{U_{u,v}}) \\ \langle t_u \rangle_{U_{u,v}} &= \sum_{\mathbf{t}|t_u=+1} \frac{1}{Z_{U_{u,v}})} \exp\{\beta(\sum_{k \neq u} [t_k t_u + \phi^{k \longrightarrow u} t_k] + \phi^{u \rightarrow v} B_u t_u)\} \\ &- \sum_{\mathbf{t}|t_u=-1} \frac{1}{Z_{U_{u,v}})} \exp\{\beta(\sum_{k \neq u} [t_k t_u + \phi^{k \longrightarrow u} t_k] + \phi^{u \rightarrow v} B_u t_u)\}. \end{aligned} \tag{5.10}$$

        In the right-hand side of this calculation we use magnetic field $B_u$ for vertex $u$ and magnetic field $B'_u = \phi^{k \longrightarrow u}$ for $k \in \{\partial u \setminus \{v\}$.;
        $t = t - 1$;
    while $t \leq \text{depth}(T_\rho)$ do
        For every node $u$ at distance $t$ with child $v$, update $\phi^{u \longrightarrow v}$ using (5.10). In the right-hand side of this calculation we again use magnetic field $B_u$ for vertex $u$ and magnetic field $B'_u = \phi^{k \longrightarrow u}$ for $k \in \{\partial u \setminus \{v\}$.;
        $t = t + 1$;
    for $u \in V$ do
        calculate $\phi^u(t) = \sum_{\mathbf{t}|t_u=t} \hat{\mu}_{u \cup \delta u}(\mathbf{t})$. In this Ising model $\hat{\mu}$ we again use magnetic field $B_u$ for vertex $u$ and magnetic field $B'_u = \phi^{k \longrightarrow u}$ for $k \in \{\partial u\}$.;
```

Now we show that bp-ordered can calculate the marginal probabilities of all nodes $u \in T$ simultaneously. We emphasize the impact of the order on the convergence time by showing that with the right order, the algorithm converges after a single iteration of both loops! Thus we need to prove that a bp-algorithm converges to a fixed point and that the steady-state is in fact all the marginal probabilities. For this next lemma we make use of theorem 5.2.1.

**Lemma 5.2.2** (Marginals are a fixed point of bp-ordered (on a tree))**.** *Suppose for every message we have that $\phi^{v \longrightarrow u} = B'_v = arctanh(\langle t_v \rangle_{T_{v \setminus u}})$, with $T_{v \setminus u}$ the subtree rooted at $v$ excluding $u$, then this will be a fixed point of bp-ordered.*

*Proof.* Let $u$ be a leaf, with parent $v$, then in the next iteration we find

$$\begin{aligned} \phi^{u \longrightarrow v} &\equiv \frac{1}{\beta} \operatorname{arctanh}(\langle t_u \rangle_{\{u\}}) = \\ &= \frac{1}{\beta} \operatorname{arctanh}(\frac{\exp\{\beta(\phi^{u \rightarrow v})\} - \exp\{\beta(-\phi^{u \rightarrow v})\}}{\exp\{\beta(\phi^{u \rightarrow v})\} + \exp\{\beta(-\phi^{u \rightarrow v})\}}) = \frac{1}{\beta} \operatorname{arctanh}(\tanh(\beta \phi^{u \rightarrow v})) = \phi^{u \rightarrow v}. \end{aligned} \tag{5.11}$$

Let $u$ not be a leaf, with neighbour $k$. Let $W_2 = \delta u \setminus \{k\}$. We use the result of theorem 5.2.1, with $T = T_{v \setminus u}$ and $U = \{u\} \cup \partial u$, which shows that when $B'_u$ is the marginal of $u$ in $T$, it can be calculated using the marginals of $k \in \partial u$ in their respective subtree $T_{k \setminus u}$, which is precisely what $\phi^{k \rightarrow u}$ is.

□

Now that we know marginals are a fixed point we can show that we reach this fixed point in one iteration.

**Lemma 5.2.3.** *Bp-ordered calculates the exact marginal probabilities of all nodes on a tree and converges after one iteration of both while loops.*

*Proof.* With this order of updates no message $\phi^{u \longrightarrow v}$ is ever updated before all of the messages it depends on are. Thus, after both while loops, we have $\phi^{u \longrightarrow v} = \langle t_u \rangle_{T_{u \setminus v}}$ with $T_{u \setminus v}$ the maximal subtree of $u$ that does not include $\{v\}$, as is desired. The marginals $\phi^u$ are calculated after the while loops and therefore only use these correct marginals. Theorem 5.2.1 with $U = \{u\} \cup \partial u$ tells us that indeed $\phi^u$ is the exact marginal probability of $u$ in $T$ □

Thus using this specific order, the algorithm reaches a fixed point in one iteration. We call this converging in one iteration. On graphs with cycles, there are no well-defined leaves and thus bp-ordered cannot be defined. Thus a different order must be chosen. One order that is used much in practice is an order chosen uniformly at random each iteration. To be more precise, instead of the two while loops producing the order as visualized in figure 5.3, the $2|E(T)|$ messages of $T$ are updated in a random order. Then the updating cycle is run again until a fixed point is reached. We investigate the convergence speed of the random order on trees.

**Corollary 5.2.4.** *Let $T = T(V, E)$, then $diam(T) = max_{u,v \in V} d(u, v)$. Then updating all messages as in bp-ordered, but in a uniform random order each iteration, called bp-random, calculates the marginal probabilities exactly and converges (almost surely) after at most $diam(T) - 1$ iterations.*

*Proof.* All messages are defined on an arc $u \to v$. Now define the in-distance $d_{\text{in}}$ of $\phi^{u \longrightarrow v}$ as the depth of the maximal subtree $T_{u\setminus v}$. Now we use induction.

Before the algorithm starts, we have $\phi^{u \longrightarrow v} = B_u$ for all $u \in V$. This is the correct expression for all nodes with $in - distance$ 0. Then the induction hypothesis is that

$$\forall \phi^{u \longrightarrow v} : d_{\text{in}}(\phi^{u \longrightarrow v}) \leq k \text{ it holds that } \phi^{u \longrightarrow v} = \langle t_u \rangle_{T_{u\setminus v}}. \tag{5.12}$$

Now let $\phi^{u \longrightarrow v} = \langle t_u \rangle_{T_{u\setminus v}}$ on all arcs with $in - distance \leq k$, then after one iteration of the algorithm, every arc is updated in a random order. Regardless of the order, all arcs with $in - distance \leq k$ are still correct as their calculation only involves other arcs with $in - distance \leq k$, which are a fixed point by Lemma 5.2.2. Furthermore all arcs with $in - distance$ $k + 1$ were updated. Those messages only use messages with $in - distance \leq k$ which satisfy $\phi^{u \longrightarrow v} = \langle t_u \rangle_{T_{u\setminus v}}$. Thus Lemma 5.2.2 shows that $\phi^{u \longrightarrow v} = \langle t_u \rangle_{T_{u\setminus v}}$ for all $\phi^{u \longrightarrow v}$ with in-distance $\leq k + 1$.

Now as edge $(u, v)$ is not part of subtree $T_{u\setminus v}$, all arcs in $T$ have $in - distance \leq diam(T) - 1$. Therefore the algorithm converges to the exact marginals after $diam(T) - 1$ iterations. □

Thus we have seen that belief propagation algorithms can calculate the exact marginals on trees in a linear amount of time. Even when a random order is used, the calculation takes at most quadratic time since $diam(T) \leq n$. This is much better than the exponential $\mathcal{O}(2^n)$ that the straightforward calculation of the marginals gives us. As we have seen in section 2.3, many random graph models are locally treelike, and thus belief propagation is intuitively thought to be a good approximation for the true marginals when $n$ is large. This is mostly because cycles are large and therefore the subtree of a message is for a direct calculation almost independent of itself. In the next section we explain a community detection algorithm which uses belief propagation on treelike graphs.

## 5.3 Decelle et al.'s belief propagation algorithm

Decelle et al. [1] use a bp-algorithm based on the intuition of calculating marginals on trees we have just seen. In this section we explain how they got to their update rules and why intuitively this algorithm correctly calculates marginals on treelike graphs. We then explain line-by-line how this algorithm functions. We will try to also explain intuitively why this algorithm should give results that detect communities. To remain consistent with notation across this thesis we have re-indexed some variables.

Let there be a graph $G = (V, E)$ of $n$ nodes generated from the IRG, so let $c_{i,j} = np_{i,j}$ with $p_{i,j}$ the probability of a node of type $i$ connecting to a node of type $j$. Let $G$ have adjacency matrix $A_{u,v}$. $q$ is the amount of types, $n_i$ the expected fraction of nodes of each type. During this section we shall talk about the diffent ideas of the algorithm, referring to the specific line in the pseudocode of BPAD, algorithm 2.

First we start with the philosophy of the algorithm. The algorithm does not intend to compute exact marginals given initialization, but rather compute a set of marginals that maximize the probability of $G$ being generated from an IRG with connecting probabilities $c$. Thus these are marginals that correspond to a different measure with a different Hamiltonian. Then, if the estimated marginals of type $i$ are different from $n_i$ for any vertex $u$ and type $i$, there will be positive overlap as defined in (2.41). Thus communities will have been successfully detected. The state where the estimation of the marginals $\nu_u(i) = n_i$ for all $u, i$ will be a fixed point of the algorithm and will be denoted *the trivial fixed point.*

We will now discuss the three different choices that can be made in a bp-algorithm.

**initialization** As the aim is no longer coupled to the initial state, the importance of the choice of the initialization decreases. Now the only remaining impact of the initialization is to which fixed point of the messages it converges. As we will see in section 7.2, there are regimes where different initializations lead to different fixed points. In section 6.4 we show that we can locally compute a free energy density (which was defined in (3.8)), which the system minimizes at constant $\beta$. In section 6.5 we show how Decelle et al. calculate the free energy density in this specific case. Thus the quality of the fixed points can be compared, where the best fixed point has the lowest free energy density. The aim is therefore to find as many fixed points as possible.

Thus one should try many different initializations. Decelle et al. therefore choose to initialize with random (normalized) marginals in line 2.

For the initial messages, only the word random is used to describe the initialization. Uniform randomness is usually implied in such a statement in physics papers. For 2 types uniform randomness is nicely defined. Let $X$ uniformly random on $[0,1]$ then letting $t_1 = x$ and $t_2 = 1-x$ has the same distribution as $t_1 = (1/2 - 1/2u)$ with $U \in [-1,1]$ uniformly random distributed. Both these methods have clear symmetry and each state $(t_1,t_2)$ in $\Omega = \{\mathbf{t}| \sum_i t_i = 1\}$ has the same probability. When $q > 2$, a uniform distribution from all states $\mathbf{t} \in \Omega$ can be created by taking a random permutation $\pi$ of the $q$ types and then letting $t_{\pi(1)}$ be uniform in $[0,1]$, then $t_{\pi(k)}$ uniform in $[0, 1-\sum_{i:pi(i)<q-1} t_i]$ and finally $t_{\pi(q)} = 1 - \sum_{i:pi(i)<q} t_i$. Also the symmetric method needs $n-1$ random variables where each computations depends on the results of the random variables before it. This is quite elaborate to code for arbitrary $q$. However, the important thing this randomness is used for is to find different fixed point of the algorithm. Therefore uniformity is not essential. Thus we opted in our code for a distribution where each type $i$ at vertex $u$ is assigned weight $w_{u,i} \in [0,1]$ uniformly at random. This is then normalized per vertex into a type distribution. This distribution favours states which are close to the trivial fixed point over the uniform distribution. However, every different distribution can be chosen using this method. As this is also how this algorithm was implemented by the authors of [10] online we expect this is what Decelle et al. have implemented as well.

**order of updating messages** There are $2|E(G)|$ messages that need to be updated. The order of the computation of messages is taken uniformly at random from all $2|E(G)|!$ permutations each iteration, as can be seen in line 7. Now the authors of [1] do not explain why they have chosen for a random order, but we can think of a reason. When we look back at bp-ordered, algorithm 1, on a tree, we see that the order in which the messages are updated can change the amount of messages being reached by a message in that iteration from one to the entire graph. This is a property that persists for graphs with cycles, and a particular order that repeats could increase the reach of some messages over others, significantly. Thus the order is chosen randomly each time, in the hope that when there are enough iterations, by the law of large numbers, the average reach of all messages over all iterations is comparable. This way, as $n \to \infty$, the bias towards any direction vanishes. Thus the solution will only depend on the graph $G$ and the initial condition.

**messages** As mentioned before, the algorithm computes a set of marginals that maximize the probability of $G$ being generated from an IRG with connecting probabilities $c$. Thus these are marginals that correspond to a different measure with a different Hamiltonian. We shall now explain how Decelle et al. get this Hamiltonian.

**Finding the Hamiltonian** In the $IRG_n$, defined in section 2.2.2, when we know all the parameters $\theta = \{q, n_i, p_{ij} = \frac{c_{ij}}{n}\}$, we can write that the probability of a graph $G$ with adjacency matrix $A_{uv}$ and type distribution $\mathbf{t}$ given $\theta$ is

$$\mathbb{P}(G, \mathbf{t}|\theta) = \prod_{u \neq v} p_{t_u,t_v}^{A_{uv}} (1 - p_{t_u,t_v})^{1-A_{uv}} \prod_u n_{t_u} \tag{5.13}$$

Where $u \neq v$ denotes all unique pairs $u, v \in V(G) | u \neq v$. But we are interested in the probability of $\mathbf{t}$, given $\{G, \theta\}$. For this we can now write

$$\mathbb{P}(\mathbf{t}|G,\theta) = \frac{\mathbb{P}(G,\mathbf{t}|\theta)}{\sum_{\mathbf{t}} \mathbb{P}(G,\mathbf{t}|\theta)} = \frac{1}{Z} \exp[-(-\sum_u \log(n_{t_u}) - \sum_{u \neq v} [A_{uv} \log(\frac{c_{t_u,t_v}}{n}) + (1 - A_{uv}) \log(1 - \frac{c_{t_u,t_v}}{n})])]. \tag{5.14}$$

Since $G$ is known, we know

$$\sum_{u \neq v} [A_{uv} \log(\frac{c_{t_u,t_v}}{n}) = [\sum_{u \neq v} [A_{uv} \log(c_{t_u,t_v})] - |E(G)| \log(n). \tag{5.15}$$

Since $|E(G)| \log(n)$ is a constant which every probability is multiplied by, we can leave it out of the Hamiltonian, since the probabilities will be normalized and that leads to the same Boltzmann distribution as including it.

Thus we can write the distribution of $\mathbf{t}$ (5.14) as a Boltzmann distribution with $\beta = 1$ and Hamiltonian

$$H(\mathbf{t}|G,\theta) = -\sum_u \log(n_{t_u}) - \sum_{u \neq v} [A_{uv} \log(c_{t_u,t_v}) + (1 - A_{uv}) \log(1 - \frac{c_{t_u,t_v}}{n}). \tag{5.16}$$

For the rest of this chapter, let $\mu$ denote the Boltzmann distribution (3.1) with this Hamiltonian (5.16). Note that in (5.16), the last term is nonzero for pairs of vertices that are not connected in the graph. This means that there are contributions to the probability between unconnected nodes. This will make the problem unlike a tree.

**Defining messages and marginal probabilities on a complete graph** Just as we did in the previous chapter for the Ising model, the marginals of the Bolzmann distribution with this Hamiltonian can be calculated iteratively on trees. As there is no inherent magnetic field we get that messages are only dependent on messages. Instead of having messages contain the magnetization of their subtree as in (5.4), which contains the marginals of both types, we instead calculate the marginal of each of the $q$ types on that subtree separately. Thus $\psi_i^{u \to v}$ represents $\sum_{\mathbf{t}|t_u=i} \mu_{T_{u \setminus v}}(\mathbf{t})$, the current estimate of the type of $u$ when $v$ is excluded from $G$. The assumption for the update rule of the messages is that they are exact if $G$ is a tree. In that case the marginals can be computed by updating the marginals based on neighbours $Diam(G)$ times. The assumption that $G$ is a tree is not far-fetched for large $n$, as the local limit of $G$, an IRG, is a tree.

However as the Hamiltonian (5.16) is dependent on both neighbours and non-neighbours, at first the complete graph needs to be considered. This is of course not very treelike. Fortunately we shall see that the influence of all non-edges can be approximated without using messages, to retain the intuition that iterating these update rules approximate the exact marginals.

For this update, the same way as we updated messages in (5.10) from the Ising model, we get

$$\nu_i^{u \to v} = \sum_{\mathbf{t}|t_u=i} \mu_{T_{u \setminus v}}(\mathbf{t}) \approx \sum_{\mathbf{t}|t_u=i} \frac{1}{Z} \exp\{\log(n_i) + \sum_{u,k|k \in \partial u \setminus v} [A_{uk} \log(c_{i,t_k}) + (1 - A_{uk}) \log(1 - \frac{c_{i,t_k}}{n}) + \log(\nu_{t_k}^{k \to u})\}. \tag{5.17}$$

Where we use that locally we can better approximate the type distribution of a vertex $k$ excluding the presence of $u$ as $\psi^{k \to u}$ then the global estimation $n_{t_k}$. We find as update definition

$$\psi_{t_u}^{u \to v} = \frac{1}{Z_{uv}} n_{t_u} \prod_{k \in \partial u \setminus v} [\sum_{t_k} c_{t_u,t_k}^{A_{uk}} (1 - \frac{c_{t_u,t_k}}{n})^{1 - A_{uv}} \psi_{t_k}^{k \to u}]. \tag{5.18}$$

Then the marginal probability of $u$ is estimated using all its neighbours, as

$$\psi_{t_u}^{u} = \frac{1}{Z_u} n_{t_u} \prod_{k \in \partial u} [\sum_{t_k} c_{t_u,t_k}^{A_{uk}} (1 - \frac{c_{t_u,t_k}}{n})^{1 - A_{uv}} \psi_{t_k}^{k \to u}]. \tag{5.19}$$

**Approximating the messages and marginal probabilities for more tree-like graph behaviour** As mentioned before, every vertex $u$ is in theory connected to every vertex $v \neq u$. This does not only imply $\mathcal{O}(n^2)$ updates per iteration but also very non-tree-like behaviour. Luckily the influence of non-edges is very small and can be approximated globally without considering every non-edge.

Now we have the approximation that

$$\prod_{k \notin \partial u \setminus v} [1 - \frac{1}{n} \sum_{t_k} c_{t_u,t_k} \psi_{t_k}^{k \to u}] = \prod_{k \notin \partial u \setminus v} [\sum_{t_k} (1 - \frac{c_{t_u,t_k}}{n}) \psi_{t_k}^{k \to u}] + \mathcal{O}(\frac{1}{n}). \tag{5.20}$$

Thus when $(u, v) \notin E$ we can express a message $\psi^{u \to v}$ as

$$\psi_{t_u}^{u \to v} = \frac{1}{Z_{uv}} n_{t_u} \prod_{k \notin \partial u \setminus v} [1 - \frac{1}{n} \sum_{t_k} c_{t_u,t_k} \psi_{t_k}^{k \to u}] \prod_{k \in \partial u} [\sum_{t_k} c_{t_u,t_k} \psi_{t_k}^{k \to u}] = \psi_{t_u}^{u} + \mathcal{O}(\frac{1}{n}), \tag{5.21}$$

As this to leading order of $n$ only dependent on the type, Decelle et al. choose to approximate $\psi_{t_u}^{u \to v} \approx \psi_{t_u}^{u}$. With this approximation, for $(u, v) \in E$ we can write

$$\begin{aligned}
\psi_{t_u}^{u \to v} &= \frac{1}{Z_{uv}} n_{t_u} \prod_{k \notin \partial u} [1 - \frac{1}{n} \sum_{t_k} c_{t_u,t_k} \psi_{t_k}^{k \to u}] \prod_{k \in \partial u} [\sum_{t_k} c_{t_u,t_k} \psi_{t_k}^{k \to u}] \\
&\approx \frac{1}{Z_{uv}} n_{t_u} \prod_{k \notin \partial u} [1 - \frac{1}{n} \sum_{t_k} c_{t_u,t_k} \psi_{t_k}^{k}] \prod_{k \in \partial u} [\sum_{t_k} c_{t_u,t_k} \psi_{t_k}^{k \to u}]
\end{aligned} \tag{5.22}$$

When the IRG is sparse, so the degree distribution does not scale with $n$ (2.9), and $n$ is large, almost all nodes are not connected to $u$. Thus we can approximate

$$\prod_{k\notin\partial u}[1-\frac{1}{n}\sum_{t_k}c_{t_u,t_k}\psi^k_{t_k}]\approx\prod_k[1-\frac{1}{n}\sum_{t_k}c_{t_u,t_k}\psi^k_{t_k}]\approx\frac{1}{Z_{uv}}n_{t_u}\exp\{-\frac{1}{n}\sum_k\sum_{t_k}c_{t_u,t_k}\psi^k_{t_k}]\}. \tag{5.23}$$

Here we can identify an auxiliary external field of type $t_u$,

$$h_{t_u}\equiv\frac{1}{n}\sum_k\sum_{t_k}c_{t_u,t_k}\psi^k_{t_k}. \tag{5.24}$$

This field is used for the calculation of the messages and the marginals and therefore updated every message update in line 10. For the messages we then find

$$\psi^{u\to v}_{t_u}\approx\frac{1}{Z_{uv}}n_{t_u}\exp\{-h_{t_u}\}\prod_{k\in\partial u}[\sum_{t_k}c_{t_u,t_k}\psi^{k\to u}_{t_k}]. \tag{5.25}$$

This is taken as the definition of the update rules in BPAD in line 7. As the influence of non-edges are incorporated using the auxiliary field, only along edges the algorithm will need to update. For a sparse graph and large $n$ this results in $\mathcal{O}(n)$ computations per iteration and a locally treelike graph. Using the same approximation on $\psi^u_{t_u}$ results in

$$\psi^u_{t_u}\approx\frac{1}{Z_u}n_{t_u}\exp\{-h_{t_u}\}\prod_{k\in\partial u}[\sum_{t_k}c_{t_u,t_k}\psi^{k\to u}_{t_k}]. \tag{5.26}$$

This is taken as the update rule for the marginals in line 9. For the updating of the messages the auxiliary field needs to be initiated. This is done by computing $\psi^u_{t_u}$ with the initialized messages and the expected auxiliary field $h_{t_u}=n_{t_u}$ and then computing $h_{t_u}$ using these initialized marginals in line 3 and line 4.

We have seen how the auxiliary field was derived from the definition of the IRG and approximations for large treelike graphs. We discuss here how to intuitively explain the function of this field. When we know the group fractions $n_i$, the field $h$ pushes the messages towards a state where the average of the marginal of all nodes $u$ being type $i$, $\psi^u_i$, is close to $n_i$. This is done because for $n\to\infty$, due to the law of large numbers, this is where the types of $G$ are very likely to be, even though it is only a small part of the solution space.

**convergence** Here we discuss when the algorithm decides to stop iterating. The idea is that the iteration only stops once a fixed point is reached. As the calculations are done numerically, being a fixed point is checked by having marginal probabilities which do not change significantly anymore. How significantly they are still allowed to change is determined by the parameter ‘crit’. In order to exclude the edge-case where the algorithm does not reach a fixed point or takes very long to get close to one, $t_{\max}$ is a very large number which is the maximal number of iterations. In line 6 we can see that an iteration starts when $conv > crit$ and $t < t_{\max}$, which is true for the first iteration due to line 5. The parameters $t_{\max}\in\mathbb{N}^+$ and crit $\in\mathbb{R}^+$ are inputs of the algorithm. In practice $t_{\max}$ will be chosen very large and crit will be chosen as a small number $> 0$, such that $\frac{\text{crit}}{n}$ is still much larger than numerical errors. This is necessary, as we can see in line 8 that conv is the sum of $2|E(G)|$ update differences $|\psi^{u\to v}_{\text{new}}-\psi^{u\to v}_{\text{old}}|$, and we want that conv $<$ crit when the algorithm reaches close to a fixed point, so that it stops iterating.

Now there are two things from the algorithm that have not been discussed yet. The first is the computation of the free energy, which will be discussed in section 6.5. The second is that groups are guessed using the argument that maximizes their marginal probability as described in (5.2) and that communities have been successfully detected if the overlap described in (2.41) is $Q > 0$.

In the next section we discuss when this algorithm successfully detects communities. The full algorithm 2 in pseudocode can be found below.

**Algorithm 2** bp algorithm used to infer parameters, BPAD

```
procedure BP-INFERENCE($q, n_i, c_{i,j}, A_{u,v}, crit, t_{max}$)
    initialize a random q-component normalized vector $\psi^{u \longrightarrow v}$ for each edge $(u, v)$;
    For each node $u$ and type $t_l$ compute estimation $\psi^u_{t_l} = \frac{1}{Z^u} n_{t_u} e^{-h_{t_u}} \prod_{v \in \delta u} (\sum_{t_v} c_{t_v t_l} \psi^{v \longrightarrow u}_{t_v})$ with $h_{t_u} = n_{t_u}$;
    compute for each type $t_l$ auxiliary field $h_{t_l} = \frac{1}{N} \sum_k \sum_{t_k} c_{t_k t_l} \psi^k_{t_k}$;
    $conv \leftarrow crit + 10$; $t \leftarrow 0$;
    while $conv > crit$ and $t < t_{max}$ do; $conv \leftarrow 0$; $t \leftarrow t + 1$;
        for every edge $(u, v)$ in a random order do; update all $q$ components of $\psi^{u \longrightarrow v}$ using $\psi^{u \longrightarrow v}_{t_u} = \frac{1}{Z^{u \longrightarrow v}} n_{t_u} e^{-h_{t_u}} \prod_{k \in \delta u \setminus v} (\sum_{t_k} c_{t_k t_u} \psi^{k \longrightarrow u}_{t_k})$;
            $conv \leftarrow conv + |\psi^{u \longrightarrow v}_{new} - \psi^{u \longrightarrow v}_{old}|$;
            Update $\psi^v$ using the new value of $\psi^{u \longrightarrow v}$;
            Update h (by subtracting the old $\psi^v$ and adding the new value);
    compute the free energy $f_{bp}(q, \{n_a\}, \{c_{ab}\}) = \frac{-1}{N} \sum_i \log Z^u + \frac{1}{N} \sum_{(u,v) \in E} \log Z^{uv} - \frac{c}{2}$
    return free energy density $f_{bp}$, converged messages $\{\psi^{u \longrightarrow v}\}$, guessed group assignment $q^*_u = argmax_i \psi^u_i$, overlap $Q = \frac{\frac{1}{N} \sum_u \psi^u_{q*u} - max_i n_i}{1 - max_i n_i}$;
```

## 5.4 Intuition behind community detection on random graphs

In this section we show when the algorithm from Decelle et al. produces successful community detection.

As a thought experiment we first look at what happens when we apply BPAD to a tree $T$. Of course this does not make much sense, as the algorithm is designed for IRG's but we will gain some insight from it. There are no cycles on a tree. As all the messages are directed, no message has any influence on itself in any of the future iterations. Thus eventually all messages will be determined by the messages of all leaves which lead to said message. In case of $n_i = \frac{1}{q}$ each of the leaf messages will be distributed as $\frac{1}{Z} e^{-h_i}$. As the auxiliary field pushes towards a distribution closer to uniform, eventually all leaf messages will be uniformly distributed. As any update of messages with only uniformly distributed incoming messages is itself uniformly distributed, this leads to convergence to the fixed point where every message and thus every marginal is uniformly distributed and therefore we know nothing about the communities.

On graphs with cycles however, a message will affect itself by affecting messages along this cycle back towards itself. When this effect is strong enough, it allows the stability of states where the type distribution of individual nodes is not dominated by the auxiliary field. This means BPAD will converge to a useful solution and community detection is possible.

Thus a definition of stability of fixed points is required.

**Definition 5.4.1** (stability fixed point)**.** *Let $\boldsymbol{t}^*$ be a fixed point of all message update equations (5.25) and therefore of BPAD. Then it is called stable if after a small perturbation in $\mathbf{t}^*$, BPAD converges back to $\boldsymbol{t}^*$ and unstable otherwise.*

Now the trivial fixed point, the fixed point where every message has their type distributed according to the sizes of the types,

$$\forall u, v \in V(G) : \psi^{u \longrightarrow v}_i = n_i, \tag{5.27}$$

is a fixed point. This is precisely the fixed point that gives us no successful community detection. As soon as the trivial fixed point is unstable, BPAD converges to some other fixed point and there is successful community detection. We will paraphrase the authors of [1] to show when the trivial fixed point becomes unstable.

#### intuition stability condition trivial fixed point

**Intuition 5.4.1** (criterion instability trivial fixed point)**.** *Let $G$ be a sparse (2.9) IRG with kernel $c$, $c_{ij} = \mathcal{O}(1)$ and $n_i$ the probability of a node having type $i$. Let $d = \sum_{i=1}^q (n_i \sum_{j=1}^q c_{ij})$ the average degree.*

*Let $L$ be a matrix with $L_{ij} = n_i(\frac{c_{ij}}{d} - 1))$. Let $\lambda$ be the largest eigenvalue of $L$.*

*Then the trivial fixed point is stable if $d\lambda^2 < 1$ and unstable if $d\lambda^2 > 1$.*

*Explanation.* We know from 2.3.9 that locally, the IRG converges to a tree. Thus there is an $R$ of $\mathcal{O}\log(n)$ s.t. the shortest loop a vertex $u$ belongs to is $> R$. Now we consider a specific subgraph which is such a tree

$(T, \rho)$ which contains $\rho$ and all vertices within distance $R$, excluding paths using one neighbour $v$, called the parent of $\rho$. We consider what happens when $R \to \infty$.

Now we perturb the trivial fixed point on all of the leaves of $(T, \rho)$ such that

$$\psi_i^{v \to u} = n_i + \epsilon_i^v. \tag{5.28}$$

Now we shall compute the influence of this perturbation on the root. When the influence dies out the fixed point will be stable, while when it gets stronger and stronger the fixed point will be unstable.

As the influence of leaves is independent, we first look at the influence a single leaf $k_R$. Let vertex $k_R$ have a path to the root $k_R, k_{R-1}, ..., k_1, \rho$. We now define a transfer matrix $L^r_{i,j}$ which shows the influence of the change in message $j$ of a node at depth $r+1$ to depth $r$ on the message of type $i$ of its parent on depth $r$ to depth $r-1$.

$$L^r_{i,j} \equiv \frac{\partial \psi^{k_r \to k_{r-1}}}{\partial \psi^{k_r+1 \to k_r}}|_{\psi_i = n_i} \tag{5.29}$$

$$= \left(\frac{\psi_i^{k_r \to k_{r-1}} c_{ij}}{\sum_l c_{il} \psi_l^{k_r+1 \to k_r}} - \psi_i^{k_r \to k_{r-1}} \sum_s \frac{\psi_s^{k_r \to k_{r-1}} c_{sj}}{\sum_l c_{il} \psi_l^{k_r+1 \to k_r}}\right)|_{\psi_i = n_i} \tag{5.30}$$

$$= n_i\left(\frac{c_{ij}}{d} - 1\right) + \mathcal{O}(1/n) \tag{5.31}$$

where the simplification was derived using (5.25). This does not depend on distance $r$. Therefore we can write $L_{ij}$ instead. Now the total perturbation of the message $\psi^{\rho \to v}$, $\epsilon^\rho$, due to the perturbation of message $\psi^{k_R \to k_{R-1}}$, $\epsilon^{k_R}$ can be written as

$$\epsilon^\rho = L^R \epsilon^{k_R}. \tag{5.32}$$

Let $\lambda$ be the largest eigenvalue of $L$. Then when $R \to \infty$, $L^R$ will be dominated by $\lambda^R$.

Now we go back to considering the influence of perturbing all leaves of $(T, \rho)$. When $\mathbb{E}[\epsilon^{k_R}] = 0$ we find $\mathbb{E}[\epsilon^\rho] = 0$. For the second moment, however, we get

$$< (\epsilon_{t_0}^{u_0}) > \approx < (\sum_{r=1}^{d^R} \lambda^R \epsilon_t^u)^2 > \approx (d\lambda^2)^R < (\epsilon_t^u)^2 > . \tag{5.33}$$

leading to the stated conjecture. This statement is not quite rigorous for finite $n$, as most of the calculations are approximations at finite $n$.

This corresponds to the Kesten-Stigum bound for community detection [11], which stems from 1966 and has been confirmed for $q = 2$. Decelle et al. empirically show that their algorithm does in fact detect communities successfully when $d\lambda^2 > 1$ and generally not when $d\lambda^2 < 1$. We discuss the exceptions where specific initial conditions still lead to community detection even for $d\lambda^2 < 1$ in section 7.2.

## 5.5 Relation of critical parameters

In the previous section we have seen under which conditions we can expect BPAD to detect communities. In this section we will relate this with detection results known on the SBM.

When we apply the SBM 2.2 to theorem 5.4.1, then we find

$$T^{ij} = \begin{cases} \frac{a-d}{dq} & \text{for } i = j \\ \epsilon = \frac{b-d}{dq} & \text{for } i \neq j \end{cases} \tag{5.34}$$

This matrix has eigenvalues $\lambda_1 = 0$ for eigenvector $(1, 1, ..., 1)$ and $\lambda_2 = \frac{a-b}{dq}$ with multiplicity $q-1$ for each eigenvector of the form $(0, ..., 0, 1, -1, 0, ..., 0)$. Thus the trivial fixed point is unstable when

$$d\lambda^2 = \frac{(a-b)^2}{dq^2} > 1 \tag{5.35}$$

$$(a-b)^2 > q^2 d = q^2(a + (q-1)b)/q = q(a + (q-1)b). \tag{5.36}$$

Thus when this criterion is satisfied BPAD successfully detects communities on the SBM. This is the same result that [4] get for the possibility of detection at $q = 2$. At higher $q$, it has been show that there can be no successful community detection at $(a - b)^2 < q(a + b)$, which is smaller than (5.36). This predicts that there is a regime where community detection is possible, but not easy using BPAD.

Thus we have seen in this chapter some examples of belief propagation. We have seen that on trees, belief propagation calculates marginal probabilities exact. On random graphs which are locally treelike, we can use this property to apply belief propagation in order to detect communities, by approximating the messages as done by [1]. This algorithm shows successful community detection when the trivial fixed point is unstable. This prediction matches the critical condition for detection found by rigorous mathematical analysis [4] for $q = 2$ and gives an upper bound for the critical condition of successful community detection for $q \geq 2$.

# Chapter 6

# Towards a mathematical description of belief propagation algorithms

In this section we will formulate a more mathematical description to describe belief propagation. This is done using a tool from computer science, factor graphs. This will be useful, as the result is that the partition function $Z$, and with it the free energy $F = -\frac{1}{\beta}\log(Z)$ can be calculated iteratively. This is useful as the free energy is minimized for constant $\beta$, which is the type of systems we look at. Thus the local free energy of multiple fixed points can be compared, and the fixed point with the best guess of the communities will be the fixed point with the lower local free energy.

In section 6.1 we define factor graphs and explain what we can do with them. In section 6.2 we define belief propagation on factor graphs which are trees. We show that also in this description there is a belief propagation which calculates marginals exact. In section 6.3 we show that on a tree, many physical properties properties of Boltzmann distributions can be calculated using only local quantities. In section 6.4 we show how using the same local computations on graphs lead to the cavity equations, which can compute the partition function iteratively. In section 6.5 we show how Decelle et al [1] adapt these equations. In section 6.6 we show how for some algorithms the tree approximation is exact on random graphs. However, this is not the case for BPAD.

## 6.1 Factor graphs

We introduce factor graphs, as introduced by the authors of [3], as they are a framework in which many properties of belief propagation have been proven rigorously.

Factor graphs are bipartite graphs, consisting of a group of variable nodes $u \in V$ which can have any type $i \in [q]$ and a group of constraints (factor nodes) $F$. Intuitively, each factor node $a \in F$ is connected to variable nodes $u \in V$, and forces a constraint on those neighbours through weight function $\psi_a$. The variable nodes $u$ can be seen as the regular nodes of graphs we have looked at until now. The factor nodes $a$ with its weight function $\psi_a$ represents the impact that the types of nodes $u$ adjacent to $a$, have on their joint probability. This is a generalization of the graphs we have seen before, in the sense that when every factor node has precisely two neighbours, we are back at the case where every factor node represents an edge $e$ in the regular graph. Thus using this representation we are able to incorporate the influence of non-neighbouring nodes!

**Definition 6.1.1** (Factor graph). *Let $[q]$ be a finite set of $q$ types. Then $G = (V, F, (\partial a)_{a\in F}, (\psi_a)_{a\in F})$ is a $[q]$-factor graph, where*

- *$V$ a set of variable nodes with $|V| = n$,*
- *$F$ a set of factor nodes with $|F| = m$,*
- *$\forall a \in F$ an amount of neighbours of $a$, $k(a)$, with order $\partial a = (\partial_1 a, ..., \partial_{k(a)} a) \in V^{k(a)}$, defining the edges of the factor graph*
- *a weight function $(\psi_a) : [q]^{k(a)} \to [0, \infty)\forall a \in F$.*

Factor graphs are designed to be able to calculate probabilities of the nodes $u \in V$ based on the weight functions and the graph structure.

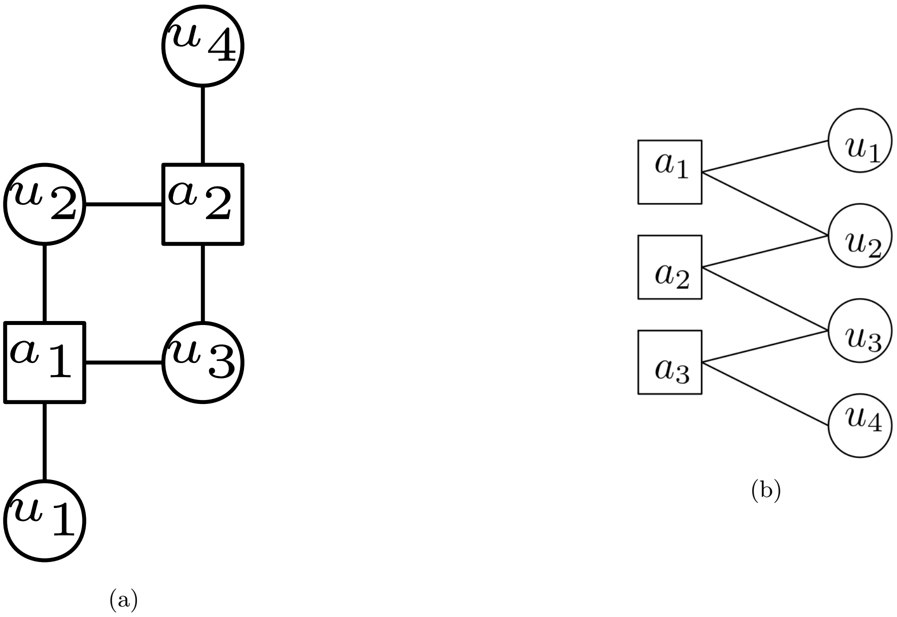


Figure 6.1: Two factor graphs. In (a) we have $\partial a_1 = (u_1, u_2, u_3)$ and $\partial a_2 = (u_2, u_3, u_4)$. In (b) we have arranged a factor graph in its bipartite representation. In this case we can identify every factor node as an edge, as every factor node has only 2 neighbours.

We are interested in the results on factor graphs because normal graphs can be written as factor graphs. Since the model of the community detection algorithm of the previous chapter was based on a Boltzmann distribution, we are interested in applying a Boltzmann distribution to factor graphs. The energy will be the product of the weight functions of all the nodes.

Let $t_u$ denote the type of $u$ in $[q]$. Let $\mathbf{t} = \{t_1, ..., t_n\} \in \Omega = [q]^n$ and $\mathbf{t}_{\partial a} = \{t_u | u \in \partial a\}$. Then the joint Boltzmann probability distribution of all nodes $u \in V$ is given by

$$\mu_G(\mathbf{t}) = \frac{1}{Z(G)} \prod_{a \in F} \psi_a(\mathbf{t}_{\partial a}) \tag{6.1}$$

With $Z(G) = \sum_{\mathbf{t} \in \Omega} \prod_{a \in F} \psi_a(\mathbf{t}_{\partial a}) > 0$ the normalizing partition function. We shall usually omit $G$ when it is clear which graph is being used. We call this a Boltzmann distribution, as we can rewrite the weight functions to the form

$$\psi(\mathbf{t}_{\partial a}) = \exp(-\beta E_a(\mathbf{t}_{\partial a})), \tag{6.2}$$

with $E_a(\mathbf{t}_{\partial a})$ the energy function characterizing constraint $a$. In this case (6.2) reduces back to a Boltzmann distribution (3.1). Such a rewriting is of the weight functions is possible when one allows that $E_a(\mathbf{t}_{\partial a}) = \infty$, to correspond with $\psi_a(\mathbf{t}_{\partial a}) = 0$.

We are again interested in the marginal probabilities, which can help us guess the type of a variable node. Thus we define

$$\mu_{G,u}(i) = \sum_{\mathbf{t} | t_u = i} \mu_G(\mathbf{t}) \tag{6.3}$$

as the marginal probability of type $i$ of variable node $u$.

## 6.2 Belief propagation on factor trees

As we have seen in section 5.2, when a regular graph is a tree $T$, the calculation of marginal of $v \in V$ can be reduced to the product of the marginals of all neighbours on $v$. Just as in section 5.2 we use messages on edges to calculate such marginals on trees. However, now there are two types of messages. First there are the messages $\nu_{u \to a}$ going from variable nodes $u$ to factor nodes $a$, and the messages $\hat{\nu}_{\to u}$ going from factor nodes to variable nodes. The choice we make for the messages is again to be equal to the subtree marginals. Identify $T_{u \backslash a}$

as the subtree rooted at variable $u$, which does not feature edge $\{u, a\}$ and the subtree rooted at $a$. Similarly let $T_{a\to u}$ be $\{u\} \cup G \setminus G_{u\setminus a}$.Then we define the messages

$$\nu_{u\to a}(i) := \mu_{T_{u\to a},u}(i) \tag{6.4}$$

$$\hat{\nu}_{a\to u}(i) := \mu_{T_{a\to u},u}(i). \tag{6.5}$$

As these are probabilities, we shall denote $\nu_{u\to a}$ and $\hat{\nu}_{a\to u}$ as the distribution of those probabilities. As all of these need to be normalized, we stop writing the normalization constant $Z$ explicitly and introduce a new notation $\cong$, which means 'equal up to normalization'

With $T$ being a tree, these messages again allow us to calculate the marginals of $T$ in an iterative way. Let $k$ denote the iteration number. Then the belief-propagation update rules (sometimes also called sum-product rules) are

$$\nu^{(k+1)}_{u\to a}(i) := \mu_{G_{u\to a},u}(i) \cong \prod_{b\in\partial u\setminus\{a\}} \hat{\nu}_{b\to u}(i) \tag{6.6}$$

$$\hat{\nu}^{(k)}_{a\to u}(i) := \mu_{G_{a\to u},u}(i) \cong \sum_{\tau\in\Omega^{\partial a}} \mathbb{1}\{t_u = i\}\psi_a(\tau) \prod_{v\in\partial a\setminus\{u\}} \nu_{v\to a}(i). \tag{6.7}$$

As the empty product is one, for leaves $u$, $\nu_{u\to a}$ is uniformly distributed. The iteration number $k$ is only updated after either the variable nodes or factor nodes have updated, but not both. Thus, in one iteration, first all variable nodes are updated and then all factor nodes are. Since the variable nodes depend only on factor nodes and vice versa, thus the specific order in which messages are updated no longer matters.

We let these messages run until a fixed point is reached, where we have $\nu^{(k)}_{u\to a} = \nu^*_{u\to a}$, $\hat{\nu}^{(k)}_{a\to u} = \hat{\nu}^*_{a\to u}$ which satisfy equations (6.6), (6.7).

After $k$ iterations we can estimate the marginal distribution $\mu_{T,u}$ using its incoming messages, using

$$\nu^{(k)}_{u}(i) = \prod_{a\in\partial u} \hat{\nu}^{(k-1)}_{a\to u}(i). \tag{6.8}$$

Now, exactly as in regular graphs, on trees $T$ the messages converge and all marginals of $T$ are calculated exactly.

**Theorem 6.2.1.** *Let $u, v \in V(T)$ have distance one when there is a factor node $a$ with $u, v \in \partial a$. Let $\nu^{(0)}_{u\to a}$ be any normalized initialization on all edges $\{u, a\} \in T$. Then using update rules (6.6), (6.7) leads to a fixed point after diam($T$) iterations. For this fixed point we have $\nu^{(k)}_{u}(i) = \mu_{T,u}(i)$.*

*Proof.* The proof is almost analogous to the proof of 5.2.4, where induction is used on the diameter of the tree $T$. The difference is that now it takes one iteration for the leaves to obtain the correct messages, which were correct in 5.2.4 by the initialization. Thus the resulting convergence time is 1 iteration slower. □

## 6.3 Properties in terms of local quantities

The calculation of physical quantities can help us grasp better what is going on in the system. We will see that on tree factor graphs, many physical quantities can be calculated locally. This is great news, as this allows us on graphs which are locally treelike to take the same local calculation as the estimate of that quantity.

Until now we have only considered the marginals of the variable nodes. We can also define the marginals of factor nodes, by considering all their adjacent variable nodes. Let

$$\mu_{G,a}(\mathbf{t}_{\partial a}) = \sum_{\mathbf{t}\in\Omega} \mu_G(\mathbf{t})\mathbb{1}\{\mathbf{t} = \mathbf{t}_{\partial a}\}. \tag{6.9}$$

It turns out that on trees, it is possible to express both the joint distribution of all variables $u$, $\mu_G(\mathbf{t})$, as well as the partition function $Z$ in terms of the marginals $\mu_{T,u}(i)$ and $\mu_{T,a}(\mathbf{t}_{\partial a})$. These marginals can be calculated with the messages $\nu_{x\to a}$ and $\hat{\nu}_{a\to x}$.

The authors of [3] show that

**Theorem 6.3.1** (local joint probaility distribution)**.** *Let $T$ be a finite factor graph tree. then*

$$\mu(\boldsymbol{t}) = \prod_{a\in F} \mu_a(\boldsymbol{t}_{\partial a}) \prod_{u\in V} \mu_u(t_u)^{1-|\partial u|} \tag{6.10}$$

The proof of Mezard et al. uses induction on the number of factors $M$. There is also an intuitive way to understand this. Each $\mu_a(\mathbf{t}_{\partial a})$ is the joint probability distribution of $\partial a$. When multiple factor nodes $a$ are connected to a variable node $u$, we are thus using the distribution of $u$ twice. Therefore we need to compensate in the joint distribution by dividing by the marginal distribution of $u$, $\mu_u(t_u)$. Thus we need to divide the joint distribution by $\mu_u(t_u)^{|\partial u|-1}$, which is the same as multiplying with $\mu_u(t_u)^{1-|\partial u|}$.

In particular we are interested in the free energy $F = -\frac{1}{\beta}\log Z = -\frac{1}{\beta}(S_\mu - \beta U_\mu)$, as this quantity is minimized under constant temperature. We shall now show the many steps that lead to expressing this quantity locally.

We start with the internal energy $U$. Using the rewriting of the weight functions in terms of an energy (6.2) and the definition of $U$ (3.5), we find

$$U = -\sum_{\mathbf{t}} \mu(\mathbf{t}) \sum_{a=1}^{M} \log(\psi_a(\mathbf{t}_{\partial a})). \tag{6.11}$$

We want to calculate $\mu(\mathbf{t})$ locally. This is possible using the fixed-point messages $\hat{\nu}^*_{a\to u}$ with

$$\mu(\mathbf{t}_a) = \frac{1}{Z_R}\psi_a(\mathbf{t}_{\partial a}) \prod_{a\in\partial\partial a} \hat{\nu}^*_{a\to u(a)}(t_{u(a)}). \tag{6.12}$$

Here $Z_a$ is the partition function and where $\partial\partial a$ represents the factor nodes connected to $a$ via only one variable node $u(a)$. Using this (6.12) and the definition of the messages from factor to variable nodes 6.7, we can rewrite the internal energy $U$ (6.11) as a locally computed quantity, as

$$U = -\sum_{a=1}^{M} \frac{1}{Z_a} \sum_{\mathbf{t}_{\partial a}} \psi_a(\mathbf{t}_{\partial a}) \log(\psi_a(\mathbf{t}_{\partial a})) \prod_{u\in\partial a} \nu^*_{u\to a}(t_j). \tag{6.13}$$

Also the entropy can be rewritten as a local quantity. For this we use the definition of entropy (3.6) and the local expression of the joint probability function (6.10) to find

$$S_\mu = -\sum_{a\in F} \mu_a(\underline{t}_{\partial a}) \log(\mu_a(\underline{t}_{\partial a})) - \sum_{u\in V} (1 - |\partial u|)\mu_u(t_u)\log(\mu_u(t_u)). \tag{6.14}$$

Then the authors of [3] take $\beta = 1$ fixed and find a local computation of $\log(Z)$. Remember that in Boltzmann distributions, $\log(Z) = (S_\mu - \beta U_\mu)$ 3.7. They define $\mathbb{F}$ as the local computation of $\log(Z)$, so $\mathbb{F} \equiv (S_\mu - \beta U_\mu)$. For $\mathbb{F}$ they find the following expression.

**Theorem 6.3.2.** *Let $\{\mu_a, \mu_i\}$ denote local marginals. Let $\vec{\nu}^* = \{\nu^*_{u\to a}, \hat{\nu}^*_{a\to u}\}$*
*Let*

$$\mathbb{F}_a(\vec{\nu}) = \log[\sum_{\underline{t}_{\partial a}} \psi_a(\underline{t}_{\partial a}) \prod_{u\in\partial a} \nu_{u\to a}(t_u)], \qquad \mathbb{F}_u(\vec{\nu}) = \log[\sum_{t_u} \prod_{b\in\partial u} \hat{\nu}_{b\to u}(t_u)] \tag{6.15}$$

$$\mathbb{F}_{au}(\vec{\nu}) = \log[\sum_{t_u} \nu_{u\to a}(t_u)\hat{\nu}_{a\to u}(t_u)]$$

*and*

$$\mathbb{F}_*(\vec{\nu}) = \sum_{a\in F} \mathbb{F}_a(\vec{\nu}) + \sum_{u\in V} \mathbb{F}_u(\vec{\nu}) - \sum_{(a,u)|u\in\partial a} \mathbb{F}_{au}(\vec{\nu}). \tag{6.16}$$

*Then on trees we have*

$$\log(Z) = \mathbb{F}[\mu] = \mathbb{F}_*(\vec{\nu}^*). \tag{6.17}$$

Having expressed $\log(Z)$ in terms of local quantities, it can be computed iteratively. This can then be used to calculate the free energy $F = -\frac{1}{\beta}\log(Z)$, which is minimized for fixed $\beta$. Using this local calculation of $F$ on graphs which are not trees as the estimate of $F$, leads to different values of $F$ for different fixed points. Thus the quality of different fixed points of the message equations (6.7) and (6.6) on graphs can be compared by the result of their locally computed free energies. This is called the cavity method, which we explain in the next section.

## 6.4 The cavity method

We have seen that on trees, many properties can be calculated locally on factor graphs. We shall now use these local calculations as the estimates of these properties on locally treelike graphs.

Even on general graphs a very nice link between the free entropy $\mathbb{F}_*(\vec{\nu})$ and bp-algorithms exists. The authors of [3] show that

**Lemma 6.4.1.** *The stationary points of the free energy $\mathbb{F}_*(\vec{\nu})$ are fixed points of belief propagation. Conversely, any fixed point $\vec{\nu}$ s.t. $\mathbb{F}_*(\vec{\nu})$ is finite is also a stationairy point of $\mathbb{F}_*(\vec{\nu})$.*

This is great, as it means that fixed points correspond to local extreme values of the free energy $-\frac{1}{\beta}\mathbb{F}$. Thus the global minimum will be a fixed point, so only the free energies of the fixed points themselves need to be considered.

Now the Replica symmetric cavity method computes the partition function iteratively. The cavity method is a method where an iterative method is used where every step a node is added, to in this case compute the partition sum $Z$,

$$Z_{v\to a}(t_v) = \prod_{b\in\partial v\backslash a}[\sum_{\underline{t}_{\partial b\backslash v}} \psi_b(\underline{t}_{\partial b}) \prod_{k\in\partial b\backslash v} Z_{k\to b}(t_k)]. \tag{6.18}$$

Using this locally computed partition function we can calculate the local free energy. As the free energy is minimized at constant $\beta$, the fixed point with the lowest free energy is the one with higher likelihood, therefore the correct fixed point.

## 6.5 The cavity method in Decelle's algorithm

We now take a look at what free energies the cavity approach yields for the messages on a regular graph in BPAD, (5.25). They rewrite the equations (6.16) in order to define a locally computed free energy density $f_{\text{bp}}$, as defined in (3.8).

$$Z^{uv} = \sum_{i<j} c_{ij}(\psi_i^{u\to v}\psi_j^{v\to u} + \psi_j^{u\to v}\psi_i^{v\to u}) + \sum_i c_{ii}(\psi_i^{u\to v}\psi_i^{v\to u}) \text{ for } (u,v)\in E \tag{6.19}$$

$$\hat{Z}^{uv} = \sum_{i,j}(1-\frac{c_{ij}}{n})\psi_i^u\psi_j^v \text{ for } (u,v)\notin E \tag{6.20}$$

$$Z^u = \sum_{t_u} n_{t_u} e^{-h_{t_u}} \prod_{v\in\partial u}\sum_{t_v} c_{t_v t_u}\psi_{t_v}^{v\to u} \tag{6.21}$$

$$f_{\text{bp}}(q,\{n_i\},c) = -\frac{1}{n}\sum_u \log Z^u + \frac{1}{n}\sum_{(u,v)\in E}\log Z^{uv} - \frac{d}{2} \tag{6.22}$$

with $\sum_{u,v}\log(\hat{Z}^{uv}) = d/2 + \mathcal{O}(1/n)$

Decelle et al. [1] look at an $IRG_n$ which satisfies that the expected outgoing degree is equal for every type,

$$d = \sum_{k=1}^q c_{ik}n_k = \sum_{k=1}^q c_{jk}n_k \forall i,j\in[q]. \tag{6.23}$$

If this is the case then we cannot base the guess of types based on the degrees of a node, which makes the community detection problem less trivial. In this case the trivial fixed point

$$\psi_{t_u}^{u\to v} = n_{t_u}, \tag{6.24}$$

leads to the free energy density using (6.22) of

$$Z^{uv} = \sum_{i<j} c_{ij}(2n_in_j) + \sum_i c_{ii}n_i^2 = \sum_{i<j} 2n_i(c_{ij}n_j) + \sum_i n_i(c_{ii}n_i). \tag{6.25}$$

Using $f_{\text{bp}}$ (6.22) as the local computation this simplifies to

$$f_{\text{trivial}} = \frac{c}{2}(1-\log(c)). \tag{6.26}$$

Therefore fixed points can now be compared using these free energies. It turns out that on anti-ferromagnetic models this gives rise to a whole new regime, where community detection is possible, but takes exponential time. We discuss this in section 7.2.

## 6.6 Results on treelike graphs

We discuss here the density evolution equations and some results on treelike graphs which make a case for the approximations done by in Decelles algorithm. However, we then also show that these results do not apply to BPAD.

The density evolution equations are the equations which describe $\nu^t_{u\to a}$ (6.7) and $\hat{\nu}^t_{a\to u}$ (6.6). Usually this is done for more general models than the Boltzmann distribution (6.1), but we stick with those models here. As the updates of the messages are only dependent on incoming messages, it is natural to introduce the directed neighbourhood of height $h$.

**Definition 6.6.1** (directed neighbourhood). *Let $G$ be a factor graph, let $E = (\partial a)_{a\in F}$ be the edge set of this factor graph. Then the directed neighbourhood of a directed edge $u \to a$ of height $h$ is denoted $B_{u\to a,h}(G)$ and is defined as the subtree including all factor nodes and variable nodes reachable from $u$ in $G' = G \setminus (u, a)$ by using at most $2h$ edges. As the graph is bipartite this shall result in a subtree with as root $u$, $h$ layers of factor nodes and the $h$ other layers of variable nodes,*

An example can be seen in figure 6.2.

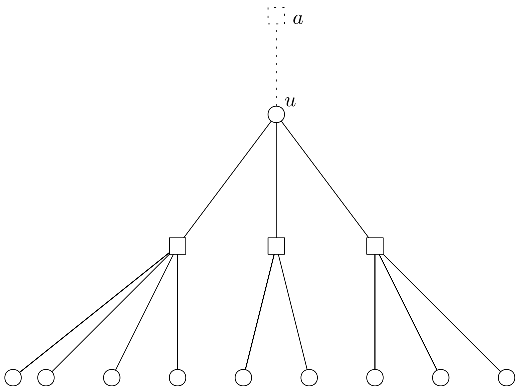


Figure 6.2: The directed neighbourhood $B_{u\to a,1}$. We have drawn the edge $(u, a)$ dotted as it itself is not part of the neighbourhood.

Using this directed neighbourhood we can see that for graphs which are locally treelike, as $n$ grows large, there is an $R$ s.t. $B_{u\to a,R}$ is a tree for any $u \in V$, $a \in F$. For iterations $t < R$ we therefore have that the messages $\nu^t_{u\to a}$, $\hat{\nu}^t_{a\to u}$ depend only on updates in trees. Thus if a belief propagation algorithm converges with $t < R$ then its results are exact. When $n \to \infty$ then $R \to \infty$, so for the local limit these results hold.

Unfortunately for us, BPAD functions only on finite graphs. The convergence of BPAD to a fixed point that is not trivial requires the information from loops, as was shown in section 5.4. Thus for our purposes the tree approximation remains an approximation.

Intuitively this can be explained as the fact that $\log(Z)$ should be constant for a given graph, but the locally computed free energy is dependent on the specific messages. However, in this chapter we have seen that the global minimum of the locally computed free energy corresponds to a fixed point of belief propagation messages. As the free energy is minimized for constant $\beta$, this global minimum should approximate the correct free energy, cycles are large and therefore the incoming messages can be approximated as coming from independent trees.

# Chapter 7

# Summary and outlook

In this chapter we first summarize our conclusions and then go into some questions that can be researched further.

## 7.1 Summary

Here we discuss our results and their significance.

We have seen in chapter 4 that on trees, Potts models with a magnetic field $\vec{B} = \mathbf{0}$ on trees can be described as a branching process, since the probability of a state is only dependent on the type probabilities of the root and the amount of flips. These descriptions are equivalent for $\exp(-2\beta) = \frac{(q-1)\epsilon}{1-(q-1)\epsilon}$. We have also seen that the Potts model can be adjusted to have a type-dependent interaction strength $J_{t_u t_v}$ to match a larger class of broadcasting processes. Using our proof method, it was not possible to describe a Potts model with nonzero magnetic field to a broadcasting process, because then the probability of a state in the Potts model becomes dependent on the size of the tree after a flip. We predict that it is impossible to couple these models, without letting the root probabilities of the broadcasting process depend on the branching factor tree. Our intuition for this is based on the fact that very branching trees have much smaller sub-trees after flips than less branching trees. Thus the root probability of the Potts model will depend on the size of the tree, the branching factor of the tree and the magnetic field. As in the broadcasting process the root probabilities need to be defined by hand it will therefore need to be dependent on the branching factor.

The other thing we have seen in chapter 4 is that a broadcasting process on the local limit of an $ER_n(\lambda/n)$ can be described as the local limit of a marked $\mathrm{IRG}_n(c)$, as long as the expected offspring of each of the types in the IRG is equal to $\lambda$. In this case we find equivalence when $\mathbb{P}(\eta_{(u,v)} = A_{ij} = \frac{c_{ij}n_j}{\lambda}$. This is the largest equivalence we can get using this method, as the ER graph simply has a offspring of $\lambda$ for all vertices.

We combine these results to identify the parameters of a marked SBM with that of the Potts model on ER in the local limit. We find that when $\lambda = \frac{a+(q-1)b}{q}$ and $\beta = -\frac{1}{2}\log(\frac{(q-1)b}{a})$, these models coincide. We fail to show that the critical parameters for community detection are equal using this identification, but we think this is due to a misinterpretation of the results of [7] on our part.

Our final result of chapter 4 was that all of these results can be extended to degree-corrected models. This is very nice, as it allows for much more freedom when trying to model a real problem realistically.

In chapter 5 we show how belief propagation algorithms were created from the property that they calculate exact marginals on trees. We showed that applying these algorithms to locally treelike graphs should approximate the marginals. Using this approximation of marginals and a Hamiltonian which maximizes the probability of being generated from the IRG, the algorithm of Decelle et al. detects communities on the IRG. Their order of message calculation is chosen randomly in order not to bias the algorithm and the initial values are chosen randomly in order to find as many fixed points as possible. We show that the critical condition of when this algorithm performs successful community detection in linear time on the SBM can be rewritten as $(a-b)^2 > q(a+(q-1)b)$. This creates a gap between the result of community detection being impossible when $(a-b)^2 < q(a+b)$ from the authors of [4]. In section 7.2 we show that this gap can partly be explained by a regime where community detection takes exponential time.

In chapter 6 we show that using the description of factor graphs it can be proven that even more physical quantities can be computed iteratively on trees. Again these local computations can be generalized to algorithms which approximate those quantities on locally treelike graphs. This way we can approximate the free energy of the system by the local calculation of the free energy of the fixed points. As the free energy is minimized for constant $\beta$ in Boltzmann distributions, which is used, this global minimum should approximate the correct free energy, Also we can use this approximated free energy to compare different fixed points and find the one which detects communities best. One thing that can still be researched and would help improve the understanding of these algorithm is to quantify how good this approximation is for cycles of length $r$ in locally treelike graphs.

One other way to go about identifying the parameters of a marked SBM with that of the Potts model on ER is to not use local limits at all, but instead prove equivalence on finite graphs We show our attempts at this in section 7.3.

## 7.2 local minima in antiferromagnetic regime

We have left the situation where $p_{\text{in}} < p_{\text{out}}$, the anti-ferromagnetic case, completely out of our scope. In the Potts models this is achieved by using negative $J$. Less research has been done in this regime, though many results from the ferromagnetic regime still apply.

The marginal calculation on trees is still exact. Thus, the same belief propagation can still be tried. Furthermore, the same Kesten-Stigum bound is found for detection using the non-rigorous analysis of section 5.4. Using $p_{\text{in}} = 0$ [1] they let $p_{\text{out}}$ grow, which corresponds with letting $\beta$ increase. They find very interesting behaviour of the cavity method for $\beta < \beta_c$. For $\beta\beta_{c_d}$, they find that the algorithm always converges to the fixed factorized point as expected.

But then for $\beta_{c_d} \leq \beta \leq \beta_{c_c}$ there is a phase where starting at precisely the original communities leads to convergence unequal to the factorized fixed point. The free energy of this state can be calculated locally using (6.22) and for $\beta \leq \beta_{c_c}$ we have $f_{bp} < f_{factorized}$, so the correct guess would be the factorized fixed point.

For $\beta_{c_c} \leq \beta \leq \beta_{c_l}$ we have the same behaviour with $f_{bp} > f_{factorized}$. Thus the correct guess is the one created with very specific initial conditions. Thus in this regime there is enough information to detect communities. Finding the right communities, however, would take exponential time as almost all initial conditions converge to the factorized fixed point.

For $\beta > \beta_{c_l}$, every initial condition converges to the same fixed point unequal to the factorized fixed points, which has positive overlap with the original communities.

The $\beta_{c_c} \leq \beta \leq \beta_{c_l}$ regime explains the discrepancy between the mathematical results for the SBM, which claim that no detection is possible for $(a-b)^2 < q(a+b)$, and the Kestem-Stigum conjecture which claims that detection is possible for $(a-b)^2 > q(a+(q-1)b)$. In between these regions there is, for anti-ferromagnetic systems, a regime where detection is possible but not easy using this algorithm. [1] conjecture that no polynomial-time algorithm can achieve positive overlap, based on the physics of glassy systems.

## 7.3 Finite random graphs

Since the Potts model on the local limit of an ER is equivalent to the local limit of the SMB, the question arises: "how local is the local limit?". Thus we investigate the equivalence on finite graphs.

It has been proven that the marked local limit of a two-type SBM with parameters $a, b$ is equal to the Ising model on the local limit of $\text{ER}_n(\lambda/n)$, $\text{GWT}(\text{Poi}(\frac{a+b}{2}))$, when $\beta = \frac{1}{2}log(\frac{a}{b})$ (4.7). It makes sense to ask the question if this equivalence can hold on finite random graphs too.

In this case we shall restrict ourselves to the case where the Ising model only assigns probabilities to states which have exactly $n/2$ type $+1$ vertices and $n/2$ type $-1$ vertices, also denoted as states with 0 magnetization.

Let $\mathbb{P}_{a,b}$ be the probability distribution of this $\text{SBM}_n(q, a, b)$ and let $\mathbb{P}_\lambda$ denote the graph distribution of $\text{ER}_n(\lambda/n)$

What we would like to find is parameters $\lambda, \beta$ in terms of $a, b$ such that

$$\mathbb{P}_{a,b}(G, \sigma) = \mathbb{P}_\lambda(G)\mu_\beta(G, \sigma). \tag{7.1}$$

Where $\mu_\beta$ is an Ising model as defined in equation (3.3). We did not find such a result. However, with a slightly adjusted Ising model such a result is possible.

We adjust the Ising model such that equation 7.1 holds. We define

$$\mu_{\beta,\gamma}(G, \sigma) = \frac{1}{Z_n(\beta, \gamma)} \exp\left(\beta \sum_{(i,j)\in E} \sigma_i\sigma_j + \gamma \sum_{i\neq j \in V} \sigma_i\sigma_j\right) \tag{7.2}$$

This $\gamma$ can be seen as the parameter controlling long-ranged interactions caused by the finite size of the graph. Now we wish to average out over the randomness of the graph. Thus we take the average over the randomness both in the numerator as well as the denominator, giving rise to the annealed measure

$$\mu^a_{\beta,\gamma}(G,\sigma) = \frac{1}{\mathbb{E}[Z_n(\beta,\gamma)]}\mathbb{E}[\exp\left(\beta\sum_{(i,j)\in E}\sigma_i\sigma_j + \gamma\sum_{i\neq j\in V}\sigma_i\sigma_j\right)] \tag{7.3}$$

Then the authors of [5] obtain the following result:

**Theorem 7.3.1.** *Take a* $\mathrm{SBM}_n(2,a,b)$ *with parameters* $a$ *and* $b$ *with* $n$ *even and precisely* $n/2$ *nodes with type* $+1$ *and* $n/2$ *types* $-1$. *Let* $\mu^a_{\beta,\gamma}(G,\sigma)$ *be an annealed Ising model conditioned on having* $n/2$ *types of* $+1$. *Then the SBM has the same law as* $\mu^a_{\beta,\gamma}(G,\sigma)$ *on an Erdős–Rényi graph* $G$ *with parameters*

$$\lambda = \lambda_n(a,b) = \frac{\sqrt{\frac{ab}{(1-a/n)(1-b/n)}}}{1+\sqrt{\frac{ab}{(1-a/n)(1-b/n)}}/n} \tag{7.4}$$

$$\gamma = \gamma_n(a,b) = \frac{1}{2}\log(\frac{1-a/n}{1-b/n}) \tag{7.5}$$

*and*

$$\beta = \beta_n(a,b) = \frac{1}{2}\log(a/b) - \gamma_n(a,b) \tag{7.6}$$

Note that this result can be extended to q groups.

Now we take the limit $n \to \infty$ of in order to compare the limit behaviour to our equivalence of chapter 4. We see that when $n \to \infty$, we get $\gamma \to 0$, $\beta \to \frac{1}{2}\log(a/b)$ and $\lambda \to \sqrt{ab}$. Then these results do not correspond to our results for the regular Ising model, where we get the same equivalence for $\lambda = \frac{a+b}{2}$.

Thus further research could look if this result on finite graphs can be extended to a larger set of Ising models, which are not annealed or restricted to having $n/2$ nodes having type $+1$.

# Bibliography

[1] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):66106, 2011. 2, 5, 15, 35, 41, 42, 45, 47, 48, 52, 55

[2] William Evans, Claire Kenyon, Yuval Peres, and Leonard J Schulman. Broadcasting on trees and the Ising model. *Annals of Applied Probability*, pages 410–433, 2000. 2, 15, 16, 25, 27

[3] Marc Mezard and Andrea Montanari. *Information, physics, and computation.* Oxford University Press, 2009. 2, 48, 50, 51, 52

[4] Lennart Gulikers, Marc Lelarge, and Laurent Massoulie. An impossibility result for reconstruction in the degree-corrected stochastic block model. *Ann. Appl. Probab.*, 28(5):3002–3027, 2018. 4, 13, 34, 47, 54

[5] Remco van Der Hofstad. Random graphs and complex networks. Volume 2.(2020+). *Cambridge Series in Statistical and Probabilistic Mathematics., available at https://www. win. tue. nl/˜ rhofstad/NotesRGCNII. pdf.* 9, 11, 12, 14, 56

[6] David Aldous and J Michael Steele. The objective method: probabilistic combinatorial optimization and local weak convergence. In *Probability on discrete structures*, pages 1–72. Springer, 2004. 10

[7] Robin Pemantle and Jeffrey E Steif. Robust phase transitions for Heisenberg and other models on general trees. *Annals of Probability*, pages 876–912, 1999. 20, 22, 27, 30, 31, 54

[8] Amir Dembo and Andrea Montanari. Ising models on locally tree-like graphs. *The Annals of Applied Probability*, 20(2):565–592, 2010. 21, 37

[9] Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3-4):431–461, 2015. 27, 30

[10] Mateusz Wilinski, Piero Mazzarisi, Daniele Tantari, and Fabrizio Lillo. Detectability of macroscopic structures in directed asymmetric stochastic block model. *Physical Review E*, 99(4):42310, 2019. 42

[11] Harry Kesten and Bernt P Stigum. A limit theorem for multidimensional Galton-Watson processes. *The Annals of Mathematical Statistics*, 37(5):1211–1223, 1966. 46

# Appendix A

# Appendix

Here we show all the abbreviations that are used throughout this thesis.

| Abbreviation | Full notation | Reference |
|---|---|---|
| ER | Erdős–Rényi random graph | 2.2 |
| DER | Degree-corrected Erdős–Rényi random graph | 2.2 |
| IRG | Inhomogeneous random grapth | 2.2.2 |
| SBM | Stochastic block model | 2.2 |
| DSBM | Degree-corrected stochastic block model | 2.2 |
| GWT | Galton Watson Tree | 2.3.5 |
| BPAD | Decelle et al.'s belief propagation algorithm | 2 |

Here we show many of the variables that are used throughout this thesis such that they can easily be looked up.

| Variable | Definition (in this thesis) |
|---|---|
| $G$ | A (possibly random) graph $G = (V(G), E(G))$ with vertex set $V(G)$ and edge set $E(G)$ |
| $V(G)$ | A set containing all the vertices of $G$, with $\|V(G)\| = n$ |
| $u, v$ | vertices of $G$, so $u, v \in V(G)$ |
| $\sim$ | used to indicate that two vertices are adjacent (in $G$), for example $u \sim v$ |
| $\partial u$ | used to indicate all neighbours of $u$, in both a normal and factor graph |
| $E(G)$ | A set containing all edges of $G$, $E(G) = \{e = \{u, v\} \| u \sim v\}$, with $\|E(G)\| = m$ |
| $n$ | $n = \|V(G)\|$ the number of vertices of $G$ |
| $m$ | $m = \|E(G)\|$ the amount of edges of $G$ |
| $q$ | The number of different types |
| $\mathcal{S}$ | the space of types $[q] = \{1, ..., q\}$ |
| $i, j$ | $i, j \in [q]$ are specific types |
| $\sigma_u, \sigma_v$ | the types of vertices $u, v$ respectively in an Ising model or the two-type SBM |
| $t_u, t_v$ | the types of vertices $u, v$ respectively |
| $\Omega$ | The set of all possible ways to assign types to all vertices in $G$, $\Omega = \mathcal{S}^n$ |
| $\omega$ | a single result $\omega \in \Omega$, usually $\omega = \vec{t}$ |
| $\mu$ | a probability measure 2.1.1 that assigns a probability to every result $\omega \in \Omega$ |
| $X, Y$ | used as random variables. The distribution is defined in the section itself |
| $\mathbb{P}(A)$ | The probability of a set $A$ |
| $\mathbb{E}(X)$ | The expectation of random variable $X$ 2.1.1 |
| $\nu$ | a continuous random variable used for the weight distribution in degree-corrected cases |
| $\bar{v}$ | the branching factor (3.10) |
| $Z$ | The partition function which normalizes Boltzmann measures |
| $H$ | the Hamiltonian which defines the energy of a state in a Boltzmann measure |
| $U$ | The internal energy (3.5) |
| $S$ | The entropy (3.6) |
| $F$ | The free energy (3.7) |
| $\mathbb{P}_\eta$ | a probability measure called a broadcasting process (2.46) |
| $\{\eta_e\}$ | a set of discrete random variables defined on every edge of a tree $T$ such that (2.45) is satisfied |
| $A$ | The $q \times q$ flipping matrix coupled with a general broadcasting process via (4.12) |
| $a$ | a factor node in a factor graph (defined in 6.1.1) |