Learning bipartite Ising models

Paula Anette Mürmann

4th Year Project Report
Computer Science and Physics
School of Informatics
University of Edinburgh

2022

Abstract

An Ising model is a graph structure with random variables assigned to each vertex. The correlations of these random variables are described by the edges in the graph. A learning task on an Ising model is then the challenge of reconstructing the edges of the underlying graph using samples collected from the random variables.

The main aim of this project was to understand and implement an algorithm proposed by Surbhi Goel (2019) which recovers sets of vertices which are connected via one intermediate vertex. To generate the required samples we explored the use of a rapidly mixing Markov chain algorithm, which allows us to sample larger Ising models than the experiments presented in Goel's paper.

We reconstructed and expanded upon these experiments and were able to show the logarithmic relationship between sample size and accuracy of the reconstruction predicted by Goel. As a major practical limitation we observed that to achieve reliable structure recovery either the number of samples has to be very large or a parameter of the algorithm has to be tuned with some further knowledge of the underlying structure.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Paula Anette Mürmann*)

Acknowledgements

I would like to thank my supervisor Mary Cryan for her guidance, insights and feedback. I enjoyed the work on this project very much and am very grateful for our interesting discussions.

I would also like to thank my family and friends for their support and interest in my project.

Table of Contents

Chapter 1

Introduction

1.1 Overview

This project will address the structure learning problem on a graphical model. Structure learning is the task of deducing the graph network from a set of measurements of some graph property. In this project we are looking at Ising models, which are graphical networks with binary random variables associated with their vertices. How these random variables are correlated depends on the structure and parameters of the Ising model. The learning problem is then the inverse challenge of recovering the underlying graph from measurements of correlation between these random variables.

The project is based on the paper ”Learning Restricted Boltzmann Machines with Arbitrary External Fields” by Goel [7], which proposes and proves the correctness of an algorithm to perform this learning task. Restricted Boltzmann machines are a type of bipartite Ising model, in which some vertices are hidden from observation.

In this report we will first outline the characteristics of an Ising model and how to generate samples from it. Next, we will discuss and explain the algorithm proposed by Goel. Building on these theoretical insights we will show an implementation of the Ising model, sampler and the algorithm and finally present experiments exploring the applicability of Goel’s algorithm.

1.2 Motivation

Ising models were first proposed in a statistical physics context as a model for magnetic phase transitions. Especially of interest were the pairwise influences between vertices that can simulate a dramatic change in the macroscopic properties (e.g. the bulk magnetisation of the material) once a certain strength of interactions is reached.

However, the idea of modelling a network of pairwise interactions and alignment effects between a number of nodes has also made the model popular in many other domains. It can be applied in economic contexts [27], medical research [10], machine learning [9] and many more.

In particular, bipartite Ising models with hidden nodes have been studied in supervised or unsupervised learning settings. For these applications the probability distribution of the data is first encoded into the model, then the model can be used to generate samples from this distribution (using a sampling algorithm)[6].
The aim of this project is to understand an algorithm for structure learning of bipartite Ising models and experimentally explore how Ising models can be simulated and sampled.

1.3 Related Work

As stated above, bipartite Ising models with hidden nodes are popular models considered in Machine Learning. However, the learning methods employed usually involve gradient descent strategies, optimising the likelihood of certain graph parameters given the data [14].
In contrast, this project considers an algorithm that recovers the structure of a graph exactly (with some probability) by considering the correlations between the random variables associated with vertices in the graph.
Goel's algorithm is largely based on a more general approach to ferromagnetic Ising models described in the paper "Efficiently learning Ising models on arbitrary graphs" by Bresler [1]. Bresler uses a measure called the *conditional influence* between random variables (a measure of how much the distribution of one random variable varies if the other one is flipped) to gauge how connected two vertices are. Goel replaces this quantity with the conditional covariance between random variables. Both algorithms work in a similar way by considering a certain node u and constructing a proposed set of neighbours (or in Goel's case the set of nodes that share a common neighbour) which consists of the nodes with the highest influence or covariance with node u. This is done until the threshold (or lower bound) on the influence/covariance is reached. Then, they go back through the proposed neighbourhood set and exclude all nodes that turn out to have a smaller influence/covariance conditioned on the rest of the proposed neighbourhood.
Bresler et al. also proposed a similar algorithm for ferromagnetic restricted Boltzmann machines (where neighbouring vertices are only positively correlated) with non-negative external fields, which again relies on an influence measure between the random variables [2]. The structure resembles the algorithm described above. However, it does not require a pruning step (i.e. the second loop to remove vertices from the proposed neighbourhood set). Goel's algorithm in contrast allowed for the restricted Boltzmann machines to be more general by including arbitrary external fields and some anti-ferromagnetic edges (i.e. neighbouring vertices are negatively correlated).
Other works extend Bresler's results to Markov random fields and improve upon the sample complexity of the algorithm by generalising Bresler's threshold criterion [13] or proposing a different approach with an online learning algorithm [17].
These works are all bounded by a result due to Santhanam and Wainwright [25] proving an information-theoretic lower bound on the number of samples required for structure recovery in Markov random fields with a certain probability.

1.4 Contribution

We summarise our main contributions to this project as this:

1. Theoretical contributions to support Goel's presentation:
   - Proof of Goel's remark on the representability of locally consistent bipartite Ising models as purely ferromagnetic models (Proposition 1)
   - Derivation of the probability of a vertex spin being $+1$ conditioned on the spin values of its neighbours (Proposition 2)
   - Proof of Goel's remark that the conditioning set S can be assumed to be empty for the proof of Goel's Lemma 1 (Proposition 3)
   - Justification of steps in Goel's proof of Theorem 1
2. Implementation and testing of an Ising model and Swendsen-Wang sampler (described in section 4.1).
   Implemented as an IsingModel class (approx. 700 lines of Python code incl. auxiliary functions) with functionality:
   - Generation of Ising models with given parameters
   - Visualisation of model
   - Loading previously generated models from file
   - Experiments with Gibbs sampler
   - Swendsen-Wang sampler
3. Implementation of the learning algorithm (approx. 15 lines of code) and calculation of the empirical conditional covariance (approx. 50 lines, see section 4.2)
4. Reconstruction of Goel's experiments (Section 5.2) and extension to larger graphs as well as denser graphs (Section 5.3), observing the logarithmic relationship between the number of samples required and the number of observed vertices (approx. 300 lines of code to set up, automate and analyse experiments)
5. Exploration and interpretation of the τ threshold used in the algorithm (Section 5.1.1)

Chapter 2

The Ising Model

In this chapter we will present the probabilistic graphical model we are working with. We will define the characteristics of the model and discuss how samples can be generated from it.

2.1 The Model

2.1.1 Idea

An Ising model is a graph $G = (V, E)$ consisting of a set of vertices (or nodes) V and a set of weighted edges E. Each vertex u is associated with a random variable X_u which can take the value -1 or $+1$ (this is often also called the spin value of the vertex). See figure 2.1 for an example. The probability distribution over X_u depends upon the distributions of each neighbouring vertex of u and the weight of the adjacent edge. So for example, X_u is more likely to be positive if its neighbours are likely to be positive and the edges between u and these neighbours have large positive weights. If the weight on any edge is negative, u is more likely to take the opposite value to this neighbour. If there is no edge between two nodes u and v, then the probability of the outcome of u will not directly depend on the value of v. However, the random variables X_u and X_v may still be correlated for example via common neighbours.

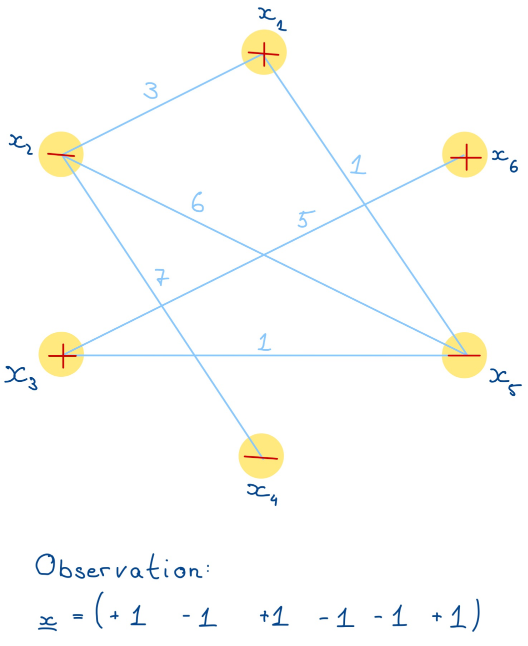


Figure 2.1: Example Ising Model with one observation of values from its distribution

In a general Ising model, each vertex is also associated with an additional weight (referred to as the *external field*, a term borrowed from the foundations of the Ising model in statistical physics). These weights can again be positive or negative and shift the node's probability distribution towards the $+1$ or -1 outcome.

A sample drawn from an Ising model is one observed outcome of the random variable over all nodes in the graph. So for example, a sample drawn from a graph with 6 vertices $V = \{v_1, \ldots, v_6\}$ could be $[(x_1 = +1), (x_2 = -1), (x_3 = +1), (x_4 = -1), (x_5 = -1), (x_6 = +1)]$ (see figure 2.1).

Learning an Ising model refers to the structure learning process of reconstructing the underlying graph from a set of samples. Specifically, the vertices (or at least some of the vertices) in the graph and a fixed number of samples are given and the challenge is to determine where the edges between these vertices must be.

This project considers a special type of model where the underlying graph is bipartite and samples are only observed from one of the vertex sets. A bipartite graph is a structure where the vertex set V of the graph can be partitioned into two disjoint subsets V_{obs} and V_{lat} such that each edge connects vertices from opposite sets. A further constraint of our learning problem is that only vertices from V_{obs} can be observed while the other set, V_{lat}, remains hidden (or sometimes called latent).
On the one hand, restricting the underlying graph to a bipartite structure makes reasoning about the graph easier because additional properties about the structure are known (for example, the neighbourhood of a node can only contain vertices from the other set).
On the other hand, the task is made harder by observing only one set of vertices because less information is known about the probability distribution.

2.1.2 Definition

Now, some notation [1] will be introduced and explained.

Definition 1 (Ising Model)

- V_{obs} & V_{lat}: set of observed/latent nodes
- $n = |V_{obs}|$ & $m = |V_{lat}|$: size of the vertex sets (indexed from 1 to n, m for convenience)
- X & Y: random variables associated with the outcome of the entire observed vertex set or the entire latent set
- X_u, Y_u or X_S, Y_S: random variables for a specific vertex u or a set of vertices S
- $\underline{x}_S \in \{\pm 1\}^{|S|}$: a particular outcome (i.e. an assignment of ±1 values) for a set of vertices S
- $\underline{h} \in \mathbb{R}^n$: external field vector for the observed vertices where h_i is the external field on vertex i
- $\underline{g} \in \mathbb{R}^m$: external field vector for the latent vertices

[1] Adapted from Goel [7] with additional useful symbols

- $J \in \mathbb{R}^{n \times m}$: matrix of edge weights (or interaction matrix) where J_{ij} gives the weight of the edge between the i^{th} observed and the j^{th} latent node. If there is no edge between the vertices, then $J_{ij} = 0$

We will use the notation $[z] = \{\ 1,\ 2, \ldots,\ z\}$ throughout.

This type of bipartite Ising models is often also called *restricted Boltzmann machine* (RBM) because the probability of a certain outcome $(X = \underline{x}, Y = \underline{y})$ (i.e. the assignment of ±1 values to each vertex) is given by a simplified [2] Boltzmann distribution:

$$p(X = \underline{x}, Y = \underline{y}) = \frac{1}{Z} \exp\left(\underline{x}^\top J \underline{y} + \underline{h}^\top \underline{x} + \underline{g}^\top \underline{y}\right) \tag{2.1}$$

where Z is the so-called partition function, a sum over all possible value assignments to all vertices in the graph:

$$Z = \sum_{\substack{\underline{x} \in \{\pm 1\}^n \\ \underline{y} \in \{\pm 1\}^m}} \exp\left(\underline{x}^\top J \underline{y} + \underline{h}^\top \underline{x} + \underline{g}^\top \underline{y}\right) \tag{2.2}$$

Z serves as the normalisation constant for the probability distribution, where $\{\pm 1\}^n$ is the set of all possible $\underline{x}$ vectors and likewise $\{\pm 1\}^m$ for $\underline{y}$. This already foreshadows a problem for simulations since the number of terms in the sum grow exponentially with the number of vertices in the model. This will be discussed further in section 2.2.
Note furthermore that all properties of the model are encoded in this probability distribution. Therefore, if two Ising models have equivalent distributions then their underlying models must be equivalent with respect to the learning algorithm. This will be very useful when motivating simplifying assumptions about the model.

2.1.3 Ferro- and Anti-Ferromagnetism

An Ising model is said to be ferromagnetic if all edges have positive weight. This means that edges between vertices only encourage alignment of the spin values. Anti-ferromagnetic models, on the other hand, only have negative edge weights so they encourage anti-alignment.
In the literature, many Ising models are restricted to be ferromagnetic because anti-ferromagnetic models can have positive and negative correlations between vertices which leads to *cancellations of correlations* [3]. This means that neighbours are much harder to detect by observing correlations.
Goel uses a model which is called *locally consistent*:

Definition 2 (Goel [7]) A bipartite Ising model with interaction matrix J is *locally consistent* if for all hidden nodes indexed $j \in [m]$:

$$J_{ij} \begin{cases} \geq 0 & \text{for all edges incident on node j (ferromagnetic w.r.t. node j)} \\ \leq 0 & \text{for all edges incident on node j (anti-ferromagnetic w.r.t. node j)} \end{cases}$$

[2] The full Boltzmann distribution $p(X = \underline{x}, Y = \underline{y}) = \frac{1}{Z} e^{-\frac{E}{kT}}$ with energy $E = -(\underline{x}^\top J \underline{y} + \underline{h}^\top \underline{x} + \underline{g}^\top \underline{y})$ [15] would include a temperature variable modifying the strength of all interactions. The Goel variant can therefore be seen as a fixed temperature simplification

Therefore, Goel's model can have ferro- or anti-ferromagnetic edges, but each latent vertex is restricted to have only positive or only negative outgoing edges.
With this restriction and the real-valued external fields Goel can essentially show that this is equivalent to a purely ferromagnetic graph as we formalise in Proposition 1:

Proposition 1 *Consider a locally consistent bipartite Ising model with a set of latent anti-ferromagnetic nodes K (i.e. the nodes which have only negative adjacent edges), external fields $\underline{h}$ (observed) and $\underline{g}$ (latent) and the probability distribution $p(X = \underline{x}, Y = \underline{y})$. Then there exists a purely ferromagnetic Ising model with distribution $\hat{p}(\, X = \underline{x}, Y = \underline{y})$ and external fields $\underline{h}$ and $\underline{g}'$ where $\underline{g}'_k = -\underline{g}_k \; \forall \; k \in K$ such that:*

$$p(\, X = \underline{x}, Y = \underline{y}) = \hat{p}(\, X = \underline{x}, Y = \underline{y}')$$

$$\text{where } y'_k = \begin{cases} y_k & \text{for } k \notin K \\ -y_k & \text{for } k \in K \end{cases}$$

Proof. Consider an anti-ferromagnetic latent node $k \in K$ such that its adjacent edges have negative weight. We analyse the probability distribution in terms of the marginal on Y_k:

$$p(X = \underline{x}, Y = \underline{y}) = \frac{1}{Z} \exp\left(\underline{x}^\top J \underline{y} + \underline{h}^\top \underline{x} + \underline{g}^\top \underline{y}\right) \tag{2.3}$$

$$= \frac{1}{Z} \exp\left(\sum_{\substack{i \in [n] \\ j \in [m]/k}} x_i J_{ij} y_j - \sum_{i \in [n]} x_i |J_{ik}| y_k + \underline{h}^\top \underline{x} + \sum_{j \neq k} g_j y_j + g_k y_k \right) \tag{2.4}$$

Now, consider mapping all assignments of values to the latent node k from $Y_k \rightarrow -Y_k$. This does not change the overall probability distribution because all possible combinations of $\pm Y_k$ are present in the distribution. This also means that the partition function Z does not change under this map leading to:

$$p(X = \underline{x}, Y = \underline{y}')$$

$$= \frac{1}{Z} \exp\left(\sum_{\substack{i \in [n] \\ j \in [m]/k}} x_i J_{ij} y_j - \sum_{i \in [n]} x_i |J_{ik}| (-y_k) + \underline{h}^\top \underline{x} + \sum_{j \neq k} g_j y_j + g_k (-y_k) \right) \tag{2.5}$$

$$= \frac{1}{Z} \exp\left(\sum_{\substack{i \in [n] \\ j \in [m]/k}} x_i J_{ij} y_j + \sum_{i \in [n]} x_i |J_{ik}| y_k + \underline{h}^\top \underline{x} + \sum_{j \neq k} g_j y_j + g'_k y_k \right) \tag{2.6}$$

$$= \hat{p}(X = \underline{x}, Y = \underline{y}) \tag{2.7}$$

Notice that this is the distribution of an Ising model that is ferromagnetic in vertex k only with the external field g_k turned negative. As Goel's definition allows external fields $\in \mathbb{R}$ this means that it is also a valid Ising model under Goel's definition.
Then, mapping $Y_k \to -Y_k$ back:

$$p(\,X = \underline{x}, Y = \underline{y}) = \hat{p}(\,X = \underline{x}, Y = \underline{y}') \tag{2.8}$$

Note furthermore, that the described transformation only changes the edges adjacent to hidden vertex k. Due to the bipartite structure of the graph no edge from k can be connected to another anti-ferromagnetic node. Therefore, this argument can be applied to all anti-ferromagnetic hidden nodes successively, giving a fully ferromagnetic Ising model. □

By proposition 1 no loss of generality occurs when considering ferromagnetic Ising models with arbitrary external fields instead of locally consistent models. However, to preserve full generality we have to allow for positive as well as negative external fields. For the rest of the project we therefore assume our Ising model to be ferromagnetic with arbitrary external fields.

2.2 Generating Samples via Markov Chain Monte Carlo Methods

2.2.1 Motivation

There are two general approaches to creating samples from an Ising model. Either the samples can be generated directly from the distribution or some algorithm approximating the distribution can be applied.
The model can be sampled directly from the probability distribution (equation 2.1) by indexing all 2^{n+m} outcomes $(X = \underline{x}, Y = \underline{y})$ and using a sampling algorithm that generates results according to the distribution (e.g. by rejection sampling [26] or more advanced methods [24]).
However, it becomes infeasible to compute the partition function (Eq. 2.2) exactly for models with $n+m \approx 15$ [8] which makes the probability distribution inaccessible. We explored whether the partition function can be factorised as this is a common method in statistical physics to handle large sums. This did not prove feasible though because the exponent contains terms with x_i elements, y_j elements or both which prevented us from factoring out common x_i or y_j factors.
Therefore, the distribution has to be approximated if larger Ising models are to be modelled. For this, a Markov chain combined with a Monte Carlo approach can be used. A Markov chain is a sequence of samples where a next sample a' is chosen based on the last sample a and a transition probability $p(a \to a')$ [18]. The chain can be initialised with a random starting state and should eventually reach a so-called *mixed state* in which the frequencies of the samples generated should closely approximate the theoretical distribution. Provided the transition probability is defined correctly a mixed state should be reached in some finite time called the mixing or burn-in time.
A Monte Carlo algorithm then describes the steps required to setup and calculate elements in such a Markov chain.

2.2.2 Gibbs Sampler

One of the most common algorithms for sampling an Ising model is a Gibbs sampler (see GIBBSSAMPLER pseudo code on page 11 adapted from [20]). This was the first experimental sampler tried in this project. It starts with a random configuration on all nodes and then randomly selects one vertex v to update. The updated value for v will be set to ±1 with a probability $p(X_v = \pm 1 | \, X_{N(v)})$. This probability describes the Boltzmann distribution 2.1 marginalised over all non-neighbours of v and conditioned on the state of the neighbouring nodes. It is computationally feasible because the conditioning on the neighbours allows all other vertex contributions to be factored out of the partition function and the numerator, which simplifies the expression. Here we give an expression for $p(X_v = \pm 1 | \, X_{N(v)})$ and its derivation. We denote the neighbourhood of a vertex v by $N(v)$.

Proposition 2 *In a bipartite Ising model, the probability for the spin X_v of some observed vertex v to be $+1$ given that its neighbouring spin values are $\underline{y}_{N(v)}$ is:*

$$p(X_v = +1 | \, Y_{N(v)} = \underline{y}_{N(v)}) = \frac{\exp\left(2\left(\sum_{j\in N(v)} J_{vj}y_j + h_v\right)\right)}{1+\exp\left(2\left(\sum_{j\in N(v)} J_{vj}y_j + h_v\right)\right)}$$

Proof. Let $\underline{x}'$ and $\underline{h}'$ denote assignments and external fields to all visible nodes except for node v and $\underline{y}'$ and $\underline{g}'$ assignments to vertices not in $N(v)$. Let $\underline{x}$ and $\underline{y}$ denote the assignment of spins to all vertices in V_{obs} and V_{lat} composed of x_v and $\underline{x}'$ or $\underline{y}_{N(v)}$ and $\underline{y}'$.

$$p(X_v = +1 | \, Y_{N(v)} = \underline{y}_{N(v)}) = \sum_{\substack{\underline{x}'\in\{\pm 1\}^{n-1} \\ \underline{y}'\in\ \{\pm 1\}^{m-|N(v)|}}} p(X_v = +1, X' = \underline{x}', Y' = \underline{y}' | \, Y_{N(v)} = \underline{y}_{N(v)}) \tag{2.9}$$

Account for edges from v to all its neighbours $N(v)$ with the sum (I) and all vertices in $V_{obs} \setminus \{v\}$ to any node in V_{lat} with the sum (II). Note that by definition of the neighbourhood, there cannot be any edges between v and $V_{lat} \setminus N(v)$.

$$= \sum_{\substack{\underline{x}'\in\{\pm 1\}^{n-1} \\ \underline{y}'\in\ \{\pm 1\}^{m-|N(v)|}}} \frac{\exp\left(\overbrace{\sum_{j\in N(v)} J_{vj}y_j}^{(I)} + h_v + \overbrace{\sum_{\substack{i\in[n]\setminus v \\ j\in[m]}} x_i' J_{ij} y_j}^{(II)} + \underline{h'}^\top \underline{x}' + \underline{g}^\top \underline{y}\right)}{Z \cdot p(Y_{N(v)} = \underline{y}_{N(v)})} \tag{2.10}$$

$$= \exp\left(\sum_{j\in N(v)} J_{vj}y_j + h_v\right) \sum_{\substack{\underline{x}'\in\{\pm 1\}^{n-1} \\ \underline{y}'\in\ \{\pm 1\}^{m-|N(v)|}}} \frac{\exp\left(\sum_{\substack{i\in[n]\setminus v \\ j\in[m]}} x_i' J_{ij} y_j + \underline{h'}^\top \underline{x}' + \underline{g}^\top \underline{y}\right)}{Z \cdot p(Y_{N(v)} = \underline{y}_{N(v)})} \tag{2.11}$$

Now rewrite $p(Y_{N(v)} = \underline{y}_{N(v)})$ as:

$$
\begin{aligned}
p(Y_{N(v)} = \underline{y}_{N(v)}) &= \frac{1}{Z} \sum_{\substack{\underline{x}\in\{\pm1\}^n \\ \underline{y}' \in \{\pm1\}^{m-|N(v)|}}} \exp\left(\sum_{\substack{i\in[n] \\ j\in[m]}} x_i J_{ij} y_j + \underline{h}^\top \underline{x} + \underline{g}^\top \underline{y} \right) \\
&= \frac{1}{Z} \sum_{x_v\in\{\pm1\}} \exp\left(\sum_{j\in N(v)} x_v J_{vj} y_j + h_v x_v \right) \sum_{\underline{x}',\underline{y}'} \exp\left(\sum_{\substack{i\in[n]\setminus v \\ j\in[m]}} x'_i J_{ij} y_j + \underline{h}'^\top \underline{x}' + \underline{g}^\top \underline{y} \right)
\end{aligned}
\tag{2.12}
$$

And observe that the Z and some parts of the partition function cancel:

$$
p(X_v = +1 |\, X_{N(v)}) =
$$

$$
\frac{\exp\left(\sum_{j\in N(v)} J_{vj} y_j + h_v\right) \sum_{\substack{\underline{x}'\in\{\pm1\}^{n-1} \\ \underline{y}' \in \{\pm1\}^{m-|N(v)|}}} \exp\left(\sum_{\substack{i\in[n]\setminus v \\ j\in[m]}} x'_i J_{ij} y_j + \underline{h'}^\top \underline{x}' + \underline{g}^\top \underline{y} \right)}{\frac{Z}{Z} \sum_{x_v\in\{\pm1\}} \exp\left(\sum_{j\in N(v)} x_v J_{vj} y_j + h_v x_v\right) \sum_{\underline{x}',\underline{y}'} \exp\left(\sum_{\substack{i\in[n]\setminus v \\ j\in[m]}} x'_i J_{ij} y_j + \underline{h}'^\top \underline{x}' + \underline{g}^\top \underline{y} \right)}
\tag{2.13}
$$

$$
= \frac{\exp\left(\sum_{j\in N(v)} J_{vj} y_j + h_v\right)}{\sum_{x_v\in\{\pm1\}} \exp\left(\sum_{j\in N(v)} x_v J_{ij} y_j + h_v x_v\right)} = \frac{\exp\left(2\left(\sum_{j\in N(v)} J_{vj} y_j + h_v\right)\right)}{1 + \exp\left(2\left(\sum_{j\in N(v)} J_{vj} y_j + h_v\right)\right)}
\tag{2.14}
$$

□

Observe that the probability $p(X_v = -1 |\, Y_{N(v)} = \underline{y}_{N(v)}) = 1 - p(X_v = +1 |\, Y_{N(v)} = \underline{y}_{N(v)})$ and probabilities for the latent vertices can be calculated in a similar way exchanging $x \leftrightarrow y$ and $h \leftrightarrow g$.

The simplified dynamics therefore circumvent the problem of calculating the full partition function. The single-vertex update is then repeated $|V|$ times which updates most vertices once. Then, the new configuration is stored as a sample in the output list and the process is repeated for T iterations. Note that out of the list of samples generated, the first $t < T_{\text{mixing}}$ samples have to be discarded because they retain the memory from the random initialisation. Furthermore, only every $(T_{step})^{th}$ sample (after T_{mixing}) should be taken to avoid correlation between samples.

2.2.3 Swendsen-Wang Sampler

The naïve Gibbs sampler does not in general exhibit *rapid mixing* (i.e. it does not reach a mixed state in polynomial time w.r.t. the input size) [16]. This has two main disadvantages.

Firstly, this means that the sampling algorithm would potentially have to run for a very long time to produce good results.

Algorithm 1: GibbsSampler(Graph $G = (V, E)$, Iterations T)
For a general (not necessarily bipartite) Ising model, where $\underline{f}$ is the general external field vector

```
$p = \frac{\exp(2(\sum_{i \in N(v)} J_{vi} X_i(t) + f_v))}{1 + \exp(2(\sum_{i \in N(v)} J_{vi} X_i(t) + f_v))}$
$X(t)$ : spin configuration (state of the Markov chain) at time $t$
$X_v(t)$ : spin associated with vertex $v$ at time $t$
$f_v$ : external field in $v$
$X(t = 0) \leftarrow$ random choice of an assignment from $\{\pm 1\}^{|V|}$
for $t = 1$ up to $T$ do
    $X(t) \leftarrow X(t-1)$
    for $j = 1$ up to $|V|$ do
        $v \leftarrow$ uniform random choice from $\{1, ..., |V|\}$
        set $X_v(t)$ to $\begin{cases} +1 & \text{with probability } p \\ -1 & \text{with probability } 1-p \end{cases}$
return $X$
```

Secondly and more importantly, it can only be estimated after how many steps this Markov chain will reach its mixed state because the true expected distribution is unknown. However, with a non-rapidly mixing Markov chain this guess is more likely to be wrong (i.e. much too low or much too high).
Therefore, it is better to use a theoretically rapid mixing sampler. One of these algorithms is the so-called Swendsen-Wang sampler, which achieves faster mixing by flipping whole clusters of spins at each step of the Markov chain.
The algorithm used in this project was adapted from [21] (see algorithm SwendsenWangSampler on page 12). We denote the Markov chain element at step t by $S(t)$ to distinguish it from the chain used in the Gibbs sampler.
The algorithm first constructs a set of monochrome edges M (see Fig. 2.2 (2)).

Definition 3 (Monochrome edge) An edge $(u, v) \in E$ is *monochrome* before iteration t if the two adjacent vertices have the same spin value i.e.

$$\text{for } (u,v) \in E\text{: } S_u(t-1) = S_v(t-1) \Rightarrow (u,v) \text{ is monochrome}$$

where $S_i(t)$ is the spin value associated with vertex i in step t of the Markov chain

We define $M = \{(u,v) \in E | S_u(t-1) = S_v(t-1)\}$. Then, edges are removed from set M with probability $p_1 = e^{-2J_{uv}}$ giving a set M' (Fig. 2.2 (3)). Note that this step is more likely to remove edges with small weights while keeping edges with higher weights. Afterwards, the graph $G' = (V, M')$ with the original vertex set and the reduced edge set is considered. The connected components in this residual graph are computed (Fig. 2.2 (4)) and randomly assigned the same spin value (Fig. 2.2 (5)), where the probability of choosing spin $+1$ is given by

$$p_2 = \frac{\exp(2(\sum_{v \in V(c) \cap V_{obs}} h_v + \sum_{v \in V(c) \cap V_{lat}} g_v))}{1 + \exp(2(\sum_{v \in V(c) \cap V_{obs}} h_v + \sum_{v \in V(c) \cap V_{lat}} g_v))} \tag{2.15}$$

which only depends on the external fields in the component.

Algorithm 2: SwendsenWangSampler(Ising Model I, Iterations T)

```
$V(c)$ : set of vertices in component c
$p_2 = \frac{\exp(2(\sum_{v \in V(c) \cap V_{obs}} h_v + \sum_{v \in V(c) \cap V_{lat}} g_v))}{1+\exp(2(\sum_{v \in V(c) \cap V_{obs}} h_v + \sum_{v \in V(c) \cap V_{lat}} g_v))}$
$S(t=0) \leftarrow$ random choice of an assignment from $\{\pm 1\}^{|V|}$
for $t = 1$ up to $T$ do
    $M \leftarrow$ set of edges between vertices with aligned spins in $S(t-1)$
    for edge $E$ in $M$ do
        remove $E = (u,v)$ from M with probability $p_1 = e^{-2J_{uv}}$
    $C \leftarrow$ set of connected components in $G = (V,\ M)$
    for connected component $c$ in $C$ do
        set all spins in $c$ to $\begin{cases} +1 & \text{with probability } p_2 \\ -1 & \text{with probability } 1 - p_2 \end{cases}$
    $S(t) \leftarrow$ new spin configuration
return $S$
```

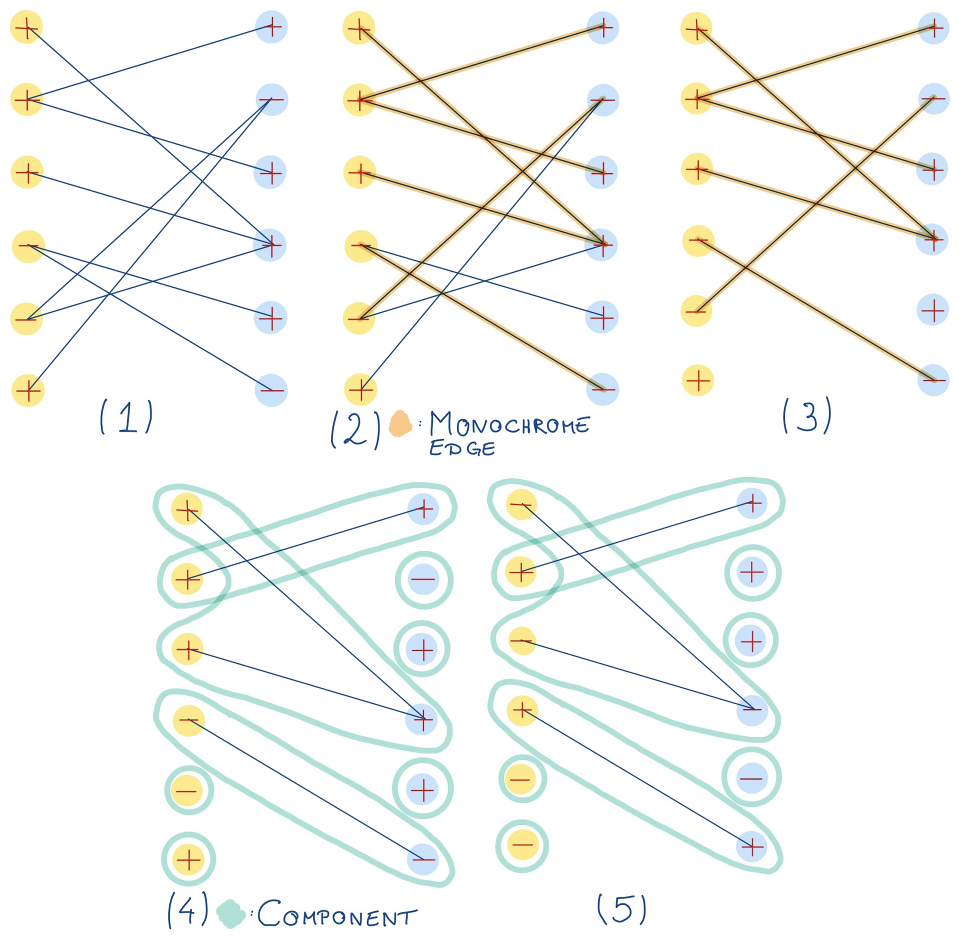


Figure 2.2: Sketch of the Swendsen-Wang Algorithm Steps

Chapter 3

The Learning Algorithm

Goel presents a greedy algorithm which analyses the empirical covariances between observed nodes in a given set of samples. It considers one particular node u at a time and constructs a proposed *two-hop neighbourhood* set $N_2(u)$.

Definition 4 (Two-hop Neighbourhood) The *two-hop neighbourhood* $N_2(u)$ of an observed vertex u is the set of vertices that share a latent vertex with u:

$$N_2(u) = \{v \in V_{obs} | \, \exists \, k \in V_{lat}.(u,k) \in E \text{ and } (v,k) \in E\}$$

In a second step, the algorithm prunes the proposed neighbourhood set so that only the real neighbours should remain. Goel proves the algorithm's correctness and a lower bound on the number of samples required to reconstruct the neighbourhood with a given probability.

In this chapter a lower bound on the covariance between two-hop neighbours will be described, which is the key structural result in Goel's paper. Then, the algorithm itself will be explained and finally, the proof of correctness given by Goel will be examined.

3.1 Structural Result on Covariance

Goel establishes a lower bound on covariances between two-hop neighbours. This is used as a stopping condition for the greedy inclusion of high covariance vertices into the proposed neighbourhood set (i.e. nodes with a covariance smaller than this threshold are very unlikely to be in the actual two-hop neighbourhood).

First, some notation for the structure of the model and the covariance will be introduced. Consider the following bounds on the parameters of the Ising model:

Definition 5 (Goel [7]) An RBM is said to be (α, λ)-locally consistent if the following conditions are satisfied:

1. the RBM is locally consistent (see def. 2)
2. $\forall\ (i, j) \in$ edges E. $|J_{ij}| \geq \alpha$
3. $\forall\ i \in [n]$. $\sum_j |J_{ij}| + |h_i| \leq \lambda$
4. $\forall\ j \in [m]$. $\sum_i\ |J_{ij}| + |g_j| \leq \lambda$

Condition 1 is the local consistency requirement already discussed in section 2.1.3. Condition 2 requires a general lower bound on the influence of each edge. This ensures that all edges are strong enough to be detected. Finally, condition 3 and 4 establish that all fields incident on a node (that is, all edges incident and the external field of the node) are bounded above by the constant λ. This is used to bound the probability of vertex outcomes in the proof of lemma 1.

Furthermore, Goel defines the conditional covariance between two observed vertices u and v conditioned on some set S of observed vertices $S \subseteq V_{obs} \setminus \{u, v\}$ as:

$$\mathrm{Cov}(u, v \mid X_S = \underline{x}_S) := \mathbb{E}[\,X_u X_v \mid X_S = \underline{x}_S] - \mathbb{E}[\,X_u \mid X_S = \underline{x}_S]\ \mathbb{E}[\,X_v \mid X_S = \underline{x}_S] \quad (3.1)$$

The condition $(X_S = \underline{x}_S)$ means that all vertices in S are assumed to have a fixed value (-1 or $+1$) given by the elements of the vector $\underline{x}_S$. The expectation values are taken over all possible values of X_u and/or X_v i.e. $\{(+,+);\ (+,-);\ (-,+); (-,-)\}$ or $\{+;-\}$. She also defines an average conditional covariance:

$$\mathrm{Cov}^{avg}(u, v \mid S) := \mathbb{E}_{\underline{x}_S}[\,\mathrm{Cov}(u, v \mid X_S = \underline{x}_S)] \quad (3.2)$$

which takes a second expectation over all possible value assignments $\{\pm 1\}^{|S|}$ to the vertices in S. This allows Goel to state:

Lemma 1 (Goel [7]) *Consider a fixed node u and any fixed subset of observed nodes $S \subseteq V_{obs} \setminus \{u\}$ with configuration $\underline{x}_S$, then for all $v \in N_2(u) \setminus S$:*

$$\mathrm{Cov}(u, v \mid X_S = \underline{x}_S) \geq \alpha^2 \exp(-12\lambda) \quad (3.3)$$

$N_2(u)$: two-hop-neighbourhood of vertex u

For the proof of this lemma Goel first observes that the model can be treated as ferromagnetic ($J_{ij} \geq 0\ \forall\ (i, j) \in E$ as discussed in section 2.1.3) and asserts that without loss of generality, $S = \emptyset$ can be assumed. This is sensible because conditioning on $X_S = \underline{x}_S$ fixes the nodes in S to a certain value. Then, the conditional distribution can be represented by another Ising model with a vertex set $V'_{obs} = V_{obs} \setminus S$ and modified external field vectors. I.e. $\underline{h}'$, the external field on the observed vertices, contains fewer elements and $\underline{g}'$, the latent external field, absorbs the influence of the fixed vertices. We justify this statement here:

Proposition 3 *Given an (α, λ)-locally consistent Ising model on a graph $G = (V, E)$ with probability distribution p and a set $S \subseteq V_{obs}$ with a fixed spin assignment $\underline{x}_S$, there exists an (α, λ)-locally consistent Ising model on a graph $G' = (V', E')$ with probability distribution $\hat{p}$ such that G' is a subgraph of G and*

$$p(X = \underline{x}', Y = \underline{y} |\, X_S = \underline{x}_S) = \hat{p}(X' = \underline{x}', Y = \underline{y}) \tag{3.4}$$

for all $\underline{x}' \in \{\pm 1\}^{n-|S|}$, the value assignments to vertices $v \notin S$, and $\underline{y} \in \{\pm 1\}^m$

Proof. Consider the conditional probability:

$$\begin{aligned} &p(X = \underline{x}', Y = \underline{y} |\, X_S = \underline{x}_S) \\ &= \frac{\exp\left(\sum_{\substack{i \in [n] \setminus S \\ j \in [m]}} x'_i J_{ij} y_j + \sum_{\substack{s \in S \\ j \in [m]}} x_s J_{sj} y_j + (\underline{h}')^\top \underline{x}' + \underline{h}_S^\top \underline{x}_S + \underline{g}^\top \underline{y} \right)}{Z \cdot p(X_S = \underline{x}_S)} \end{aligned} \tag{3.5}$$

Note that $\underline{h}_S^\top \underline{x}_S = C$ and $x_s J_{sj} = c_{sj}$ are constants for every $s \in S$ and $j \in [m]$. And again:

$$p(X_S = \underline{x}_S) = \frac{1}{Z} \sum_{\substack{\underline{x}' \in \{\pm 1\}^{n-|S|} \\ \underline{y} \in \{\pm 1\}^m}} \exp\left(\sum_{\substack{i \in [n] \setminus S \\ j \in [m]}} x'_i J_{ij} y_j + \sum_{\substack{s \in S \\ j \in [m]}} x_s J_{sj} y_j + (\underline{h}')^\top \underline{x}' + \underline{h}_S^\top \underline{x}_S + \underline{g}^\top \underline{y} \right) \tag{3.6}$$

$$\Rightarrow Z p(X_S = \underline{x}_S) = \exp(C) \sum_{\substack{\underline{x}' \in \{\pm 1\}^{n-|S|} \\ \underline{y} \in \{\pm 1\}^m}} \exp\left(\sum_{\substack{i \in [n] \setminus S \\ j \in [m]}} x'_i J_{ij} y_j + (\underline{h}')^\top \underline{x}' + \sum_{\substack{s \in S \\ j \in [m]}} c_j y_j + \underline{g}^\top \underline{y} \right) \tag{3.7}$$

$$= \exp(C) \cdot \hat{Z} \tag{3.8}$$

where $\hat{Z}$ can be recognised as the partition function of an Ising model with $V'_{obs} = V_{obs} \setminus S$ and external field $\underline{g}' = (\underline{c} + \underline{g})$ where the j^{th} element of c is $c_j = \sum_{s \in S} c_{sj}$
Therefore:

$$p(X = \underline{x}', Y = \underline{y} |\, X_S = \underline{x}_S) = \frac{\exp(C)}{\exp(C)\hat{Z}} \exp\left(\sum_{\substack{i \in [n] \setminus S \\ j \in [m]}} x'_i J_{ij} y_j + (\underline{h}')^\top \underline{x}' + (\underline{g} + \underline{c})^\top \underline{y} \right) \tag{3.9}$$

$$= \frac{1}{\hat{Z}} \exp\left(\sum_{\substack{i \in [n] \setminus S \\ j \in [m]}} x'_i J_{ij} y_j + (\underline{h}')^\top \underline{x}' + (\underline{g} + \underline{c})^\top \underline{y} \right) \tag{3.10}$$

$$= \hat{p}(X' = \underline{x}', Y = \underline{y}) \tag{3.11}$$

Finally, check whether the Ising model described by eq. 3.11 is (α, λ)-locally consistent. By def. of (α, λ)-local consistency (Def. 2):

1. Local consistency is obeyed because the E' is a subset of E and therefore no edges are added or change sign
2. α bounds all elements of J' from below as no new edges are added
3. The influence on the remaining observed nodes V'_{obs} is the same as in V_{obs} because their adjacent edges and external fields do not change, so they are still bounded above by λ
4. previously $\forall\ j \in [m].\ \sum_i\ |J_{ij}| + |g_j| \leq \lambda$
   For some j:

$$\sum_{i\in[n]} |J_{ij}| + |g_j| = \sum_{i\in[n]\backslash S} |J_{ij}| + \sum_{s\in S} |J_{sj}| + |g_j| = \sum_{i\in[n]\backslash S} |J_{ij}| + |g'_j| \tag{3.12}$$

$$\text{Therefore, } \sum_{i\in[n]\backslash S} |J_{ij}| + |g'_j| \leq \lambda \tag{3.13}$$

Hence, $\hat{p}$ is also (α,λ)-locally consistent. □

Since the covariance bound only depends on the parameters α and λ the same bound proposed in lemma 1 must hold for both Ising models.

For the proof of lemma 1, Goel establishes that the covariance between two-hop neighbours can only be positive (following an argument proposed by Percus in [22]). Then, she derives a lower bound on the covariance extending Percus's method of studying two identical copies of the Ising model together.

Note that the bound established in lemma 1 turns out to be very small in experiments. This leads to practical problems which will be discussed in the experimental section (see Section 5.1.1).

3.2 The Structure Learning Algorithm

3.2.1 Setup

The algorithm LEARNRBMNBHD[1] (see page 17) described by Goel takes a vertex u, a threshold τ and a set of M samples (where one sample is the assignment of spin values to each vertex in V_{obs} i.e. a vector $\underline{x} \in \{\pm 1\}^n$) as inputs and returns the exact two-hop neighbourhood of u with a probability of $1-\zeta$.

The parameter τ is set to the lower bound on the covariance between u and its two-hop neighbours. τ can be set to the lowest possible bound established in lemma 1 or if some additional information is available to some higher threshold which results in better recovery for smaller sample sizes (see Section 5.1.1 for a more detailed discussion).

The algorithm relies heavily on the covariance between vertices. However, as the algorithm has only access to a finite number of samples from the Ising model spin outputs, it can only estimate the covariance between vertices. Therefore, define an *empirical covariance* estimate.

[1] Algorithm reproduced from [7] with some modifications

Definition 6 (Empirical Conditional Covariance [2]) The *empirical conditional sample covariance* of a set of samples $Z = \{\underline{x}^{(1)} ... \underline{x}^{(M)}\}$ from an Ising model conditioned on a set S with nodes $u, v \in V_{obs} \setminus S$ is given by:

$$\widehat{Cov}_{avg}(u, v|S) = \frac{1}{|Z_S|} \sum_{\underline{x}_S \in Z_S} \widehat{Cov}(u, v|X_S = \underline{x}_S) \tag{3.14}$$

where $Z_S = \{\underline{x}_S |\ \underline{x}_S$ occurs in some sample $\underline{x}^{(i)} \in Z\}$: set of distinct spin assignment vectors to the vertices in S that occur in Z [3]
X_S: set of random variables associated with the vertices in set S (as defined earlier)
$\#(c) = \sum_{\underline{x} \in Z} \mathbb{I}_{\{c(\underline{x})\}}$: number of samples for which condition c holds and

$$\begin{aligned}\widehat{Cov}(u, v|X_S = \underline{x}_S) = & \sum_{\substack{\underline{x}_u \in \{\pm 1\} \\ \underline{x}_v \in \{\pm 1\}}} x_u x_v \frac{\#(X_u = x_u, X_v = x_v, X_S = \underline{x}_S)}{\#(X_S = \underline{x}_S)} \\ & - \left(\sum_{\underline{x}_u \in \{\pm 1\}} x_u \frac{\#(X_u = x_u, X_S = \underline{x}_S)}{\#(X_S = \underline{x}_S)} \right) \left(\sum_{\underline{x}_v \in \{\pm 1\}} x_v \frac{\#(X_v = x_v, X_S = \underline{x}_S)}{\#(X_S = \underline{x}_S)} \right) \end{aligned} \tag{3.15}$$

3.2.2 Algorithm

Algorithm 3: LEARNRBMNBHD(Vertex u, Threshold τ, Samples $\underline{x}^{(1)} ... \underline{x}^{(M)}$)

```
$S \leftarrow \emptyset$
while $\max_{v \in [n] \setminus S \cup \{u\}} \{\widehat{\text{Cov}}_{\text{avg}}(u, v|\ S)\} \geq \tau$ do
    $i^* \leftarrow \arg\max_{v \in [n] \setminus S \cup \{u\}} \{\widehat{\text{Cov}}_{\text{avg}}(u, v|\ S)\}$
    $S \leftarrow S \cup \{i^*\}$
for $v \in S$ do
    if $\widehat{Cov}_{avg}(u, v|\ S \setminus \{v\}) \leq \tau$ then
        $S \leftarrow S \setminus \{v\}$
return $S$
```

Consider the steps of Goel's algorithm.
First, an empty set S is initialised, which represents the proposed two-hop neighbourhood of vertex u. Next, the observed vertex v with the highest average sample covariance is greedily chosen from the vertices not yet included in S (i.e. $\max_v \{\widehat{Cov}_{avg}(u, v|S)\}$ where $v \in V_{obs} \setminus (S \cup \{u\})$) and added to S. This loop is executed until all remaining vertices have a covariance with u lower than the threshold τ. Finally, each vertex in the proposed neighbourhood S is examined again and removed if its covariance with u conditioned on the rest of S turns out to be smaller than the threshold.

[2] Goel gives a definition but no equation for calculating the empirical conditional covariance, so this specific definition was deduced from the usage in the algorithm

[3] I.e. if all possible spin configurations for S occur in the samples Z then $Z_S = \{\pm 1\}^{|S|}$. However, if the set of samples is small, not all of these assignments might occur giving $Z_S \subset \{\pm 1\}^{|S|}$.

The intuition behind this algorithm is that the larger the covariance between nodes u and v the more likely it is that v is in the two-hop neighbourhood of u because their spins seem to align. The threshold τ is associated with the lower bound on the covariance of two-hop neighbours (lemma 1) and ensures therefore that all possible neighbourhood vertices get included into S.

At first glance, the pruning step seems to repeat the procedure of comparing covariances with τ. Note however, that the conditioning set S now contains the full two-hop neighbourhood of u.

Consider some vertex $v \in N_2(u)$. By lemma 1, v will have a covariance with u greater than τ regardless of the conditioning on the rest of S and hence remains in S. This means that $N_2(u) \subseteq S$ is maintained throughout the pruning loop.

However, if $v \notin N_2(u)$, then the covariance of v with u conditioned S should theoretically be zero (u is effectively screened from the influence of v by all the two-hop neighbours in S, see fig. 3.1). Therefore, v is removed from S. For a rigorous argument the differences between the theoretical and empirical covariance have to be taken into account. This will be considered in section 3.3.1.

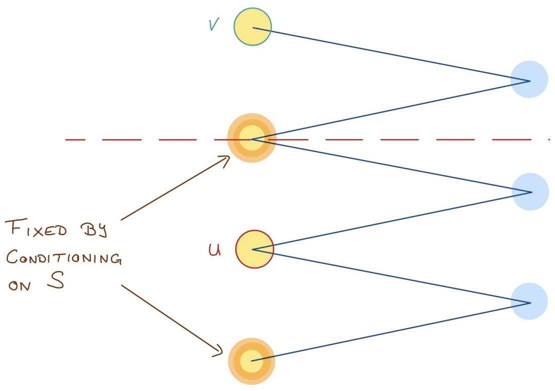


Figure 3.1: Example of vertex u, its neighbourhood S and a non-neighbour v

3.2.3 Sample Size Bound

Goel shows an upper bound on the sample size:

Theorem 1 (Goel [7]) *The algorithm* LEARNRBMNBHD *returns the exact two-hop neighbourhood of a vertex u with probability* $1-\zeta$ *if the number of given samples M is bounded by:*

$$M \geq \Omega\left((\log(\frac{1}{\zeta}) + T^*\log(n)) \frac{2^{2T^*}}{\tau^2\ \delta^{2T^*}} \right) \tag{3.16}$$

$$\text{where } \tau = \frac{\alpha^2}{2}\exp(-12\lambda))^2 \tag{3.17}$$

$$\delta = \frac{1}{2}e^{-\lambda} \tag{3.18}$$

$$T^* = \frac{8}{\tau^2} = \frac{8}{(\frac{\alpha^2}{2}\exp(-12\lambda))^2} \tag{3.19}$$

Note that the threshold τ is set to $\frac{1}{2}$ of the theoretical lower bound on neighbouring node covariances (lemma 1). This allows the empirical covariances to differ by at most $\frac{\tau}{2}$ from the theoretically expected value for the algorithm to still work. The required

accuracy of the empirical covariance then leads to a lower bound on the number of samples M required.
Furthermore, observe that τ is a very small value for any reasonable choices of α and λ , which leads to huge lower bounds on the sample size. This is a problem for the experimental evaluation of the algorithm (as will be discussed in 5.1.1).
δ is a variable dependent on the structure of the graph specifically an upper bound on the total weight of all vertices and external field incident upon any vertex.
Finally, T^* turns out to be an upper bound on the possible size of the neighbourhood set S.

3.3 Proof of Correctness

The proof of theorem 1 as presented by Goel has three steps:

1. Show that for a certain sample size the empirical covariance estimates and the theoretically expected covariance values are closer than a given difference
2. Show that the algorithm terminates in no more than T^* steps
3. Show that after the refining step $S = N_2(u)$

We will now examine Goel's proof, explain the steps and fill in the details.

3.3.1 Closeness of Covariance Estimates

Goel defines the event $\mathcal{A}$ describing the closeness of estimated and theoretical covariance:

Definition 7 (Goel [7]) Let $\mathcal{A}(\ell,\varepsilon)$ be the event such that for vertices u, v and a conditioning set S with $|S| \leq \ell$:

$$\left|\widehat{\text{Cov}}^{\text{avg}}(u,v|S) - \text{Cov}^{\text{avg}}(u,v|S)\right| \leq \varepsilon \tag{3.20}$$

This allows Goel to state:

Lemma 2 (Goel [7]) *For a fixed ℓ (upper bound on the size of the conditioning set S), ε (precision) and ζ (1 - probability) with $\ell,\varepsilon,\zeta \geq 0$, if the number of samples $M \geq \Omega\left((\log(\frac{1}{\zeta}) + \ell\log(n))\frac{2^{2\ell}}{\tau^2\,\delta^{2\ell}}\right)$ then the probability of $\mathcal{A}$ occurring is: $Pr(\mathcal{A}(\ell,\varepsilon)) \geq 1-\zeta$.*

Goel gives the proof in her paper, providing the equations with little commentary. We explain and justify the argument, sometimes inserting necessary intermediate steps:

Proof. Let m be the number of samples. Goel states that for any subset $W \subseteq V_{obs}$ and configuration $\underline{x}_W \in \{\pm 1\}^{|W|}$:

$$Pr\left(\left|\widehat{Pr}(X_W = \underline{x}_W) - Pr(X_W = \underline{x}_W)\right| \geq \gamma\right) \leq 2\exp\left(-2\gamma^2 m\right) \tag{3.21}$$

Where the difference γ will be associated with ε from definition 7 later on.
We identify this as the Hoeffding bound [19] see Appendix B where we observe that

$$\begin{aligned} \mathbb{E}\left[\widehat{Pr}\left(X_W = \underline{x}_W\right)\right] &= \frac{1}{m}\sum_m \mathbb{E}\left[\mathbb{I}_{\{X_W = \underline{x}_W\}}\right] = \frac{1}{m}\sum_m Pr\left(X_W = \underline{x}_W\right) \\ &= \frac{m}{m} Pr\left(X_W = \underline{x}_W\right) = Pr\left(X_W = \underline{x}_W\right) \end{aligned} \tag{3.22}$$

The Hoeffding bound is an upper bound on the probability that the empirical probability estimate differs by more than γ from the real probability. However, Goel requires a bound for the probability including not just one specific subset W and one assignment vector $\underline{x}_W$, but all possible subsets and all assignment vectors. Therefore, it is necessary to consider how many subsets with $|W| \leq \ell + 2$ (ℓ: maximum size of S plus the vertices u and v) and vector assignments $\underline{x}_W$ exist:

$$\#(\{\underline{x}_W\}) = \sum_{k=1}^{\ell+2} \binom{n}{k} 2^k \tag{3.23}$$

$\binom{n}{k}$ for number of ways to pick k vertices out of n
2^k for the number of ± 1 assignments to a vertex set of size k
Then, Goel roughly bounds this sum from above by observing:

$$\binom{n}{k} \leq n^k \Rightarrow \#(\{\underline{x}_W\}) \leq \sum_{k=1}^{\ell+2} (2n)^k \leq \sum_{k=1}^{\ell+2} (2n)^{\ell+2} = (\ell+2)(2n)^{\ell+2} \tag{3.24}$$

(Where we supplied some of the in between steps)
Now, introduce a probability ζ as an upper bound on the probability of getting at least one estimate $\left|\widehat{Pr} - Pr\right| \geq \gamma$. This is equivalent to a lower bound on the probability $1 - \zeta$ for finding all $\left|\widehat{Pr} - Pr\right| \leq \gamma$.
ζ must be:

$$\zeta \geq \#(\{\underline{x}_W\}) \times 2\exp\left(-2\gamma^2 m\right) \tag{3.25}$$

i.e. $\#(\{\underline{x}_W\})$ tries for an event with probability $2\exp\left(-2\gamma^2 m\right)$. Now, solving for m, we satisfy inequality 3.25:

$$(3.25) \Leftrightarrow \quad \frac{\zeta}{2 \times \#(\{\underline{x}_W\})} \geq \exp\left(-2\gamma^2 m\right) \tag{3.26}$$

$$\Leftrightarrow \quad \log(\zeta) - \log(2 \times \#(\{\underline{x}_W\})) \geq -2\gamma^2 m \tag{3.27}$$

$$\Leftrightarrow \quad m \geq \frac{-\log(\zeta) + \log(2 \times \#(\{\underline{x}_W\}))}{2\gamma^2} \tag{3.28}$$

$$\Leftrightarrow \quad m \geq \frac{\log\left(\frac{1}{\zeta}\right) + \log\left(2(\ell+2)(2n)^{\ell+2}\right)}{2\gamma^2} \tag{3.29}$$

$$\Leftrightarrow \quad m \geq \frac{\log\left(\frac{1}{\zeta}\right) + \log(2(\ell+2)) + (\ell+2)\log(2n)}{2\gamma^2} \tag{3.30}$$

Based on this expression, Goel then establishes an upper bound on the covariance difference ε, which applies if $|S| \leq \ell$ and γ is chosen appropriately. The choice of γ substituted into 3.30 then gives the required bound on the sample size.

$$\left|\widehat{\text{Cov}}^{\text{avg}}(u,v|S) - \text{Cov}^{\text{avg}}(u,v|S)\right| \tag{3.31}$$

Expansion of the covariance definitions:

$$\begin{aligned}
&= \Big|\widehat{\mathbb{E}}_{x_S}\left[\widehat{\mathbb{E}}\left[X_uX_v|X_S=\underline{x}_S\right] - \widehat{\mathbb{E}}\left[X_u|X_S=\underline{x}_S\right]\widehat{\mathbb{E}}\left[X_v|X_S=\underline{x}_S\right]\right] \\
&- \mathbb{E}_{x_S}\left[\mathbb{E}\left[X_uX_v|X_S=\underline{x}_S\right] - \mathbb{E}\left[X_u|X_S=\underline{x}_S\right]\mathbb{E}\left[X_v|X_S=\underline{x}_S\right]\right]\Big|
\end{aligned} \tag{3.32}$$

By definition of expectation and linearity of expectation:

$$\begin{aligned}
= \Bigg| \sum_{x_u,x_v=\pm 1} x_ux_v \Bigg\{ &\widehat{\mathbb{E}}_{x_S}\Big[\widehat{Pr}\left(X_u=x_u,X_v=x_v|X_S=\underline{x}_S\right) \\
&\qquad -\widehat{Pr}\left(X_u=x_u|X_S=\underline{x}_S\right)\widehat{Pr}\left(X_v=x_v|X_S=\underline{x}_S\right)\Big] \\
&- \mathbb{E}_{x_S}\Big[Pr\left(X_u=x_u,X_v=x_v|X_S=\underline{x}_S\right) \\
&\qquad - Pr\left(X_u=x_u|X_S=\underline{x}_S\right)Pr\left(X_v=x_v|X_S=\underline{x}_S\right)\Big]\Bigg\}\Bigg|
\end{aligned} \tag{3.33}$$

The expectation over $\underline{x}_S$ gets resolved and $Pr\left(X_S=\underline{x}_S\right)$ gets absorbed into the conditional probabilities:

$$\begin{aligned}
= \Bigg| \sum_{x_u,x_v,\underline{x}_S} x_ux_v \Bigg\{ &\Big[\widehat{Pr}\left(X_u=x_u,X_v=x_v,X_S=\underline{x}_S\right) \\
&-\widehat{Pr}\left(X_u=x_u,X_S=\underline{x}_S\right)\widehat{Pr}\left(X_v=x_v|X_S=\underline{x}_S\right)\Big] \\
&- \Big[Pr\left(X_u=x_u,X_v=x_v,X_S=\underline{x}_S\right) \\
&- Pr\left(X_u=x_u,X_S=\underline{x}_S\right)Pr\left(X_v=x_v|X_S=\underline{x}_S\right)\Big]\Bigg\}\Bigg|
\end{aligned} \tag{3.34}$$

Note that the sum has positive and negative elements ($x_ux_v = \pm 1$), therefore it can only get larger if the negative elements are turned positive:

$$\begin{aligned}
&\leq \sum_{x_u,x_v,\underline{x}_S} \Big|\Big[\widehat{Pr}\left(X_u=x_u,X_v=x_v,X_S=\underline{x}_S\right) - \widehat{Pr}\left(X_u=x_u,X_S=\underline{x}_S\right)\widehat{Pr}\left(X_v=x_v|X_S=\underline{x}_S\right)\Big] \\
&- \left[Pr\left(X_u=x_u,X_v=x_v,X_S=\underline{x}_S\right) - Pr\left(X_u=x_u,X_S=\underline{x}_S\right)Pr\left(X_v=x_v|X_S=\underline{x}_S\right)\right]\Big|
\end{aligned} \tag{3.35}$$

Recognise $\gamma \geq \left|\widehat{Pr}\left(X_u=x_u,X_v=x_v,X_S=\underline{x}_S\right) - Pr\left(X_u=x_u,X_v=x_v,X_S=\underline{x}_S\right)\right|$ where

$\{u\} \cup \{v\} \cup S = W$ and therefore the condition $|W| \leq |S|+2$ holds

$$\begin{aligned}\Rightarrow \leq 2^{|S|+2}\gamma + \sum_{x_u,x_v,\underline{x}_S} \Big| & Pr\left(X_u = x_u, X_S = \underline{x}_S\right) Pr\left(X_v = x_v|X_S = \underline{x}_S\right) \\ & - \widehat{Pr}\left(X_u = x_u, X_S = \underline{x}_S\right) \widehat{Pr}\left(X_v = x_v|X_S = \underline{x}_S\right) \Big|\end{aligned} \tag{3.36}$$

Now bound the second term:

$$\begin{aligned} & \Big| Pr\left(X_u = x_u, X_S = \underline{x}_S\right) Pr\left(X_v = x_v|X_S = \underline{x}_S\right) \\ & - \widehat{Pr}\left(X_u = x_u, X_S = \underline{x}_S\right) \widehat{Pr}\left(X_v = x_v|X_S = \underline{x}_S\right) \Big| \\ = & \Big| Pr\left(X_u = x_u, X_S = \underline{x}_S\right) Pr\left(X_v = x_v|X_S = \underline{x}_S\right) \\ & - \widehat{Pr}\left(X_u = x_u, X_S = \underline{x}_S\right) \widehat{Pr}\left(X_v = x_v|X_S = \underline{x}_S\right) \\ & + Pr(X_u = x_u, X_S = \underline{x}_S)\widehat{Pr}(X_v = x_v|X_S = \underline{x}_S) \\ & - Pr(X_u = x_u, X_S = \underline{x}_S)\widehat{Pr}(X_v = x_v|X_S = \underline{x}_S) \Big| \end{aligned} \tag{3.37}$$

$$\begin{aligned} \leq & \underbrace{Pr\left(X_u = x_u, X_S = \underline{x}_S\right)}_{\leq 1} \left| \widehat{Pr}(X_v = x_v|X_S = \underline{x}_S) - Pr\left(X_v = x_v|X_S = \underline{x}_S\right) \right| \\ & + \underbrace{\widehat{Pr}\left(X_v = x_v|X_S = \underline{x}_S\right)}_{\leq 1} \underbrace{\left| \widehat{Pr}\left(X_u = x_u, X_S = \underline{x}_S\right) - Pr(X_u = x_u, X_S = \underline{x}_S) \right|}_{\leq \gamma} \end{aligned} \tag{3.38}$$

remove multiplication by factors ≤ 1 and by the definition of conditional probability:

$$\leq \gamma + \left| \frac{\widehat{Pr}(X_v = x_v, X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} - \frac{Pr\left(X_v = x_v, X_S = \underline{x}_S\right)}{Pr(X_S = \underline{x}_S)} \right| \tag{3.39}$$

Add the terms $\frac{Pr(X_v=x_v,X_S=\underline{x}_S)}{\widehat{Pr}(X_S=\underline{x}_S)}$ and $-\frac{Pr(X_v=x_v,X_S=\underline{x}_S)}{\widehat{Pr}(X_S=\underline{x}_S)}$ (which sum to zero) into the expression between the absolute value bars. Break up the absolute value expression which can only increase its value (by the triangle inequality):

$$\begin{aligned} \leq \gamma + & \underbrace{\left| \frac{\widehat{Pr}(X_v = x_v, X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} - \frac{Pr(X_v = x_v, X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} \right|}_{(I)} \\ + & \underbrace{\left| \frac{Pr(X_v = x_v, X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} - \frac{Pr\left(X_v = x_v, X_S = \underline{x}_S\right)}{Pr(X_S = \underline{x}_S)} \right|}_{(II)} \end{aligned} \tag{3.40}$$

Consider the second (I) and third term (II) separately. We define (I) and (II) as:

$$(I) = \left| \frac{\widehat{Pr}(X_v = x_v, X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} - \frac{Pr(X_v = x_v, X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} \right| \tag{3.41}$$

$$(II) = \left| \frac{Pr(X_v = x_v, X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} - \frac{Pr\left(X_v = x_v, X_S = \underline{x}_S\right)}{Pr(X_S = \underline{x}_S)} \right| \tag{3.42}$$

By the definition of γ:

$$(I) \leq \frac{\gamma}{\widehat{Pr}(X_S = \underline{x}_S)} \tag{3.43}$$

Now, bound the denominator from above, which gives a lower bound on (I):

$$\widehat{Pr}(X_S = \underline{x}_S) = Pr(X_S = \underline{x}_S) \pm \gamma \geq Pr(X_S = \underline{x}_S) - \gamma \geq \delta^{|S|} - \gamma \tag{3.44}$$

where $\delta = \frac{1}{2}e^{-\lambda}$ and Bresler [1] shows that this is a lower bound on the conditional probability of any vertex having any spin. Therefore:

$$(I) \leq \frac{\gamma}{\delta^{|S|} - \gamma} \tag{3.45}$$

Next consider (II):

$$(II) = \underbrace{\frac{Pr(X_v = x_v, X_S = \underline{x}_S)}{Pr(X_S = \underline{x}_S)}}_{\leq 1} \left(\frac{Pr(X_S = \underline{x}_S)}{\widehat{Pr}(X_S = \underline{x}_S)} - 1 \right) \leq \frac{\delta^{|S|}}{\delta^{|S|} - \gamma} - 1 = \frac{1}{1 - \frac{\gamma}{\delta^{|S|}}} - 1 \tag{3.46}$$

Assume that $\frac{\gamma}{\delta^{|S|}}$ is small (this is reasonable to assume because the error on the probability should be small w.r.t. the lowest possible probability) and make a Taylor series expansion:

$$(II) \leq \frac{1}{1 - \frac{\gamma}{\delta^{|S|}}} - 1 = 1 + \frac{\gamma}{\delta^{|S|}} + \left(\frac{\gamma}{\delta^{|S|}}\right)^2 \sum_{n=0}^{\infty} \left(\frac{\gamma}{\delta^{|S|}}\right)^n - 1 = \frac{\gamma}{\delta^{|S|}} \left[+ \left(\frac{\gamma}{\delta^{|S|}}\right)^2 \frac{1}{1 - \frac{\gamma}{\delta^{|S|}}} \right] \tag{3.47}$$

Putting it all together:

$$\left| \widehat{\mathrm{Cov}}^{\mathrm{avg}}(u, v|S) - \mathrm{Cov}^{\mathrm{avg}}(u, v|S) \right| \leq 2^{|S|+2} \left(2\gamma + \frac{\gamma}{\delta^{|S|} - \gamma} + \frac{\gamma}{\delta^{|S|}} \left[+ \left(\frac{\gamma}{\delta^{|S|}}\right)^2 \frac{1}{1 - \frac{\gamma}{\delta^{|S|}}} \right] \right) \tag{3.48}$$

Then, Goel chooses $\gamma \leq \varepsilon 2^{-\ell} \frac{\delta^{\ell}}{20}$ to show that the difference between the empirical and theoretical covariance is bounded above by ε. First, let $\frac{\gamma}{\delta^{|S|}} = \eta$

$$2\gamma + \frac{\gamma}{\delta^{|S|} - \gamma} + \frac{\gamma}{\delta^{|S|}} \left[+ \left(\frac{\gamma}{\delta^{|S|}}\right)^2 \frac{1}{1 - \frac{\gamma}{\delta^{|S|}}} \right] = 2\gamma + \eta \frac{1}{1 - \eta} + \eta + \left(\eta^2 \frac{1}{1 - \eta} \right) \tag{3.49}$$

$$= 2\gamma + \eta \left(1 + \frac{1 + \eta}{1 - \eta} \right) = 2\gamma + 2\frac{\eta}{1 - \eta} \tag{3.50}$$

bound η, where we observe that $0 \leq \varepsilon, 2^{-\ell}, \delta^{\ell - |S|} \leq 1$ as $\ell > 0$ and $\ell \geq |S|$

$$\eta = \frac{\gamma}{\delta^{|S|}} \leq \frac{1}{20} \varepsilon 2^{-\ell} \delta^{\ell - |S|} \leq \frac{1}{20} \tag{3.51}$$

Therefore:

$$\left|\widehat{\mathrm{Cov}}^{\mathrm{avg}}(u,v|S) - \mathrm{Cov}^{\mathrm{avg}}(u,v|S)\right| \leq 2^{|S|+2}\left(2\gamma + \frac{2\gamma}{19\delta^{|S|}}\right) \tag{3.52}$$

$$= 2^{|S|+2}2^{-\ell}\varepsilon\frac{\delta^{\ell}}{20}\left(2+\frac{2}{19\delta^{|S|}}\right) = \varepsilon\underbrace{2^{|S|-\ell}}_{\leq 1}\frac{4}{20}\left(2\underbrace{\delta^{\ell}}_{\leq 1}+\frac{2}{19}\underbrace{\delta^{\ell-|S|}}_{\leq 1}\right) \tag{3.53}$$

$$\leq \varepsilon\frac{1}{5}\left(2+\frac{2}{19}\right) \leq \varepsilon \tag{3.54}$$

This proves that with probability $Pr\left(\mathcal{A}(\ell,\varepsilon)\right) \geq 1-\zeta$ for

$$m \geq \frac{\log\left(\frac{1}{\zeta}\right)+\log\left(2(\ell+2)\right)+(\ell+2)\log(2n)}{2\gamma^2} \tag{3.55}$$

$$\geq \frac{1}{2\left(\varepsilon 2^{-\ell}\frac{\delta^{\ell}}{20}\right)^2}\frac{\log\left(\frac{1}{\zeta}\right)+\log\left(2(\ell+2)\right)+(\ell+2)\log(2n)}{2\gamma^2} \tag{3.56}$$

$$\in \Omega\left(\frac{2^{2\ell}}{\varepsilon^2\delta^{2\ell}}\left(\ell\log(n)+\log\left(\frac{1}{\zeta}\right)\right)\right) \tag{3.57}$$

□

Definition 8 Let the event $\mathbb{A} = \mathcal{A}(T^*, \frac{\tau}{2})$ for $T^* = \frac{8}{\tau^2}$,the maximum size of set S and $\frac{\tau}{2}$, the maximum difference between empirical and expected covariances.

$\Rightarrow$ if $M \in \Omega\left(\frac{2^{2T^*}}{\tau^2\delta^{2T^*}}\left(T^*\log(n)+\log\left(\frac{1}{\zeta}\right)\right)\right)$ then $\mathbb{A}$ holds with probability $1-\zeta$

From the rest of the argument, Goel assumes that $\mathbb{A}$ holds.

3.3.2 Termination in $T \leq T^*$ steps

First, Goel establishes lemma 3, a relationship between the average covariance and the *mutual information* between two vertices conditioned on some set S from an information theoretic argument. We leave the proof to Goel's paper as it does not provide further insight into the workings of the algorithm. For reference, we first give the definitions of the required information theoretic functions:

Definition 9 (Entropy [4]) The *entropy* of a discrete random variable X with sample space $\mathcal{X}$ is defined as:

$$H(X) = -\sum_{x\in X} p(x)\log p(x) \tag{3.58}$$

The *entropy* is essentially a measure of uncertainty in the probability distribution of X.

Definition 10 (Mutual Information [4]) The *mutual information* of two random variables X and Y with sample spaces $\mathcal{X}$ and $\mathcal{Y}$ is defined by:

$$I(X;Y) = \sum_{\substack{x\in X\\ y\in\mathcal{Y}}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{3.59}$$

The *mutual information* captures the decrease of uncertainty in one variable due to knowledge of the other variable. Note that for $H(X|Y)$, the conditional entropy: $I(X;Y) = H(X) - H(X|Y)$ as given in [4]
It can be seen from definition 9 that the entropy is always non-negative. Therefore, $I(X;Y) \leq H(X)$

Definition 11 (Conditional Mutual Information [4]) The *conditional mutual information* of two random variables X and Y given random variable Z is defined by:

$$I(X;Y|Z) = \sum_{\substack{x \in \mathcal{X} \\ y \in \mathcal{Y} \\ z \in \mathcal{Z}}} p(x,y,z) \log \frac{p(x,y|z)}{p(x|z)p(y|z)} \tag{3.60}$$

Based on these definitions, consider Goel's lemma:

Lemma 3 (Goel [7]) *For vertices $u \neq v \in V_{obs}$ and a subset of vertices $S \subseteq V_{obs} \setminus \{u,v\}$:*

$$\sqrt{2I\left(X_u, X_v|X_S\right)} \leq \text{Cov}^{avg}\left(u,v|S\right) \tag{3.61}$$

Lemma 4 *For algorithm* LEARNRBMNBHD *the size of the proposed neighbourhood set $|S|$ is bounded above by T^**

Proof. Goel considers a sequence of vertices $i_1, \dots, i_T$ consecutively added to the proposed neighbourhood set S and a set $S_\ell = \{i_i, \dots, i_\ell\}$, the set of vertices added up to some number ℓ where $1 \leq \ell \leq T$.
For the j^{th} vertex (where $j \in [T]$) added to S, $\widehat{\text{Cov}}^{\text{avg}}(u, i_j|S_{j-1}) \geq \tau$ (by line 2 in the algorithm).
Now, by contradiction, assume that $T > T^*$. Recall that $\mathbb{A} := \mathcal{A}(T^*, \frac{\tau}{2})$ (Def. 7) is the event that for $|S| \leq T^*$:

$$\left|\widehat{\text{Cov}}^{\text{avg}}(u, i_j|S_{j-1}) - \text{Cov}^{\text{avg}}(u, i_j|S_{j-1})\right| \leq \frac{\tau}{2}$$

Therefore, for all $j \leq T^* + 1$, the difference between the empirical and the theoretic average covariance is smaller than $\frac{\tau}{2}$, leading to the theoretical covariance $\text{Cov}^{\text{avg}}(u, i_j|S_{j-1}) \geq \frac{\tau}{2}$
The contradiction is constructed as follows:

$$1 \geq H\left(X_u\right) \geq I\left(X_u; X_S\right) \tag{3.62}$$

This can be derived from the definition of *entropy* H (Def. 9) and the *mutual information* I (Def. 10). Note that in Goel's paper she uses $I\left(X_u|X_S\right)$ instead of $I\left(X_u;X_S\right)$. However, it seems as though this might be a misprint or a shorthand because mutual information is usually defined between two random variables as opposed to just one random variable with a condition.
By the *chain rule of mutual information* [4]:

$$I\left(X_u;X_S\right) = \sum_{j=1}^{T} \underbrace{I\left(X_u; X_{i_j}|X_{S_{j-1}}\right)}_{\substack{\geq \frac{1}{2}\left(\text{Cov}^{avg}(u,v|S)\right)^2 \\ \text{by Lemma 3}}} \geq \sum_{j=1}^{T} \frac{1}{2}\left(\text{Cov}^{avg}\left(u,v|S\right)\right)^2 \geq \frac{T^*+1}{2}\left(\frac{\tau}{2}\right)^2 \tag{3.63}$$

Recall that $T^* := \frac{8}{\tau^2}$

$$\Rightarrow 1 \geq \frac{\frac{8}{\tau^2}+1}{2}\left(\frac{\tau}{2}\right)^2 = \frac{8+\tau^2}{8} \quad \Rightarrow \text{Contradiction!} \tag{3.64}$$
$$\Rightarrow T \leq T^*$$

□

3.3.3 Termination with $S = N_2(u)$

Lemma 5 *After at most T^* iterations and pruning steps:*

$$S = N_2(u)$$

Proof. Show that after at most T^* iterations $N_2(u) \subseteq S$ and that after pruning only $N_2 = S$ remains.
Goel assumes by contradiction that $N_2(u) \nsubseteq S$. Then, there must be some vertex $v \in N_2(u)$ and $v \notin S$. By lemma 1, $\text{Cov}^{avg}(u,v|S) \geq \alpha^2 \exp(-12\lambda) = 2\tau$
By $\mathbb{A}$, $\widehat{\text{Cov}}^{avg}(u,v|S) \geq \frac{2}{3}\tau \geq \tau$.
This contradicts line 2 of the algorithm (i.e. the node would have been included before termination of the algorithm)
$\Rightarrow$ there is no $v \in N_2(u)$ and $v \notin S$ $\Rightarrow$ Contradiction!
$\Rightarrow N_2(u) \subseteq S$

Consider the set S after pruning and assume by contradiction that there exists a vertex $v \in S$ and $v \notin N_2(u)$. Goel notes that $\text{Cov}^{avg}(u,v|S \setminus \{v\}) = 0$ because conditioned on a set S that contains the full neighbourhood of u, X_v must be independent of X_u by definition of the Ising model.
Again, by $\mathbb{A}, \widehat{\text{Cov}}^{avg}(u,v|S) \leq \frac{\tau}{2}$ which means that v would have been pruned. This contradicts the assumption.
$\Rightarrow v \notin S$ and $v \notin N_2$ $\Rightarrow$ Contradiction!
Goel also remarks that by lemma 1, if instead $v \in N_2(u)$ then $\text{Cov}^{avg}(u,v|S \setminus \{v\}) \geq 2\tau$ and v will not be pruned from S.

$\Rightarrow N_2(u) = S$ after the termination on the algorithm. □

This concludes the proof of correctness for the algorithm LEARNRBMNBHD under assumption of M samples.

Chapter 4

Experimental Setup

The aim of the experiments is to explore how many samples M are required to reliably reconstruct the two-hop neighbourhood of vertices in the graph. We also investigate how structural parameters of the graph influence this number.
In this chapter we explain our Ising model simulation and the Swendsen-Wang sampler as well as the implementation of Goel's algorithm and the required auxiliary functions. For this project, Python (Version 3.9) [23] was used as the programming language because it provides fast, easy-to-use mathematical operations (e.g. via the NumPy package [12]) and data structures for graphs (e.g. thought the NetworkX package [11]), while the performance seemed to be adequate for the problem.

4.1 Sample Generation

4.1.1 Ising Model

To represent an Ising model as described by Goel it is sufficient to store the interaction matrix $J^{\,n\times m}$ and the two external field vectors $\underline{h}$ and $\underline{g}$ as they encapsulate the whole structure of the network.
The model was implemented as a class which could either represent a stored Ising model or generate a new graph at random. In general, there are several properties that can be varied in this type of Ising model:

- the **number of observed vertices** n **and hidden vertices** m can be changed giving a small or large network with vastly different performance (as previously mentioned). n and m can also change with respect to one another, giving an asymmetric network (w.r.t. exchange of the observed and hidden side)
- the **degree of connectedness** can change giving a denser or sparser network, resulting in larger or smaller two-hop neighbourhoods. Note in particular that a density over 0.5 (i.e. each observed node is connected to more than half of the hidden nodes) would immediately result in each observed node being in the two-hop neighbourhood of every other node. This occurs because any two observed nodes u and v : $|N(u)| > \frac{m}{2}$ and $|N(v)| > \frac{m}{2} \Rightarrow |N(u)| + |N(v)| > m$ so (by the pigeonhole principle) the neighbourhoods must overlap in some hidden

node making them automatic neighbours. This also means that any networks with a density higher than 0.5 are indistinguishable for the algorithm.

- the **edges can be assigned at random**, potentially generating many disconnected components **or in a regular pattern**
- the **edge weights can be assigned at random** within a certain interval (of a size to be determined) **or with certain values**. As remarked previously, we restricted the edge weights to be positive (ferromagnetic)
- the **external fields can be assigned at random or with certain values**. However, here all external fields must be allowed to be positive or negative to preserve the correspondence to a locally consistent model (see proposition 1)

This presented the challenge of determining which parameters to vary in experiments.

4.1.2 Random Graph Library

Note that generating new graphs with randomised parameters makes the interpretation of the algorithm harder as this introduces further uncertainty into the model. Therefore, we created a data set of graphs specifying $J, \underline{h}$ and $\underline{g}$ for different combinations of parameters which were then sampled with the sampling algorithm. However, from first explorations it turned out that these random graph structures required huge numbers of samples (something we will come back to in section 5.1.1) which let us to adopt a more structured graph model for further analysis.

4.1.3 Goel's Experimental Setup

We decided to reproduce some of Goel's experiments that are presented in a second paper ”Learning Ising and Potts Models with Latent Variables” [8]. In the paper's Ising Model section Goel reproduces the theoretical insights from [7] and adds an experimental section.

We extend these experiments to larger as well as some denser graphs.

Goel works on graphs with restricted specifications:

- **numbers of vertices** set to $n = m \leq 15$ as Goel uses a direct sampling method. We extend this up to $n = m = 128$ with an approximate sampler
- **degree of each vertex** fixed to 2 in a regular zig-zag pattern (see figure 4.1). This is extended to degree 3 in some experiments (see figure 4.2).
- **edge weights** are all set to 1
- the **external fields** are assigned values ±0.2 or ±0.4 at random

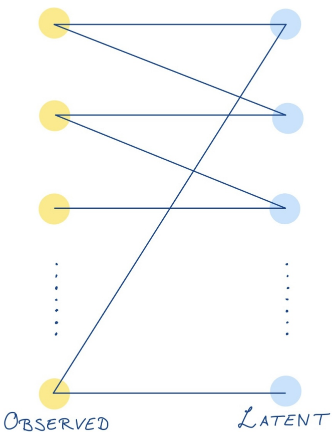


Figure 4.1: Example graph structure used by Goel

We used graphs with $n = m \in \{\, 4,\ 8,\ 16,\ 32,\ 64,\ 128\}$. Furthermore, we generated 5 sets (1 for experimentation and 4 for data collection) of external fields for each configuration and averaged our data over the last four graphs to reduce the dependence of the randomly chosen external fields.

In the setup of our experiments, we made a mistake and assigned the random values to the external fields in a range [- 0.2, 0.2] and [- 0.4, 0.4] as opposed to discrete values ± 0.2 or ± 0.4. This should make the difference between the experiments for the first and the second setting less pronounced as the first interval is a subset of the second. However, as we will discuss in section 5.2.2 we do not observe a large difference to Goel's results.

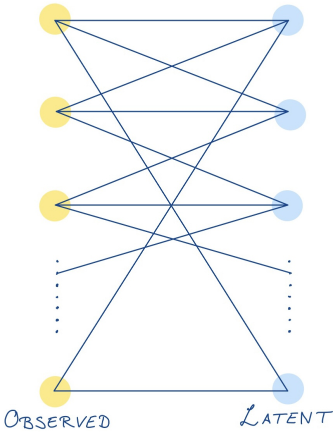


Figure 4.2: Example graph structure with degree 3

4.1.4 The Sampler

As described above, we used a Swendsen-Wang sampler (see algorithm SWENDSEN-WANGSAMPLER on page 12) to generate samples from the Ising models. The algorithm itself is relatively simple to implement as can be seen from the pseudo code. One challenging point was to calculate the connected components of the residual graph. For this we used a function provided by the NetworkX package [11].

We used varying burn-in times (i.e. the number of steps at the beginning that are discarded) and experimented with the number of iterations between recorded samples. For debugging and testing purposes we used small graphs for which the probability distribution can be calculated exactly and compared it with results from the sampler. We tested the sampler with graphs of size up to $n = m = 8$ as our direct calculation breaks down for $n = m = 16$. We then compared the expected and the observed probability distributions and concluded that the distributions seemed to match well.

Then, we went on to explore how the burn-in time affects the accuracy of the observed distribution. From our observations we concluded that the number of samples used to calculate the frequencies has a much stronger influence on the accuracy of the observed distribution than the burn-in time. However, this is probably due to the very small size of the networks and the rapid mixing of the Markov chain.

We also tried to derive an expression for the covariance in the bipartite Ising model in order to check that the covariances in the samples for larger networks were accurate. This however turned out to be infeasible because we did not manage to decouple the distribution over one spin from the distribution of all other spins. Looking back at this exploration now, it could also be interesting to consider the conditional covariances again and explore whether they could be a measure of how well the Markov chain is mixed.

In the end, we concluded that the sampler seemed to be debugged (i.e. working as it should be) but we were not able to establish in general which burn-in times ensure a mixed Markov chain[1].

[1] From personal communication with Mary Cryan et al. we gathered that the theoretical study of a perfect sampler (i.e. a sampler with definite burn-in time) for the RBM model is still ongoing

We decided to choose a presumably large enough value (approx. 1000 steps) and factor in some uncertainty about the sample distribution in the analysis of the data.
Furthermore, we only take approximately every third sample in the chain to reduce the likelihood of obtaining correlated samples (the step size between samples varies slightly to avoid potential cycles in the chain). This step size was chosen because the Swendsen-Wang sampler flips many spins at the same time which means that the correlation between time steps should be relatively small (compared to a single-site update algorithm such as the Gibbs sampler). Furthermore, it became computationally infeasible to generate large numbers of samples with larger intervals.

4.2 Learning Algorithm

Again, the algorithm is simple to implement. However, it relies on the calculation of the conditional covariance of vertices in the sample and a τ value as an input. Here, we will outline how we calculated the covariance and how we set up our automated experiments varying sample sizes and vertex numbers:

4.2.1 Covariance Calculation

Rather than calculate the conditional covariance between specific vertices we calculate the covariance matrix for all the observed nodes at once. We observed that this is computationally cheap through NumPy array operations and to obtain the maximum of these values the learning algorithm requires many of these covariances anyway. The empirical covariance can be calculated from equation 3.15. To generalise this to the covariance matrix:

1. Sort the sample set $Z = \{\underline{x}^{(1)} \dots \underline{x}^{(M)}\}$ by their spin values vector over S and create a set of subsets so that each subset consists of the samples that have the same spin value vector $\underline{x}_S$ with respect to S.
2. For each of these subsets calculate the covariance matrix
3. Take the mean of all subset covariances weighted by the number of samples in each subset (i.e. $\#(X_S = \underline{x}_S)$)

The so-calculated covariance matrix can then be used to find the vertex with the maximum covariance with respect to vertex u.

4.2.2 Testing for fixed-structure Graphs

To reproduce Goel's experiments we measured whether the algorithm recovers the neighbourhood of all observed nodes in a given graph when given a certain number of samples (See the pseudo code EXPERIMENT on page 31).

For the zig-zag graphs (fig, 4.1) and the degree-3 graphs (fig. 4.2) it is easy to derive the expected two-hop neighbourhood which is used to check the algorithm results:

- for zig-zag graphs:
  $N_2(i) = \{(i-1) \mod n,\ (i+1) \mod n\}$

- for degree-3 graphs:
  $N_2(i) = \{(i-2) \mod n,\ (i-1) \mod n,\ (i+1) \mod n,\ (i+2) \mod n\}$
  where i is the index of the input node.

Algorithm 4: EXPERIMENT(IsingModel, maxSampleSize, stepsSize)

```
G = (V, E) ← graph of the IsingModel
for M up to maxSampleSize in steps of stepSize do
    c = 0
    for i = 1 up to 10 do
        generate M samples
        flag ← True
        for v in V do
            execute LEARNRBMNBHD on vertex v
            if returned set is False then
                flag ← False
        if Flag is True then
            c ← c + 1
    store c/10
return List of c fractions, one for each sample size
```

The EXPERIMENT is repeated 4 times for graphs with the same structure but different random values assigned to the external fields. These measurements are then averaged to reduce the influence of the randomly chosen external fields. Then, this setup is repeated for graphs with $n = m \in \{4, 8, 16, 32, 64, 128\}$ nodes. The measurements are conducted for |external fields| ≤ 0.2 and ≤ 0.4.

Chapter 5

Results and Discussion

5.1 First Explorations

We tried the algorithm on small graphs with random edge assignments, choosing structures that show some connectedness. See for example the graph in figure 5.1. We calculated $\alpha^2 \exp(-12\lambda) \approx 10^{-22}$ to be our τ value. When using 10^5 samples the algorithm recovered the correct neighbourhood of a single node only in 7 out of 20 runs. This seemed to be an unsatisfactory result because the graph itself has only a sample space of size $2^8 = 256$ which seems small compared to the number of samples used. After similar experiments with other small graphs we turned to Goel's experiments and noticed that she *"tuned"* her τ threshold. This led us to considering the role of τ in the experiments.

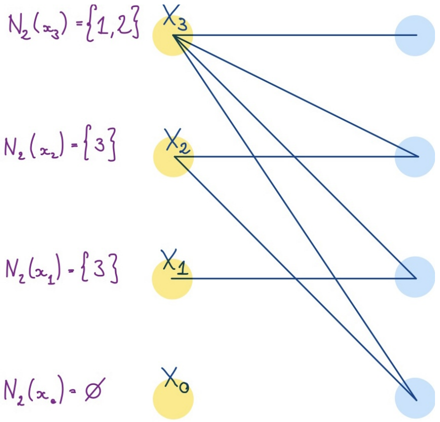


Figure 5.1: Example graph for experiments

5.1.1 The τ Parameter

The τ parameter should be the lower bound on the theoretical conditional covariance for neighbours of vertex v. However, observe that if τ is set to the value suggested by lemma 1 this is a very small value. For example, for an Ising model with a regular structure as proposed by Goel and external fields $\in [-0.2, 0.2]$ (see fig. 4.1):

$\alpha \leq 1$: lower bound on the edge weights
$\lambda \geq 2.2$: upper bound on sum of adjacent edges and external field for any node

$$\Rightarrow \tau = \alpha^2 \exp(-12\lambda) \leq 1 \times e^{-12\times 2.2} \approx 3 \times 10^{-12} \tag{5.1}$$

However, note that the algorithm requires the accuracy of the empirical covariance estimates to be greater than $\frac{\tau}{2}$ which in turn leads to a huge number of samples required to obtain this accuracy.

Therefore, the τ value has to be increased to render experiments comutationally feasible. Goel uses $\tau = 0.2$ and states that *"The choice of parameters is arbitrary and we observed similar performance up to tuning* τ*"*[8], where the choice of parameters refers to the experimental structure of the Ising model. We assume that by *"tuning* τ*"* she means that τ has to be adjusted manually so that the algorithm works well.

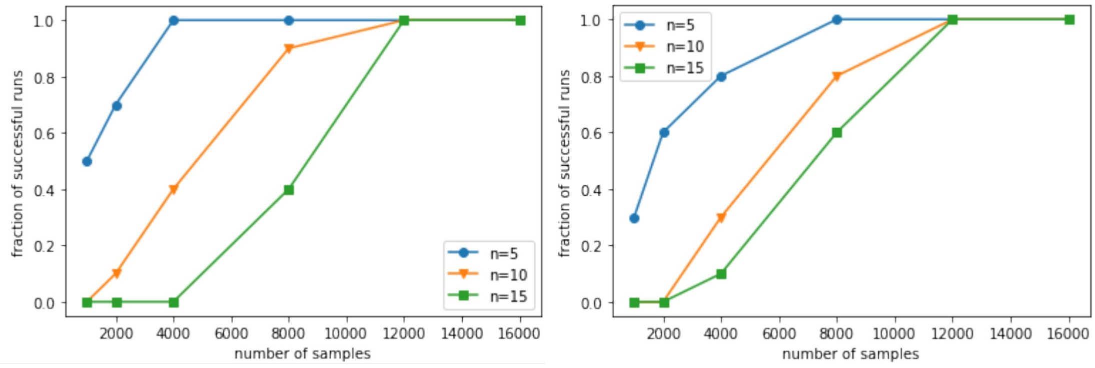
However, this tuning process requires knowledge of the underlying neighbourhood structure and in our observations τ turned out to be quite sensitive to the structure of the network used.

Hence, the tuning has to be quite specific to the graph structure to be learned, posing the question of how much additional knowledge about the graph has to be available to make the algorithm successful in practice.

For our experiments this created the additional challenge of finding appropriate τ values. We tuned the τ value by varying it against the number of samples and observing what neighbourhoods the algorithm returned. We noted that, a consistently too small neighbourhood suggested a too large τ because it prevented some neighbours from being added. On the other hand, a too large neighbourhood set suggests a too small number of samples (or a too small τ) as the accuracy was not sufficient to exclude falsely included vertices. For the general random graph it turned out to be infeasible to tune the τ sufficiently, which led us to adopt the more structured graphs similar to the ones proposed by Goel.

5.2 Reconstruction of Goel's results

5.2.1 External Fields of magnitude ≤ 0.2



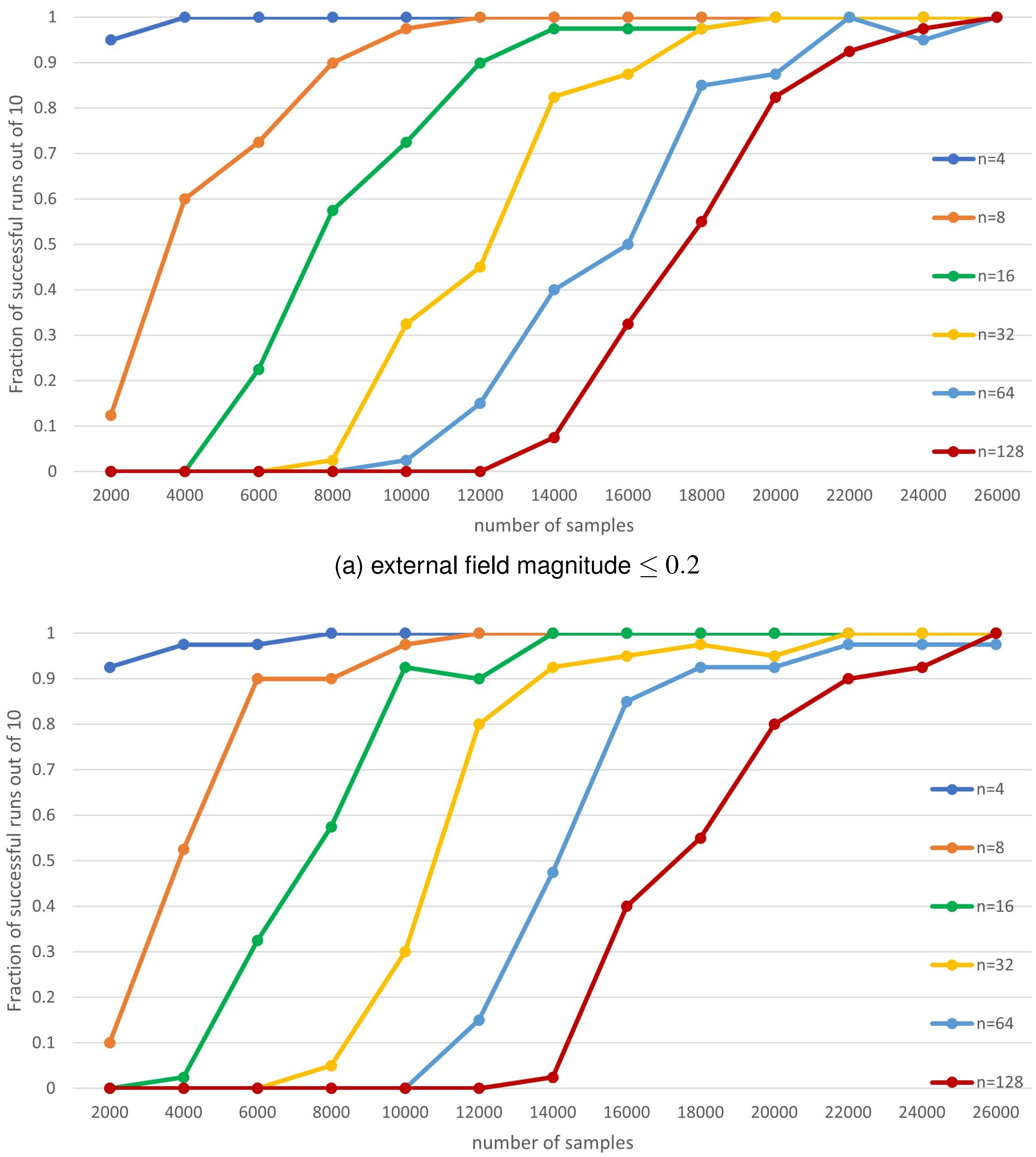
(a) external fields ± 0.2

(b) external fields ± 0.4

Figure 5.2: Goel's experimental data figures from [8]

We compare our generated results to Goel's figures. Note that Goel presents data for vertices of size $n \in \{5, 10, 15\}$ while we chose to generate graphs with $n \in \{4, 8, 16, 32, 128\}$ because this allows us to analyse what happens on doubling of n.

In general, observe that the number of samples required to reconstruct a full graph increases with the number of vertices in the graph, as expected.



(a) external field magnitude ≤ 0.2

(b) external field magnitude ≤ 0.4

Figure 5.3: Fraction of full successful graph recoveries vs. number of samples used, based on our experiments

Comparing the figures for $n = 4$ (fig. 5.3a) and $n = 5$ (Goel's data, fig. 5.2a), dark blue line, the data's general shape seems to be comparable. For $n = 5$, the fraction of successful runs starts at 0.5 and reaches 1 at with 4×10^3 samples. For $n = 4$, our measurements begin with 2×10^3 samples with a success fraction of 0.95 increasing to 1 on the next step (4×10^3 samples). The discrepancy at for the first measurements can be explained by the slightly different size of the two networks, but overall both measurements show reliable recovery for a similar number of samples.

Similarly, for the orange line compare $n = 8$ with $n = 10$ (Goel) we observe a slightly lower success fraction for $n = 10$ at the beginning of the measurements but both approach a success fraction of 1 for around 12×10^3 samples.

Finally, for the green lines our graphs where of size $n = 16$ while Goel presents data for $n = 15$. Here, the shapes show similar behaviour at for smaller sample sizes while for $n = 15$ a success fraction of 1 is reached for 12×10^3 samples while our measurements approach 1 very gradually between 14×10^3 and 20×10^3. This is probably due to random variations in the graphs and samples.

The shapes of Goel's graphs and ours are generally similar showing a vaguely sigmoidal shape, however more experiments would be needed to make any reliable statements about the similarity of the shapes.

Overall, our experiments seem to agree with Goel's data showing similar sample sizes required to achieve a high success rate at predicting the two-hop neighbourhoods.

Now consider the data for larger number of vertices as well (yellow, light blue and red lines). The number of samples required to recover graph structures of increasing size seems to be increasing linearly while the number of observed vertices doubles on each step. This is interesting because other measures of the Ising model (such as for example the partition function) become much harder to compute as the number of vertices increases. However, it agrees well with the relationship Goel gives in theorem 1 where M is proportional to the logarithm of n.

Furthermore, the number of samples is surprisingly small. Note for example that to reliably recover the two-hop neighbourhood of a graph with $n = m = 32$ nodes it seems to be sufficient to use approximately 2×10^4 while the whole sample space of the graph is of size $2^{32+32} \approx 10^{19}$.

Speculating, why this might be the case, we note that the graphs used here are very regular and have only very few edges which decreases the complexity of the problem. Additionally, we see that most of the 10^{19} configurations will have very small probabilities of actually occurring. This is because the Boltzmann weight favours configurations with more aligned spins (see 2.1). However, there are far more configurations of with disordered spin alignments than ordered ones which are assigned small probabilities potentially making the effective sample space smaller.

5.2.2 Comparison to External Fields of magnitude ≤ 0.4

The figures for external fields of magnitude ≤ 0.4 both in Goel's data (fig. 5.2b) and our reconstruction (fig. 5.3b) are very similar to the data for fields of magnitude ≤ 0.2 particularly when looking at the points where a fraction of 1 is reached.

As pointed out earlier, we collected data with external fields in intervals $[-0.2, 0.2]$ and $[-0.4, 0.4]$ which means in the experiments for 0.4 external fields of magnitude ≤ 0.2 can still occur. This makes the experiments less distinctive. However, we do not observe a large difference in Goel's results either.

The plots for the external fields ≤ 0.4 can be described as slightly steeper than the plot for ≤ 0.2 seeing that and success fraction ≥ 0.8 is reached already with approx 2×10^3 less samples than for in the ≤ 0.2 case (e.g. $n = 16$ (green) $\rightarrow 10 \times 10^3$ samples vs. 12×10^3 samples).

This is surprising because generally we would expect the algorithm to perform worse with more variable external fields because they exert more influence on the spin configurations compared to the constant influence of the edges with we want to detect. However, to explore whether this observation is just an artifact from the limited number of experiments we ran or potentially originates from the sampling method, it would be necessary to conduct more experiments. These experiments could take more graphs into account and also use even more extreme external field values to establish whether the trend continues.

5.3 Degree 3 Exploration and Beyond

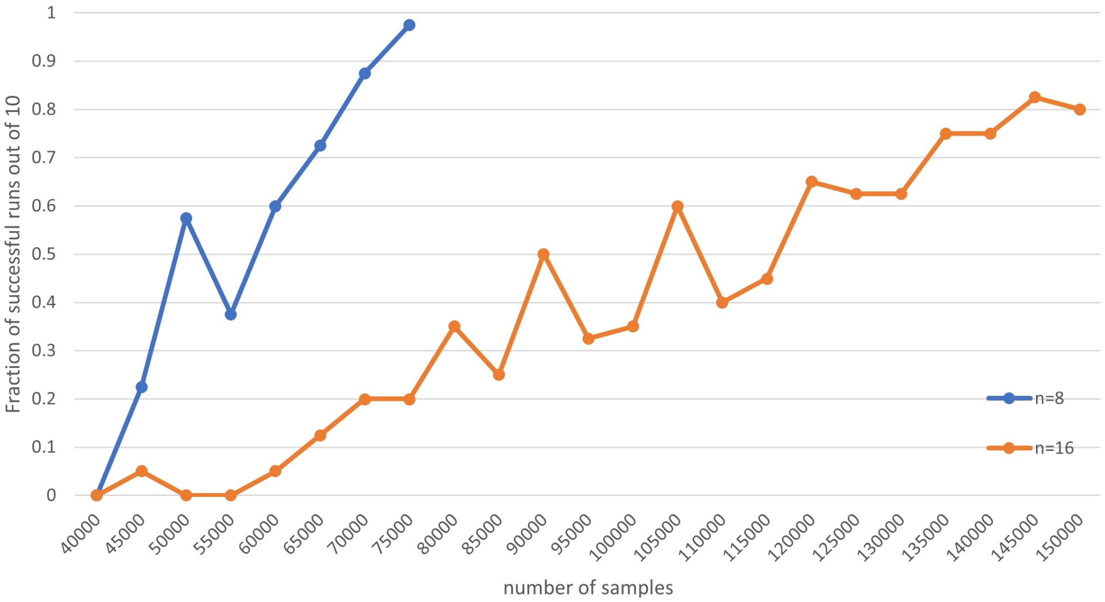


Figure 5.4: Fraction of full successful graph recoveries vs. number of samples used on a graph of degree 3 ($\tau = 0.001$)

We also experimented with graphs of degree three (structure shown in figure 4.2). This was done to see how the τ value has to be tuned for a different structure. We first observed that with $\tau = 0.02$ the returned two-hop neighbourhood sets were too small (smaller than the expected four nodes). This probably occurs because with the inclusion of every further node v, the observed covariance between the rest of the nodes and u is reduced (due to the added conditioning on v). So with more nodes needed to be included in the prospective neighbourhood set, $\tau = 0.02$ turns out to be a too high cut-off for the covariance. After some tries we found $\tau = 0.001$ to work reasonably well for small graphs (as can be seen in fig. 5.4). Again, we averaged our measurements over four different external filed configurations randomly drawn from $[-0.2, 0.2]$.
However, note that now, even for a graph of size $n = 8$, far more samples are required to achieve graph recoveries (e.g. approx. 7×10^4 samples for recovery in more than 8 cases vs. 8×10^3 previously). For $n = 16$ we needed to use even larger samples to recover the structure of the graphs with some success. We ended our measurements at 1.5×10^5 samples because the time to compute this number of samples became prohibitively long.

This experiment shows the link between smaller τ values and the larger number of samples required to successfully recover two-hop neighbourhoods. This can be explained by the accuracy requirement of the algorithm. It works well if the empirical covariance is closer than $\frac{\tau}{2}$ to the theoretical covariance because the τ value is used as a threshold to prune the prospective neighbourhood set. However, if the τ value is smaller, the number of samples has to be much larger to achieve this accuracy.

Finally, we also looked at some graphs with degree 4 extending the previously presented regular structure to a two-hop neighbourhood of size 6. However, we could not find a τ value which captured the structure of the graph within reasonable sample sizes.

Chapter 6

Conclusion

6.1 Summary

In this project we examined Surbhi Goel's algorithm for learning the two-hop neighbourhood of a vertex in a bipartite Ising model based on samples of spin values obtained from one partition of the nodes. We explained how a bipartite Ising model works and how samples from it can be generated using a Swendsen-Wang sampler. Then, we stepped through the structural details of the learning algorithm and explained how Goel proves its correctness.

In our experimental section we showed how we implemented an Ising model and obtained samples from it. We conducted experiments to establish some confidence in the correctness of the sampler.

For our experiments on the learning algorithm, we implemented Goel's algorithm with its auxiliary functions (e.g. the empirical covariance estimate) and tested it on different graph structures. We analysed the algorithm more systematically by reconstructing some of Goel's own experiments. These experiments were then extended to larger graphs which allowed us the observe the logarithmic relationship between the sample size and the number of observed nodes n as predicted by Goel.

We established that the τ threshold input to the algorithm is crucial to the applicability of the algorithm. Setting the threshold to the lower bound on the covariance between two-hop neighbours (as Goel does in her proof of correctness) results in a lower bound on the sample size which was unattainable for our experiments. Smaller τ values require more samples as the τ threshold is proportional to the accuracy of the estimated covariance from the samples. Hence the practical τ value has to be manually increased. However, if the τ value becomes too large it cuts off parts of the two-hop neighbourhoods of the vertices. Therefore, the τ value has to be adjusted to fit for a certain graph structure.

This entails that for a graph with unknown bipartite structure (up to the λ and α bounds on the interactions) the τ value would have to be set to the theoretical lower bound making the structure discovery only feasible under huge sample sizes. On the other hand, the algorithm with tuned τ can be successful if some additional information about the structure is known and the structure is relatively regular.

6.2 Future Work

We would also have liked to explore the characteristics of our sampler further, establishing how many samples are required to reach a mixed state. This could be done by conducting more experiments measuring characteristics of the produced distributions. Another avenue worth exploring could be more theoretical considerations as to when a mixed state is reached.

On the theoretical side of the learning algorithm, it would be interesting to explore other methods of predicting τ values based on structural characteristics of the graph. For example, one could consider fixed-degree graphs or otherwise characterise graphs by the size of their two-hop neighbourhoods.

To widen the experimental understanding, the measurements could be extended by varying the weights on the edges in the graph, the external fields or the ratio of observed to latent nodes. This could give evidence for the some of the other dependencies (namely δ and τ) of the sample size expression given by Goel.

Bibliography

[1] Guy Bresler. Efficiently learning Ising models on arbitrary graphs, *arXiv preprint arXiv: 1411.6156v2*, 2014

[2] Guy Bresler, Frederic Koehler, and Ankur Moitra. Learning restricted Boltzmann machines via influence maximization *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019). Association for Computing Machinery, New York, NY, USA, Pages 828–839*, 2019

[3] Guy Bresler, David Gamarnik, and Devavrat Shah. Structure learning of antiferromagnetic Ising models *Advances in Neural Information Processing Systems 27*, 2014

[4] T. M. Cover and Joy A. Thomas. Elements of Information Theory *Second edition. Hoboken, N.J: Wiley-Interscience*, 2006

[5] J. Duchi. CS229 Supplemental Lecture notes Hoeffding's inequality *Lecture Notes CS229: Machine Learning, Standford University* http://cs229.stanford.edu/extra-notes/hoeffding.pdf, accessed 06.04.2022

[6] Asja Fischer, Christian Igel. Training restricted Boltzmann machines: An introduction, *Pattern Recognition, Volume 47, Issue 1, Pages 25-39*, 2014

[7] Surbhi Goel. Learning Restricted Boltzmann Machines with Arbitrary External Fields, *arXiv preprint arXiv:1906.06595*, 2019

[8] Surbhi Goel. Learning ising and potts models with latent variables, *International Conference on Artificial Intelligence and Statistics, Pages 3557-3566*, 2020

[9] Surbhi Goel, Adam Klivans, Frederic Koehler. From Boltzmann Machines to Neural Networks and Back Again, *arXiv preprint arXiv: 2007.12815v1*, 2020

[10] Lars-Petter Granan, MD, PhD. The Ising Model Applied on Chronification of Pain, *Pain Medicine, Volume 17, Issue 1, January 2016, Pages 5–9*, 2016

[11] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. *Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), Pages 11–15*, 2008

[12] C.R. Harris, K.J. Millman, S.J. van der Walt et al., Array programming with NumPy. *Nature 585, Pages 357–362*, 2020

[13] Linus Hamilton, Frederic Koehler and Ankur Moitra. Information theoretic properties of Markov random fields and their algorithmic applications. *Advances in Neural Information Processing Systems 30*, 2017

[14] G.E. Hinton, A Practical Guide to Training Restricted Boltzmann Machines. In: *Montavon, G., Orr, G.B., Müller, KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg*, 2012

[15] Kerson Huang. *Statistical Mechanics, Second Edition, Wiley, Pages 341-344*, 1987

[16] Mark Jerrum and Alistair Sinclair. Polynomial-Time Approximation Algorithms for the Ising Model *SIAM Journal on Computing, colume 22, 5 pages 1087-1116*, 1993

[17] Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS). IEEE*, 2017

[18] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques, MIT Press, Page 507*, 2009

[19] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques, MIT Press, Page 1145*, 2009

[20] Elchanan Mossel and Allan Sly. Exact thresholds for Ising–Gibbs samplers on general graphs *The Annals of Probability 41.1 Pages 294-328*, 2013

[21] Sejun Park, Yunhun Jang, Andreas Galanis, Jinwoo Shin, Daniel Stefankovic, Eric Vigoda. Rapid Mixing Swendsen-Wang Sampler for Stochastic Partitioned Attractive Models. *CoRR abs/1704.02232*, 2017

[22] J. Percus. Correlation inequalities for ising spin lattices. *Communications in Mathematical Physics, 40(3):283–308*, 1975

[23] G. Van Rossum and F. L. Drake. Python 3 Reference Manual. *Scotts Valley, CA. CreateSpace*, 2009

[24] Feras A. Saad, Cameron E. Freer, Martin C. Rinard, and Vikash K. Mansinghka. Optimal approximate sampling from discrete probability distributions. *Proc. ACM Program. Lang. 4, POPL, Article 36*, 2020

[25] N. P. Santhanam and M. J. Wainwright. Information-Theoretic Limits of Selecting Binary Graphical Models in High Dimensions, in *IEEE Transactions on Information Theory, vol. 58, no. 7, pp. 4117-4134*, 2012

[26] Britton Smith and Sergey Koposov. Lecture 6, *Course: Numerical Recipes (PHYS10090), School of Physics and Astronomy, University of Edinburgh*, 2021

[27] Didier Sornette. Physics and financial economics (1776–2014): puzzles, Ising and agent-based models, *Rep. Prog. Phys.* 77 062001, 2014

Appendix A

Experimental Data

This data was used to generate the graphs seen in the Results and Discussion Chapter.
n: refers to the number of observed nodes used (where $n = m$ as described above)
ext: refers to the range of the absolute value of external fields
the left-most column shows the number of samples used
the tables contain data from four different external field vectors and the avg column giving the average of the four

Table A.1: degree 2 data, zig-zag structure

n=4, ext=0.2	1	2	3	4	avg	n=8, ext=0.2	1	2	3	4	avg
2000	0.9	1	0.9	1	0.95	2000	0.1	0	0.1	0.3	0.125
4000	1	1	1	1	1	4000	0.8	0.5	0.5	0.6	0.6
6000	1	1	1	1	1	6000	0.7	0.7	0.8	0.7	0.725
8000	1	1	1	1	1	8000	1	1	1	0.6	0.9
10000	1	1	1	1	1	10000	0.9	1	1	1	0.975
12000	1	1	1	1	1	12000	1	1	1	1	1
14000	1	1	1	1	1	14000	1	1	1	1	1
16000	1	1	1	1	1	16000	1	1	1	1	1
18000	1	1	1	1	1	18000	1	1	1	1	1
20000	1	1	1	1	1	20000	1	1	1	1	1
22000	1	1	1	1	1	22000	1	1	1	1	1
24000	1	1	1	1	1	24000	1	1	1	1	1
26000	1	1	1	1	1	26000	1	1	1	1	1

n=16, ext=0.2	1	2	3	4	avg	n=32, ext=0.2	1	2	3	4	avg
2000	0	0	0	0	0	2000	0	0	0	0	0
4000	0	0	0	0	0	4000	0	0	0	0	0
6000	0.3	0.2	0.1	0.3	0.225	6000	0	0	0	0	0
8000	0.6	0.7	0.5	0.5	0.575	8000	0	0	0.1	0	0.025
10000	0.7	0.5	0.9	0.8	0.725	10000	0.4	0.4	0.2	0.3	0.325
12000	0.9	0.9	0.9	0.9	0.9	12000	0.7	0.5	0.3	0.3	0.45
14000	1	1	0.9	1	0.975	14000	1	0.9	0.9	0.5	0.825
16000	1	1	0.9	1	0.975	16000	0.9	0.9	0.8	0.9	0.875
18000	0.9	1	1	1	0.975	18000	0.9	1	1	1	0.975
20000	1	1	1	1	1	20000	1	1	1	1	1
22000	1	1	1	1	1	22000	1	1	1	1	1
24000	1	1	1	1	1	24000	1	1	1		1
26000	1	1	1	1	1	26000	1	1	1		1

n=64, ext.0.2	1	2	3	4	avg	n=128,ext=0.2	1	2	3	4	avg
2000	0	0	0	0	0	2000	0	0	0	0	0
4000	0	0	0	0	0	4000	0	0	0	0	0
6000	0	0	0	0	0	6000	0	0	0	0	0
8000	0	0	0	0	0	8000	0	0	0	0	0
10000	0	0.1	0	0	0.025	10000	0	0	0	0	0
12000	0.1	0.1	0.2	0.2	0.15	12000	0	0	0	0	0
14000	0.3	0.5	0.5	0.3	0.4	14000	0.1	0.1	0.1	0	0.075
16000	0.4	0.5	0.9	0.2	0.5	16000	0.3	0.4	0.3	0.3	0.325
18000	0.9	0.9	0.7	0.9	0.85	18000	0.6	0.5	0.7	0.4	0.55
20000	0.9	0.7	0.9	1	0.875	20000	0.6	0.9	0.9	0.9	0.825
22000	1	1	1	1	1	22000	1	1	0.9	0.8	0.925
24000	1	0.9	0.9	1	0.95	24000	1	1	1	0.9	0.975
26000	1	1	1	1	1	26000	1	1	1	1	1

n=4, ext=0.4	1	2	3	4	avg
2000	1	0.8	0.9	1	0.925
4000	1	1	1	0.9	0.975
6000	1	0.9	1	1	0.975
8000	1	1	1	1	1
10000	1	1	1	1	1
12000	1	1	1	1	1
14000	1	1	1	1	1
16000	1	1	1	1	1
18000	1	1	1	1	1
20000	1	1	1	1	1
22000	1	1	1	1	1
24000	1	1	1	1	1
26000	1	1	1	1	1

n=8, ext=0.4	1	2	3	4	avg
2000	0.1	0.1	0	0.2	0.1
4000	0.5	0.7	0.2	0.7	0.525
6000	0.9	1	0.9	0.8	0.9
8000	0.8	0.9	1	0.9	0.9
10000	1	1	1	0.9	0.975
12000	1	1	1	1	1
14000	1	1	1	1	1
16000	1	1	1	1	1
18000	1	1	1	1	1
20000	1	1	1	1	1
22000	1	1	1	1	1
24000	1	1	1	1	1
26000	1	1	1	1	1

n=16, ext=0.4	1	2	3	4	avg
2000	0	0	0	0	0
4000	0.1	0	0	0	0.025
6000	0.4	0.5	0.1	0.3	0.325
8000	0.8	0.6	0.6	0.3	0.575
10000	1	0.9	0.8	1	0.925
12000	1	1	0.8	0.8	0.9
14000	1	1	1	1	1
16000	1	1	1	1	1
18000	1	1	1	1	1
20000	1	1	1	1	1
22000	1	1	1	1	1
24000	1	1	1	1	1
26000	1	1	1	1	1

n=32, ext=0.4	1	2	3	4	avg
2000	0	0	0	0	0
4000	0	0	0	0	0
6000	0	0	0	0	0
8000	0	0	0	0.2	0.05
10000	0.3	0.4	0.3	0.2	0.3
12000	0.9	0.7	0.9	0.7	0.8
14000	0.9	1	1	0.8	0.925
16000	1	1	0.9	0.9	0.95
18000	1	1	1	0.9	0.975
20000	0.9	1	0.9	1	0.95
22000	1	1	1	1	1
24000	1	1	1	1	1
26000	1	1	1	1	1

n=64, ext=0.4	1	2	3	4	avg
2000	0	0	0	0	0
4000	0	0	0	0	0
6000	0	0	0	0	0
8000	0	0	0	0	0
10000	0	0	0	0	0
12000	0.1	0.1	0.1	0.3	0.15
14000	0.6	0.2	0.4	0.7	0.475
16000	0.7	1	0.9	0.8	0.85
18000	1	0.8	0.9	1	0.925
20000	1	0.7	1	1	0.925
22000	1	1	0.9	1	0.975
24000	1	1	0.9	1	0.975
26000	0.9	1	1	1	0.975

n=128, ext=0.4	1	2	3	4	avg
2000	0	0	0	0	0
4000	0	0	0	0	0
6000	0	0	0	0	0
8000	0	0	0	0	0
10000	0	0	0	0	0
12000	0	0	0	0	0
14000	0.1	0	0	0	0.025
16000	0.5	0.7	0.2	0.2	0.4
18000	0.8	0.6	0.4	0.4	0.55
20000	0.9	0.9	0.8	0.6	0.8
22000	0.9	0.9	0.9	0.9	0.9
24000	1	0.9	0.9	0.9	0.925
26000	1	1	1	1	1

Table A.2: degree 3 data

n=8, tau = 0.001	1	2	3	4	avg
5000	0	0	0	0	0
10000	0.3	0.2	0.2	0.2	0.225
15000	0.4	0.7	0.6	0.6	0.575
20000	0.3	0.5	0.3	0.4	0.375
25000	0.4	0.7	0.5	0.8	0.6
30000	1	0.5	0.7	0.7	0.725
35000	0.9	0.6	1	1	0.875
40000	1	1	1	0.9	0.975

n=16, tau=0.001	1	2	3	4	avg
40000	0	0	0	0	0
45000	0.2	0	0	0	0.05
50000	0	0	0	0	0
55000	0	0	0	0	0
60000	0.1	0.1	0	0	0.05
65000	0.1	0.2	0.1	0.1	0.125
70000	0.1	0.4	0.3	0	0.2
75000	0.2	0.1	0.2	0.3	0.2
80000	0.5	0.4	0.2	0.3	0.35
85000	0.2	0.3	0.3	0.2	0.25
90000	0.4	0.5	0.8	0.3	0.5
95000	0.2	0.4	0.4	0.3	0.325
100000	0.6	0.2	0.4	0.2	0.35
105000	0.4	0.6	0.5	0.9	0.6
110000	0.4	0.5	0.3	0.4	0.4
115000	0.6	0.4	0.5	0.3	0.45
120000	0.5	0.6	0.6	0.9	0.65
125000	0.7	0.8	0.4	0.6	0.625
130000	0.6	0.8	0.4	0.7	0.625
135000	0.8	0.8	0.7	0.7	0.75
140000	0.8	0.9	0.7	0.6	0.75
145000	0.7	0.9	0.9	0.8	0.825
150000	0.8	0.9	0.8	0.7	0.8

Appendix B

Hoeffding Bound

Theorem 2 (Hoeffding Bound [19]) *Let $\mathcal{D} = \{X[1], \ldots, X[M]\}$ be a sequence of M independent Bernoulli trials with probability of success p. Let $T_{\mathcal{D}} = \frac{1}{M}\sum_m X[m]$. Then*

$$P_{\mathcal{D}}(T_{\mathcal{D}} > p + \varepsilon) \leq e^{-2M\varepsilon^2} \tag{B.1}$$

$$P_{\mathcal{D}}(T_{\mathcal{D}} < p - \varepsilon) \leq e^{-2M\varepsilon^2} \tag{B.2}$$

Applied to the problem in inequality 3.21:

M: number of samples

$X[i]$: a Bernulli trial with positive outcome if $(X_W = \underline{x}_W)$ in sample i (here the notation clashes, $X[i]$ belongs to theorem 2 while X_W refers to the sample of spin assignments to the vertices in set W)

$T_{\mathcal{D}}$: the empirically estimated probability of $(X_W = \underline{x}_W)$ occuring i.e. $\widehat{Pr}(X_W = \underline{x}_W)$

p: the theoretical probability of $(X_W = \underline{x}_W)$ occuring i.e. $Pr(X_W = \underline{x}_W)$

ε: the difference γ

The Probability of $(T_{\mathcal{D}} > p + \varepsilon)$ or $(T_{\mathcal{D}} < p - \varepsilon)$:

$$Pr\left(|T_{\mathcal{D}} - p| > \varepsilon\right) \leq 2e^{-2M\varepsilon^2} \tag{B.3}$$

Note that Goel uses a bound where $(|T_{\mathcal{D}} - p| \geq \varepsilon)$ which is a slightly stronger statement. However, this seems to be just another formulation of the Hoeffding bound (see for example [5]).