Graphical Models for Mixed Data with Categorical Latent Variables

A thesis submitted in fulfillment of the requirements for the
Bachelor's degree in Mathematics
Bachelor's degree in Data Science and Engineering

Author:

Luis Sierra Muntané

Supervisors:

Piotr Zwiernik (University of Toronto)

Juan José Rué Perna (UPC)

May 2023

UNIVERSITY OF
TORONTO

Acknowledgments

To the many wonderful people at the DoSS of UofT, and especially to my supervisor, P. Zwiernik
thanks to whom I got a glimpse of what life in academia is like,
and for making the cold Canadian winter feel like home.

To my family and closest friends,
for bearing with my numerous peaks and valleys these last 4 years.

Abstract

This thesis aims to provide an overview of probabilistic graphical models in the context of other widely used machine learning methods, and how these methods can be formalised using conditional independence models and algebraic statistics. By comparing the extremely popular Gaussian Mixture Models and Neural Networks as generative graphical models, we are able to propose a graphical model for mixed data (discrete and continuous components) that provides a solid theoretical background and a way to analyse the Gaussian-Bernoulli Restricted Boltzmann Machine. This is used to model variables with a Gaussian conditional distribution, with discrete latent variables. On top of that, this thesis then goes into learning and sampling procedures as well as using techniques from algebraic statistics to further depict the expressibility of the model, making use of independence and mixture models for cumulant semi-algebraic varieties.

Keywords: Graphical Models, Statistical Learning, Gaussian Mixtures, Markov Random Fields, CG Distributions, Restricted Boltzmann Machine, Gaussian-Bernoulli RBM

MSC: 46A32, 62H22, 62R01, 62R07

Resum

Aquesta tesi pretén proporcionar una visió general dels models gràfics probabilístics en el context d'altres mètodes d'aprenentatge automàtic àmpliament utilitzats, i com aquests mètodes es poden formalitzar utilitzant models de independència condicionada i estadística algebraica. A base de comparar les Mixtures Gaussianes amb les Xarxes Neuronals interpretades com a models generatius, proposem un model gràfic per a dades mixtes (variables discretes i contítnues) que proporciona una base teòrica sòlida i una manera d'analitzar la Màquina de Boltzmann Restringida Gaussiana-Bernoulli. Això s'utilitza per modelar variables amb una distribució gaussiana condicionada, amb variables latents discretes. A més a més, aquesta tesi es centra en els procediments d'aprenentatge i mostreig, així com en l'ús de tècniques d'estadística algebraica per a descriure la expressivitat del model, fent servir models d'independència i de mixtura per a les varietats semi-algebraiques dels cumulants.

Mots clau: Models Gràfics, Aprenentatge Estadístic, Mixtures Gaussianes, Camps Aleatoris de Markov, Distribucions CG, Màquina de Boltzmann Restrictiva, RBM Gaussiana-Bernoulli

MSC: 46A32, 62H22, 62R01, 62R07

Resumen

Esta tesis pretende proporcionar una visión general de los modelos gráficos probabilísticos en el contexto de otros métodos de aprendizaje automático ampliamente utilizados, y cómo estos métodos pueden ser formalizados utilizando modelos de independencia condicional y estadística algebraica. Al comparar los populares modelos de Mezclas Gaussianas con las redes neuronales vistas como modelos generativos, proponemos un modelo gráfico para datos mixtos (variables discretas y continuas) que sirve de base teórica sólida a la vez que permite analizar la Máquina de Boltzmann Restringida Gaussiana-Bernoulli. Esto se utiliza para modelar variables con una distribución gaussiana condicionada, con variables latentes discretas. Además, esta tesis se explaya sobre los procedimientos de aprendizaje y muestreo del modelo, así como en el uso de técnicas de estadística algebraica para describir la expresividad del mismo, utilizando modelos de independencia y mezcla para las variedades semi-algebraicas de los cumulantes.

Palabras clave: Modelos Gráficos, Mezclas Gaussianas, Campos Aleatorios de Markov, Distribuciones CG, Máquina de Boltzmann Restrictiva, RBM Gaussiana-Bernoulli

MSC: 46A32, 62H22, 62R01, 62R07

Contents

1 Introduction

In a measure space $(\Omega, \mathcal{A}, \nu)$ where Ω is our sample space, $\mathcal{A}$ is a σ-algebra on Ω, and ν is a dominating measure, which for us will always be the Lebesgue measure, a *statistical model* $\mathcal{M}$ is a collection of measures $\{\mathbb{P}_\theta\}_{\theta\in\Theta}$ dominated by ν, where in this work, the index set Θ will be taken to be a finite dimensional subset of $\mathbb{R}^d$, in other words, a parametric model. We define the density as the Radon-Nikodym derivative of the measure with respect to the dominating measure ν

$$p_\theta(x) = p(x;\theta) = \frac{\mathrm{d}\,\mathbb{P}_\theta}{\mathrm{d}\nu}\,(x)$$

When encountering a statistical learning task, the aim is to find a statistical model that can approximate the true population distribution, whatever that may be. Examples of statistical models may be the Bernoulli distribution for tossing a coin, a Poisson distribution to model the frequency of car accidents etc. Therefore, we need some way to compare statistical models to measure their effectiveness at their chosen task. The canonical way to do so is using the maximum likelihood estimation, especially for parametric models, due some important theoretical results, but as a way to start with the geometric spirit of this report, we may define the maximum likelihood as a sideline of a distance consideration.

Definition 1.1. *Let $\mathcal{M}$ be a set of probability measures. A* divergence *refers to a function $D(\cdot||\cdot) : \mathcal{M}\times\mathcal{M} \to \mathbb{R}\cup\{\infty\}$ satisfying*

$$\begin{aligned} &(i) \quad D(p||q) \geq 0 \\ &(ii) \quad D(p||q) = 0, \text{ only if } p = q \end{aligned}$$

Note that symmetry is not one of the requirements, and so it is often useful to consider the dual divergence *$D^*(p||q) = D(q||p)$.*

Now for P, Q two probability measures on $(\mathcal{X}, \mathcal{A})$, the Kullback-Leibler Divergence*, also referred to as* relative entropy *is given by*

$$D_{KL}(P||Q) = \int_{\mathcal{X}} \log\frac{dP}{dQ}dP, \text{ if } P \ll Q \tag{1.1}$$

which is clearly not symmetric. If $Q \ll P$ then $D_{KL}(P||Q) = \infty$. When p, q are the densities of P, Q respectively, this is equivalent to the more common expression

$$D_{KL}(p||q) = \int_{\mathcal{X}} p\log\frac{p}{q}d\nu$$

Suppose we observe a sample of n data points given by $x_1, \ldots, x_n$, then the *log-likelihood function* can be written as

$$\hat{\theta}_{MLE} = \arg\max_{\theta \in \Theta} \{\ell(\theta)\} = \arg\max_{\theta \in \Theta} \left\{ \sum_{i=1}^{n} \log p_\theta(x_i) \right\}$$

Assuming the true density is $p(\cdot)$ (mind you it need not be in $\mathcal{M}$) then for a fixed $\theta \in \Theta$, assuming we obtain our datapoints x_i from a r.v. $X_i \sim p$ then the *expected* normalized log-likelihood is given by

$$\mathbb{E}\left[\frac{1}{n}\ell(\theta)\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(X_i)\right] = \int_{\mathcal{X}} p(x) \log p_\theta(x) d\nu(x) \tag{1.2}$$

Now by computing the KL Divergence of p and p_θ we find

$$\begin{aligned} D_{KL}(p||p_\theta) &= \int_{\mathcal{X}} p(x) \log \frac{p(x)}{p_\theta(x)} d\nu(x) \\ &= \int_{\mathcal{X}} p(x) \log p(x) d\nu(x) - \int_{\mathcal{X}} p(x) \log p_\theta(x) d\nu(x) \\ &= -\mathbf{H}(p) - \int_{\mathcal{X}} p(x) \log p_\theta(x) d\nu(x) \end{aligned}$$

where we have used $\mathbf{H}$ for Shannon's entropy. Comparing this previous equation with that in 1.2 we can interpret the maximum likelihood as the minimisation of $D_{KL}(p||p_\theta)$ "in the sense of expectation", because by assuming constant ground truth p, it is found by minimising the second term from the previous equation, which is just the log-likelihood. In this way, the maximum likelihood estimation can be seen as having an information geometry interpretation, and in this way we can think of finding the best possible distribution inside a given family of models for extracting the most information from a specific real-world task. The MLE can also already serve as an example of the discrepancy between models for discrete and for continuous variables, since the likelihood function for discrete data is well defined in simple terms, whereas for continuous variable models the likelihood loses probabilistic meaning.

Models for mixed variables, discrete and continuous pose great theoretical difficulties as they fail to combine nicely in many instances like oil and water. This work will be dedicated at exploring how to deal with this problem and what simplifications can be made to give rise to meaningful models for such a combination.

In recent years, the tremendous progress and success in machine learning applications has come about mostly through cruder engineering work, and the statistical knowledge of the models used in ML has lagged behind. Maximum likelihood has been replaced by the mean squared

error (MSE), and empirical risk measures based on a more pragmatic and experimental heuristics (albeit some equivalences have been found). A current concern of modern ML models is their lack of interpretability, and with the new architectures coming out for image classification, large language models and other industry requirements, this statistical understanding is straggling even more.

This phenomenon is probably a consequence of the very difference between statistics/ mathematics and machine learning, since the former is interested in obtaining theoretical results and extracting inferences from data, the latter has a particular goal centered in applications. Such a contrast is akin to that of physics and engineering, where theory and application in a same field can lead to vastly different cultures in a community.

The aim here is that of recovering some of the lost ground from the mathematical statistics perspective, starting with this project, where an exploration into the probabilistic foundations of neural networks and other related models will be given. On top of that, an analysis will be made of a network-like model involving independent Gaussian variables which has been seldom been looked at from an algebraic or statistical perspective, seeing how we can use various tools in graphical models, algebraic statistics and tensors to examine its properties.

Each of these areas individually are huge; graphical models as a way to analyse conditional dependencies between variables are an active area of study with applications in many different domains, and the invention of algebraic statistics as a way to bridge the power and generality of algebraic geometry into statistical challenges has proved to be very fruitful in the past decades. We hope that looking at their intersection in this work can lead to some interesting new ways to look at the problem of data modelling and have future contributions in data science.

To motivate this intersection and analysis, we will start with a look at several well-known models for data analysis and feature extraction which will contain many of the key ideas we will use later on. As such, this work is meant to serve as an overview of how mathematics, statistics and machine learning can fit together, provide some interesting results about why some things work the way they do, and try to leverage their respective strengths for each other's benefit: new model ideas, better understanding of their properties and better results in their subsequent applications.

2 Literature Review

2.1 Gaussian Mixture Models

What can be done when a dataset that could reasonable come from a normal distribution does not fit the model? This question came to Karl Pearson in the 19th century when exploring a dataset on crab dimensions [1]. He observed that the distribution was not symmetric as a normal distribution should be, to a larger extent than what could have happened by random chance. His solution to the problem was creative as well as founded: assume that the crabs came from two different populations, normally distributed, and the observations observed were the result of the sum of two independent normal distributions.

He then fitted the distribution functions using methods that today would seem extremely inefficient, by solving a system of polynomial equations in the first two moments of each distribution (the means and the variances) as explained in [2] which relates this technique to *algebraic statistics*, which will be explored later on. However, the main rationale of assuming a distribution consisting of an additive combination of other distributions was born, and is known as a *mixture distribution*. As such, the idea would be encapsulated by the following equation, where we assume our overall distribution comes from the combination of K separate, independent Gaussians $X_i \sim \mathcal{N}(\mu_i, \Sigma_i)$.

$$p(x) = \sum_{k=1}^{K} \pi_i \phi_k(x), \quad \sum_{k=1}^{K} \pi_k = 1, \quad 0 \leq \pi_k \leq 1 \tag{2.1}$$

Where the ϕ_i are the probability density functions for each variable X_i. In modern notation, the actual object of study is known as a mixture distribution, defined as follows.

Definition 2.1. *A* mixture distribution *consists of a family of probability distributions $\mathcal{F}_{X|T=t}$ with distribution function $F_{X|T=t}(x)$ where t is not a fixed parameter but comes from a family of distributions $\mathcal{F}_T$, in such a way that we have $\mathcal{F}_X = \mathcal{F}_{X|T=t} \wedge \mathcal{F}_T$ the mixture, where $X|T=t$ has distribution function $F_{X|T=t}(x)$.*

Proposition 2.2. *The distribution of $\mathcal{F}_X$ is given by $F_X(x) = \mathbb{E}_T\left[F_{X|T=t}\right]$*

Proof. Directly from the definition, taking

$$F_X(x) = \int_{\mathbb{R}} f_X(y)dy = \int_{\mathbb{R}}\int_{\mathbb{R}} f_{(X,T)}(y,t)dtdy = \int_{\mathbb{R}}\int_{\mathbb{R}} f_T(t) f_{X|T=t}(y,t)dtdy$$

and now using Fubini we get

$$\int_{\mathbb{R}} f_T(t)\left[\int_{\mathbb{R}} f_{X|T=t}(y,t)dy\right]dt = \int_{\mathbb{R}} f_T(t)F_{X|T=t}(x)dt = \mathbb{E}_T[F_{X|T=t}]$$

From this we can actually also obtain the density function as

$$f_X(x) = \int_{\mathbb{R}} f_{(X,T)}(x,t)dt = \int_{\mathbb{R}} f_{X|T=t}(x,t) f_T(t)dt = \mathbb{E}_T[f_{X|T=t}(x,t)]$$

□

Corollary 2.1. *For finite or countable mixtures, we can always write*

$$F(x) = \sum_{i \in I} w_i F_i(x)$$

where w_i are the weights coming from $\mathcal{F}_T$ and $F_i(x)$ are the distributions of the random variables $X_i \in \mathcal{F}_{X|T=t}$. The issue is that, in general, finding the coefficients w_i in inference is no trivial task. Moreover, by linearity, we can also write a similar expression for the density function.

Such an approach has been adopted by the machine learning community in a number of different ways for classification tasks such as Linear or Quadratic Discriminant Analysis, general GMMs and even a K-Means clustering algorithm is a particular variant of a Gaussian Mixture Model.

We can interpret these models in a particular way that will be useful later on, to motivate what we are aiming to do. In short, we may assume a latent categorical distribution corresponding to, for a given observation, the prior probability that it belongs to a given sub-populations in the model $p(h)$, where each h will be one of the sub-populations, and $p(h)$ its relative frequency. Then, given this population, now the observation follows a Gaussian distribution of mean μ_h and covariance matrix Σ_h, that is, a Gaussian distribution with density $p(x|h)$. This conditional structure serves as a nice way to generalize this scheme towards *Bayesian Networks*, as we will see in later chapters.

To fit the parameters $\pi = (\pi_1, \ldots, \pi_K), \mu = \mu_1, \ldots, \mu_K, \Sigma = \{\Sigma_1, \ldots, \Sigma_K\}$, a direct approach would be to look at the likelihood function, which from 2.1 can be calculated, assuming N observations, as:

$$\ln p(\mathbf{x}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \phi_k \left(x_n | \mu_k, \Sigma_k\right) \right\} \tag{2.2}$$

Maximising the function in 2.2 is not as easy as the uni-Gaussian case because even after taking the gradient we find that the summation inside the logarithm becomes a quotient that depends on the parameters in a non-convex way, so maximisation is not an easy task [3]. Other approaches involve iterative procedures using numerical methods [4][5], but another approach to be exploited is that of assuming the population variables are latent and use a method known as *Expectation Maximisation* or EM, which will be the one explored in later chapters, after we provide the appropriate context.

This computational obstacle is probably one of the main reasons why in practical machine learning applications involving Gaussian mixture-like scenarios, simplifications are made in order to facilitate inference and/or training, a common one being that of assuming *isotropy*, i.e. $\Sigma = \sigma \mathrm{Id}$ for all components of the mixture, some of the others having been mentioned earlier.

It is also noteworthy to mention that this flexibility attained by a Gaussian mixture in terms of it being an exponentially decreasing function with respect to the squared distance to the given prototypes (each mean μ_k) has also made them a main character in kernel methods. Without going into too much detail, the Radial Basis Function kernel from [6] aimed at interpolation, uses this function as a proxy for the Gaussian density, namely:

$$K(x, x') = \exp\big(-\gamma \|x - x'\|^2\big)$$

Which is reminiscent of an isotropic Gaussian mixture, where the hyperparameter γ controls the scale (inverse of twice the variance), with means set on all the existing observations, and with a number of components as large as the number of observations, where training is conducted in a completely different way so as to not run into the drawbacks expressed previously.

Another noteworthy point about this scheme is the number of modes of the density function, which is of course closely related to the number of local maxima in the likelihood function. Until recently, this was an unsolved problem, which is unsurprising given the counter-intuitive behaviour these functions have in higher dimensions, that arise due to the fact that specific combinations of Gaussian densities produce many more local maxima than the number of components in the mixture [7].

In fact, there are still some major details that are left to decipher, such as proving that the number of modes is actually finite. The tight bounds shown in [7] were given with this finiteness assumption, following a 2011 AIM conjecture by Bernd Sturmfels, which is something that at least to the writer seemed very surprising.

2.2 Probabilistic PCA

An important related model, again originating from Karl Pearson's genius, is that of Probabilistic PCA, based on Pearson's plane approximation in [8]. PCA is ubiquitous as a dimensionality reduction technique since it works by projecting the data onto a subspace of lower dimensionality, where such a subspace is chosen so as to retain maximal variance. From this direct computational technique we can devise a probabilistic model proposed in [9], which works by expressing the assumption of gaussianity in the original data, represented by the random vector X of dimension D, and letting Z be a latent Gaussian vector of dimension $M << D$. As such, $Z \sim \mathcal{N}_M(0, \mathrm{Id})$ and

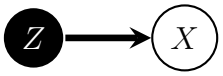
$X|Z = \mathbf{z} \sim \mathcal{N}_D(\mu + WZ, \sigma^2\mathrm{Id})$, where W is a $D \times M$ matrix. Then we may take

$$X = WZ + \mu + \epsilon$$

Yielding a generative version of the PCA procedure, where WZ represents the random variable obtained after the projection onto the lower dimensional subspace. Recall PCA was done over centered data, and so for the general case we include the μ term to account for any possible mean. Note also the factorization of the distribution, this will again come up later as an example of a Bayesian Network, and even for this case, as an example of the Naive Bayes Model in 3.1.4. The matrix W here corresponds to the principal subspace transformation, and ϵ is taken to be zero-mean random Gaussian noise of covariance $\sigma^2\mathrm{Id}$. In this way, to marginalise the observed distribution we could simply use the law of total probability.

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z}$$

And since we are marginalising from a Gaussian, we know X is also Gaussian, such that $X \sim \mathcal{N}_D\left(\mu, WW^T + \sigma^2\mathrm{Id}\right)$. In this vein, we can represent this conditional distribution relationship that appears in Gaussian mixtures and Probabilistic PCA as a directed graph.



The only difference being that for the Gaussian mixture, Z represented a $1, \ldots, K$-valued categorical random variable, whereas in Probabilistic PCA it is an M-dimensional standard multivariate Gaussian. We will see later on how to formalise this graph structure.

Remark 2.1. *The also well-known technique for multivariate data* Factor Analysis *[10] would also be a part of this family, as it is formally almost like Probabilistic PCA only with the difference of the conditional distribution $X|Z = \mathbf{z}$ being Gaussian with a* diagonal *covariance matrix instead of isotropic: $X|Z = \mathbf{z} \sim \mathcal{N}_D(\mu + WZ, \Sigma)$ with diagonal Σ.*

In this sense, we can see how this sort of structure also lends itself to dimensionality reduction, and so the underlying model that one uses can often, as is the case with the graphical models we will later explore, be flexible tools to support many different applications.

2.3 Product of Experts

A third important class of models, which actually came about much more recently than the two previous ones, from purely machine learning contexts are that of Product of Experts models, or

PoE models [11]. The rationale is rather simple, instead of aiming for a model whose distribution is centered around some fixed prototypes with some random variation (as with Gaussian mixtures), we can combine densities multiplicatively, as if they came from M informed or expert opinions each represented by a distribution p_m, and we took a product of their individual certainties.

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{m=1}^{M} p_m(\mathbf{x}|\theta_m), \quad Z(\boldsymbol{\theta}) = \int_{\mathcal{X}} \prod_{m=1}^{M} p_m(\mathbf{x}|\theta_m)\, \mathrm{d}x \tag{2.3}$$

We have taken the densities p_m to be from a parametric family with parameters θ_m since all examples in this dissertation will be of this type, but other approaches involve non-parametric methods too. As such, our parameters are $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$

Of course, since the product of probability density functions is not a density function, the normalisation via Z is necessary, and can sometimes be a nuisance, as we shall see later. The intuitive interpretation of this structure, even though, as of now, it doesn't seem to have a solid statistical background, is that the probability of an observation $\mathbf{x}$ will be high only when all the individual densities $p_m(\mathbf{x})$ are high; instead of having a prototype with noise structure (as with Gaussian mixtures), we have a consensus and veto power structure; if one of the "experts" assigns a low probability to an observation (if $\exists m : p_m(\mathbf{x})$ is small) then the overall probability will also be low. We will see later how to make sense of this.

For inference, if we consider the log-likelihood function and its gradient we can observe an important detail.

$$\begin{aligned} \ell(\boldsymbol{\theta}|x_n) &= \sum_{n=1}^{N} \sum_{m=1}^{M} \log p_m(x_n|\theta_m) - N \log Z(\boldsymbol{\theta}) \\ \implies \nabla_m \ell(\boldsymbol{\theta}|x_n) &= \sum_{n=1}^{N} \nabla_m p_m(x_n) - N \cdot \mathbb{E}_p \left[\nabla_m \log p_m(x)\right] \end{aligned} \tag{2.4}$$

The term highlighted in red is important because it is an expectation with respect to the distribution given by p, and so it required calculating the normalising constant $Z(\boldsymbol{\theta})$ which in many cases can be intractable, as can be seen from its calculation in 2.3, thus posing computational issues.

2.4 Neural Networks Overview

Finally, we will make a quick overview of some of the basic properties of the queen of deep learning: the multilayer perceptron, or feed-forward neural network. DNNs have been successfully used to model a plethora of different distributions, but despite their numerous successes, their

interpretability and statistical soundness are not well-understood. In this work, we will just look at some of their basic features in order to motivate the work done in later chapters.

Of course, there is a whole zoo of different so-called Neural Networks [12], all with different architectures aimed at adapting to a particular task, but we will be focusing on one of the oldest and most flexible architectures, still widely used as of 2023, which served as a starting point for many of the other more sophisticated types we see in current machine learning applications.

Let's define the architecture used for this type of model. For every layer, there shall be n neurons, described by the parameters $(W^{[i]}, \mathbf{b}^{[i]})$ known as *weights* $W^{[i]} \in \mathcal{M}_n(\mathbb{R})$ and *biases* $\mathbf{b}^{[i]} \in \mathbb{R}^n$ respectively. For an input vector $\mathbf{x} \in \mathbb{R}^d$ we have a first layer $(W^{[1]}, \mathbf{b}^{[1]})$ where $W^{[1]} \in \mathcal{M}_{d\times n}$ for the sizes to match. The feed-forward equation to pass from an input $\mathbf{x} \in \mathbb{R}^n$ coming from the output of the previous layer, to the next is given by

$$\mathbf{a} = S\left(W^{[i]}\mathbf{x} + \mathbf{b}^{[i]}\right) \tag{2.5}$$

Where S denotes an *activation function*, which is the main strength of a neural network since it introduces non-linearity, increasing the expressive power. An illustration of a NN is displayed in 1. One of the oldest activation functions used is the *sigmoid* function, known classically to statisticians as the inverse of the *logit* or *log-odds* function used in regression. The sigmoid function is given by

$$S(x) = \frac{1}{1 + e^{-x}}$$

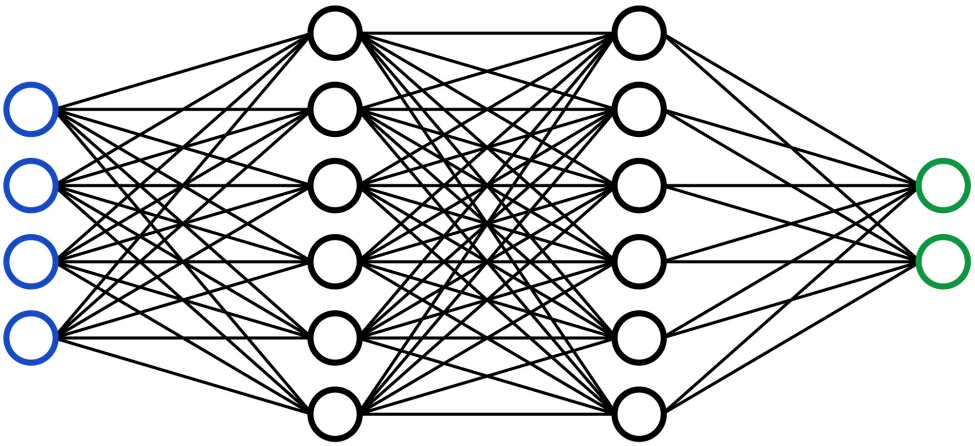


Figure 1: Diagram of a picture representation of a neural network with 4 inputs, 2 hidden layers of 6 neurons each, and 2 output variables

In fact, some authors attribute the importance of this function to a practical reason. For instance, [13] states that the importance of the sigmoid is that it was used to obtain a proof of the *Universal Approximation Theorem* in 1989. What this means is that a neural network, under certain conditions, is able to approximate a whole class of functions arbitrarily well. That being said, there is a theoretical reason for why the sigmoid is the natural choice for an activation function, which we will show in the section on Restricted Boltzmann Machines.

In terms of the approximation capabilities of neural networks, we may state the following theorem:

Theorem 1. *(Universal Approximation, [14]) For an input in $\mathbb{R}^n$ and output in $\mathbb{R}^m$, a single feed-forward neural network with at least one hidden layer and a sigmoid activation is able to uniformly approximate any continuous function $f : X \subset \mathbb{R}^n \longrightarrow \mathbb{R}^m$ with compact domain.*

This theorem is of large theoretical interest since it basically provides us with a guarantee that a neural network can model any absolutely continuous distribution in a compact domain, but the sigmoid requirement is unnecessary for such a result. It was shown in [15] that a larger class of activation functions do the job, and the state of the art is to use *Rectified Linear Units*, or ReLU, given by $S(x) = \max\{0, x\}$ for activations in image classification architectures [16].

In [13], we also have some new results and refinements of the universal approximation theorem, but the key takeaway is that neural networks can approximate a large class of functions, and so they are a very strong modelling tool provided the conditions for them to be trained effectively are met.

Training them is notoriously hard due to the large parameter space and non-convexity, but machine learning practitioners have found training procedures involving different versions of stochastic gradient descent and regularisation to provide very good practical solutions. The notion that convexity is overrated [17] has also been widespread for a number of years in many machine learning tasks, and so all state of the art models have non-convex loss functions.

On the other hand, approximating is not the same as learning, and the usefulness of neural networks is limited by their lack of interpretability, since the meaning of the parameters $W, \mathbf{b}$ is quite tenuous. In fact, many statisticians consider neural networks to be, from a philosophical standpoint, an example of a non-parametric model, since their approximation capabilities are more relevant than the interpretability of their parameters.

Instead of seeing neural networks as having an "architecture" and rules for computation, we will look at them as a probabilistic graphical model that serves as a way to explain their nature probabilistically.

3 Graphical Models

In this chapter we will see a natural way to model random vectors with dependence relationships between their components, which generalise the factorisation of the previously seen models of Gaussian mixtures 2.1, Probabilistic PCA 2.2, factor analysis, and beyond, using graph theory.

Before starting with any definitions, a short note on conditional independence in terms of the covariance.

Definition 3.1. *The* covariance *between two random variables X, Y is given by*

$$\mathrm{Cov}(X, Y) = \mathbb{E}\left[(X - \mathbb{E}\left[X\right])\left(Y - \mathbb{E}\left[Y\right]\right)\right]$$

For a set of random variables $X_1, \ldots, X_n$ we may define the covariance matrix *as $\Sigma = \mathrm{Cov}(\mathbf{X}) = \{\mathrm{Cov}(X_i, X_j)\}_{ij}$. When any two variables are independent, then their covariance is equal to 0 since in particular they will be mean independent, but the reverse implication is only true whenever $\mathbf{X}$ follows a multivariate normal distribution. We may also define the* precision matrix *as $K = \Sigma^{-1}$, which exists whenever the covariance matrix is full rank. By direct computation using a diagonal change of basis we obtain that zeros in the i, j entry of the precision matrix correspond to the variables X_i, X_j being conditionally uncorrelated given all other variables.*

$$K_{ij} = 0 \iff \rho_{X_i, X_j | X_{\setminus \{i,j\}}} = 0$$

For this reason, this matrix is quite relevant in the handling of graphical models.

3.1 Bayesian Networks

Graphical Models have been used in a variety of different contexts for modelling, most prominently in statistical physics and phylogenetics, and as such we can find some of the terminology spilling in from these fields. In essence, their aim is to model the dependency relationships between several attributes, modelled in turn by random variables.

For a graph $\mathcal{G} = (V, E)$ with vertices V and edges E comprising of pairs of vertices, we can distinguish between *directed* and *undirected* graphs, according to whether the elements of E are ordered or unordered pairs, respectively. Accordingly, for a graphical model, we take V to be a set of random variables. When the graph is directed, the graphical model is referred to as a *Bayesian Network* while when the graph is undirected, the standard term in the literature is a *Markov Random Field*.

We will start off with the case where $\mathcal{G}$ is a directed graph. In order for the dependence relations

to be defined correctly, this underlying graph $\mathcal{G}$ must not contain any cycles, as will shall see in 3.2, that is, it has to be a *Directed Acyclic Graph*. But first, we shall see how the graph is used to encode the dependencies between the variables.

When the graph contains an edge $(a, b) \in E$, we say a is a *parent* of b and they are often referred to as the *cause* and the *effect*, respectively. This nomenclature points towards the usefulness of these models in the study of causality and general causal inference, (see for example [18]). Let $v \in V$, we will denote as $\mathbf{pa}(v)$, the set of *parents* of v, given by the vertices w such that $(w, v) \in E$. The set $\mathbf{de}(v)$ of *descendants* of v is given by the vertices w for which there exists a directed path $v \to w$ in $\mathcal{G}$, and we may also define the set $\mathbf{nd}(v)$ of *non-descendants* as $V \setminus (\{v\} \cup \mathbf{de}(v))$. The *local Markov property* of conditional independence states that

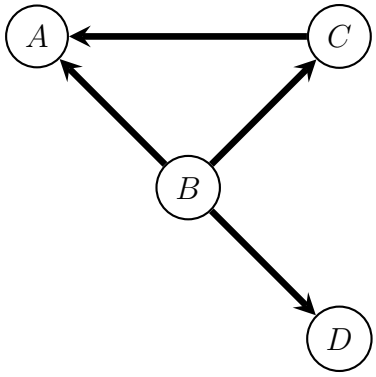
$$X_v \perp\!\!\!\perp X_{\mathbf{nd}(v)\setminus\mathbf{pa}(v)} \mid X_{\mathbf{pa}(v)}, \quad v \in V \tag{3.1}$$

That is, given the parent nodes of a vertex v, the variable X_v is independent from all others. From it, we obtain the key property used to define graphical models, that of their factorisation. Using this notion for conditional independence, we may recursively factorise the joint density of the random vector $\mathbf{X} = (X_1, \ldots, X_n)$, where $V = \{X_i\}_{i=1}^n$, as:

$$p(\mathbf{x}) = \prod_{i=1}^{n} p\left(x_i | \mathbf{pa}(x_i)\right) \tag{3.2}$$

Where it is understood that $p\left(x\right) := p\left(X = x\right)$. This is often known as the *chain rule* of Bayesian networks, following the chain rule of conditional probability and Markov chains. This can be intuitively seen as a design in which the effects are being factorised in terms of their causes, in a way where cause and effect are related sequentially, and two variables are independent if there is no path between them, or if we know the value of a variable found inside the path connecting them. To illustrate this point is the following illustrative example of a Bayesian Network.

Example 3.1.1. *Consider the graphical model associated to the following graph over the variables* A, B, C, D.

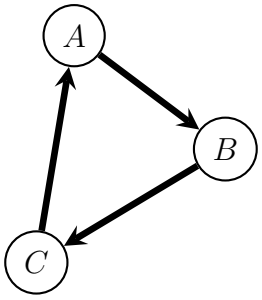


Where we would have the factorization $p(a,b,c,d) = p(a|b,c)p(c|b)p(d|b)p(b)$

Now, we are ready to see why in order for Bayesian Networks to be well defined, the underlying graph must be acyclic.

Proposition 3.2. *The underlying graph in a Bayesian network must be a DAG.*

Proof. Let A, B, C be three binary variables with dependency graph given by the following directed graph, and where the value of each parent completely determines the value of each child, meaning $\mathbb{P}(A = x|B = x) = 1$, as depicted in the probability tables below.



	a_0	a_1
b_0	1	0
b_1	0	1

	b_0	b_1
c_0	1	0
c_1	0	1

	c_0	c_1
a_0	1	0
a_1	0	1

In this way, if we calculate the sum of all events (which should be 1) using the Bayesian Network factorisation from 3.2 yields

$$\sum_{\mathcal{A}\times\mathcal{B}\times\mathcal{C}} p(A=a,B=b,C=c) = \sum_{\mathcal{A}\times\mathcal{B}\times\mathcal{C}} p(B|A)p(A|C)p(C|B) = 2 > 1$$

□

The problem that arises is due to the fact that it is not possible to guarantee that the distribution is correctly normalised in this case, whereas when we have a DAG, we can factorise the distribution starting from a root node (which exists due to there being no cycles) and can easily see how the factorisation produces a distribution with total measure 1.

This definition is deceptively simple, and we can find, even with small examples, already some interesting phenomena.

Example 3.1.3. *Consider three r.v.s X, Y, Z in a v-structure: $X \to Y \leftarrow Z$. Then we have that $p(x, y, z) = p(x)p(z)p(y|x, z)$. We can observe that $X \perp\!\!\!\perp Z$ since their marginal distributions factor completely, but when we condition on Y we obtain*

$$p(x, z|y) = \frac{p(x)p(z)p(y|x, z)}{p(y)}$$

And so despite the marginal independence of X and Z, they are not *independent when we condition on Y! In some sense, we can think of two parents being independent but when we condition on a child, whatever is not explained by one of the parents must be explained by the other one, so the independence is broken.*

One of the most well-known general classifiers is that of the *Naive Bayes* model, introduced as a simple and fast to train classification scheme that is often taught as a way introduce more complex classifiers later on.

Example 3.1.4. *(Naive Bayes Classifier) Consider a collection of so called "features", that is, some vector of random variables $X = (X_1, X_2, \ldots, X_m)$ and a variable Z representing to which of k classes an observation may belong. In order to predict for an observation of data $\mathbf{x} = (x_1, \ldots, x_m)$ to which class z it belongs, Bayes' rule is applied in order to compute the posterior probability of the class:*

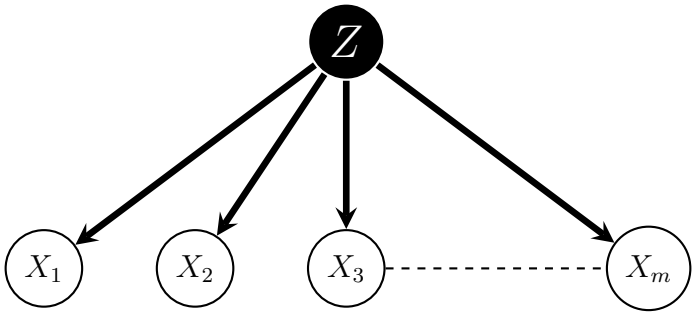
$$\mathbb{P}(z|\mathbf{x}) = \frac{\mathbb{P}(z)\,\mathbb{P}(\mathbf{x}|z)}{\mathbb{P}(\mathbf{x})} = \frac{\mathbb{P}(z)\prod_{i=1}^{m}\mathbb{P}(x_i|z)}{\mathbb{P}(\mathbf{x})}$$

Where now we would estimate $\mathbb{P}(x_i|z)$ according to the observed frequencies from a given dataset with respect to the class of the datapoint, and using Bayes' decision rule we would assign the label with the highest posterior probability:

$$z = \arg\max_{z} \{\mathbb{P}(z)\prod_{i=1}^{n}\mathbb{P}(x_i|z)\}$$

And thus the inference of the parameters of the model, which would consist on the class frequencies, is performed in a simple and direct manner. This ease of calculation comes at a cost of

course: such an independence assumption is a rather strong constraint on the data, and so this "naivety" of the model which comes in assuming the conditional independence of features (variables) of the observations given the class Z, is a rather optimistic way of modelling data. As such, this could be represented as the following Bayesian Network.



The node for the variable Z is coloured in black to indicate that this is a latent variable, meaning that it is not observed, as with the population in a Gaussian Mixture. In this way, Naive Bayes can be thought of as a generative graphical model with the simple dependence structure $X_i \perp\!\!\!\perp X_j | Z$.

Compare such a simple model to the task of inference or learning in a general Bayesian Network with several observable and latent variables. When learning from complete data, one can go about maximising the log-likelihood directly with respect to the parameters $\boldsymbol{\theta}$, or with some other improved numerical approximations, but in any case the approach is straightforward for a sample of size N.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta} \in \Theta} \left\{ \sum_{i=1}^{N} \ln p(\mathbf{x_i}|\boldsymbol{\theta}) \right\} \tag{3.3}$$

Where the expression for $p(\mathbf{x_i}|\boldsymbol{\theta})$ will be computable whenever the Bayesian Network structure is known. We will not be looking at validating the structure of a Bayesian network, or any graphical model in this analysis, as we will take $\mathcal{G}$ as known.

Note that, so far, no assumptions are made on the distribution of each of the individual variables in the graphical model, and this general structure is agnostic to such a choice. We will impose concrete distributions later on, but for now, assume each variable in the network has a known distribution. In fact, for the factor analysis model, a key assumption is that the variables are jointly distributed as a multivariable normal, and the graphical model is a Bayesian Network.

A classical maximum likelihood scheme would work to fit the parameters once we assume said distribution, but for models with latent variables, despite the simplicity of the Naive Bayes model

where there was a single latent for which we had known prior distributions, in general we run into the problem that we do not know the full expression of the likelihood function, and thus it is not possible to maximise the expression in 3.3 directly. As such, the main approach to avoid this pitfall is that of using the *Expectation Maximisation* Algorithm, introduced in the late 1970s, which follows the same ideas for the MLE as in the introduction, but averaging out the latent variables.

Definition 3.5. *(Expectation-Maximization) The EM algorithm is an iterative procedure used to find the maximum likelihood estimation when there are latent variables involved. Let $V = (Z, X)$ where Z are latent and X are observables and define, for a current set of parameters $\theta^{(t)}$ the function $Q(\theta, \theta^{(t)})$ as the partial likelihood defined below.*

E-Step:

$$\begin{aligned} Q(\theta, \theta^{(t)}) &:= \mathbb{E}_{Z \sim p(\cdot|X,\theta^{(t)})} \left[\ln p(X, Z|\theta)\right] \\ &= \sum_{Z} \ln p(X, Z|\theta) p(Z|X, \theta^{(t)}) \end{aligned}$$

M-Step:

$$\theta^{(t+1)} := \arg\max_{\theta} \{Q(\theta, \theta^{(t)})\}$$

These two steps are performed iteratively until a given precision is satisfied. For the M-Step*, assuming some regularity for the distribution, typical gradient ascent schemes are often used [19].*

Due to the procedure being monotone increasing in the estimated parameters, it produces a consistent estimator (convergent in probability) whenever the likelihood has a single local maximum, which for practical purposes is quite a strong assumption. As always, when working with a non-convex likelihood function, using multiple starting points is the practical solution to avoid missing the global maximum, without going into concerns about the degree of non-convexity [17].

3.2 Markov Random Fields

On the other hand, we have the case where the graph $\mathcal{G}$ is undirected, which can be thought of as having both directional edges for each pair of connected variables. As before, we take $\mathcal{G} = (V, E)$ with V a set of random variables, but this change in E causes the graphical models to be very different, given that the training/inference procedures and whole description change substantially. They often take the name in the literature of *Markov Random Fields* and many of their uses came

from statistical mechanics, as a way to model the energy of particles set in a lattice structure, where the energy in a node depends on the energy in the neighbouring nodes.

As before, we may define conditional independence via graph separation, only now the factorisation we defined in 3.2 doesn't make sense. This time, the factorisation of the joint distribution is given in terms of *potential functions*, again borrowing nomenclature from physics. In fact, the way in which we need to preserve the notion of locality (which previously was just path-connectedness in a DAG) in the factorisation in order for the conditional independence to be well defined, will be done by factorising the joint density into functions of the correct subsets of V.

Consider two nodes $X_i, X_j \in V$ where $\{X_i, X_j\} \notin E$. We should require that these two be independent given the rest of the variables, since they will be separated in $\mathcal{G}$

$$p(x_i, x_j|\mathbf{x}_{\setminus\{i,j\}}) = p(x_i|\mathbf{x}_{\setminus\{i,j\}})p(x_j|\mathbf{x}_{\setminus\{i,j\}})$$

This restriction is known as the *pairwise Markov property*, which we will name (P), but a stronger condition, the *local Markov property* or (L) of the graph is that of requiring that only the variables that "surround" a node x_i be required for conditional independence with any other node, that is, if we define $\mathrm{bd}(v)$ to be nodes that are connected to $v \in V$, then (L) states:

$$v \perp\!\!\!\perp V\setminus\{\mathrm{bd}(v) \cup \{v\}\} \mid \mathrm{bd}(v)$$

This only refers to a single variable, and so the strongest condition, known as the *global Markov property* or (G) states that:

$$X_A \perp\!\!\!\perp X_B \,|X_C \quad \text{whenever} \quad C \text{ separates A and B in } \mathcal{G} \tag{3.4}$$

In many texts such as [20], the previous condition is also written abusing the notation by using $A, B, C \subset V$ and saying:

$$A \perp\!\!\!\perp B \,|C \quad \text{whenever} \quad C \text{ separates A and B in } \mathcal{G}$$

Remark 3.1. *In the previous equations for condition (G), we only mentioned the reverse implication instead of stating it as an if and only if condition. The reason for this is that the implication from left to right states that the graph $\mathcal{G}$ is* faithful *to the distribution, and would be concern when trying to find a graph that could model a given distribution. In practice this is something that would be evaluated according to conditions from the data. This technical detail will not be relevant to our study since we take the graph $\mathcal{G}$ as given, but it is nevertheless an important condition for the general study of graphical models.*

Proposition 3.1. *It is easy to see that for any graph G and probability measure P on $\mathcal{X}$ we have that* $(G) \implies (L) \implies (P)$

Proof. Pretty straightforward, just notice that $\mathrm{bd}(v)$ is a separating set between v and $V \backslash \{\mathrm{bd}(v) \cup \{v\}\}$. Then, for the second implication, by non-adjacency we have that $w \in V \backslash \{\mathrm{bd}(v) \cup \{v\}\}$ and we may observe that $\mathrm{bd}(v) \cup ((V \backslash \{\mathrm{bd}(v) \cup \{v\}\}) \backslash \{w\}) = V \backslash \{v, w\}$. □

The reverse implications require quite a bit more work, and an extra assumption, in the form of the property that for disjoint sets A, B, C, D we have that

Proposition 3.2. *Suppose the density over the graph $\mathcal{G}$ is strictly positive, then*

$$A \perp\!\!\!\perp B \mid (C \cup D) \text{ and } A \perp\!\!\!\perp C \mid (B \cup D) \implies A \perp\!\!\!\perp (B \cup C) \mid D \tag{3.5}$$

This condition makes intuitive sense but is not for free; some degenerate graphs may not satisfy it, and it is equivalent on some condition on the dual graph of G that we will not go into. Fortunately, any probability measure with continuous and positive density will satisfy this, and this is already a more than strong enough assumption [20].

Proof. The two conditional independences imply the density factorisations

$$p_{A\cup B|C\cup D}(\mathbf{x}_A, \mathbf{x}_B|\mathbf{x}_C, \mathbf{x}_D) = p_{A|C\cup D}(\mathbf{x}_A|\mathbf{x}_C, \mathbf{x}_D) p_{B|C\cup D}(\mathbf{x}_B|\mathbf{x}_C, \mathbf{x}_D) \tag{3.6}$$

$$p_{A\cup C|B\cup D}(\mathbf{x}_A, \mathbf{x}_C|\mathbf{x}_B, \mathbf{x}_D) = p_{A|B\cup D}(\mathbf{x}_A|\mathbf{x}_B, \mathbf{x}_D) p_{C|B\cup D}(\mathbf{x}_C|\mathbf{x}_B, \mathbf{x}_D) \tag{3.7}$$

Multiplying 3.6 by $p_{C\cup D}(\mathbf{x}_C, \mathbf{x}_D)$ and 3.7 by $p_{B\cup D}(\mathbf{x}_B, \mathbf{x}_D)$ we obtain the full density, and so we can equate the results on the right hand side, and dividing by the (positive) density $p_{B\cup C\cup D}(\mathbf{x}_B, \mathbf{x}_C, \mathbf{x}_D)$ we obtain

$$p_{A|C\cup D}(\mathbf{x}_A|\mathbf{x}_C, \mathbf{x}_D) = p_{A|B\cup D}(\mathbf{x}_A|\mathbf{x}_B, \mathbf{x}_D)$$

Since the left-hand-side does not depend on $\mathbf{x}_B$, we have that $p_{A|C\cup D}(\mathbf{x}_A|\mathbf{x}_C, \mathbf{x}_D) = p_{A|D}(\mathbf{x}_A|\mathbf{x}_D)$ and so if we take the product we had for 3.6 and condition on $\mathbf{x}_D$ we get:

$$p_{A\cup B\cup C|D}(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C|\mathbf{x}_D) = p_{A|D}(\mathbf{x}_A|\mathbf{x}_D) p_{B\cup C|D}(\mathbf{x}_B, \mathbf{x}_C|\mathbf{x}_D)$$

Which implies the independence result we wanted. □

Theorem 2. *(Pearl and Paz) If a probability distribution on $\mathcal{X}$ is such that 3.5 holds for all disjoint sets A, B, C, D, then*

$$(G) \iff (L) \iff (P)$$

Proof. It is sufficient to show that $(P) \implies (G)$. To do so, we will show that for sets A, B, S where S separates the non-empty sets A, B we have that condition (G) holds. By reverse strong induction on the size of S, let $n = |S|$ and consider $n = |V| - 2$, then, since both A and B consist of a single vertex, (P) is trivially enough for the conditional independence. Now assume (G) holds for all sets of more than n elements and take $n < |V| - 2$. Take $V = A \sqcup B \sqcup S$ and so either of A, B has more than one element. Without loss of generality, assume A has more than one element. Take $\alpha \in A$ and note $S \cup \{\alpha\}$ separates $A \backslash \{\alpha\}$ and B, as well as $S \cup (A \backslash \{\alpha\})$ separates $\{\alpha\}$ and B. Thus, from the induction hypothesis we have the two conditions

$$A \backslash \{\alpha\} \perp\!\!\!\perp B \mid S \cup \{\alpha\} \quad \text{and} \quad \{\alpha\} \perp\!\!\!\perp B \mid S \cup (A \backslash \{\alpha\}) \implies A \perp\!\!\!\perp B \mid S$$

where the implication follows from 3.5. We can of course have that $A \cup B \cup S \subset V$, in which case we can simply take $\alpha \in V \backslash (A \cup B \cup S)$ and then observe that $S \cup \{\alpha\}$ separates A, B, and either $S \cup A$ separates $\{\alpha\}, B$ or $S \cup B$ separates $\{\alpha\}, A$ from which we can apply 3.5 to yield our desired result. □

These so-termed *Markov conditions* will force us to factorise $p(\mathbf{x})$ into a product of the distribution over the maximal *cliques* of $\mathcal{G}$, which are the maximal complete induced subgraphs of $\mathcal{G}$. In other words, if we let $\mathcal{C}(\mathcal{G})$ be the set of maximal cliques in $\mathcal{G}$, then we can factorise the joint density as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}(\mathcal{G})} \psi_C(\mathbf{x}_C), \quad Z = \int_{\mathcal{X}} \prod_{C \in \mathcal{C}(\mathcal{G})} \psi_C(\mathbf{x}_C) \tag{3.8}$$

Where Z is the normalising function with a similar role as that in the product of experts model we had in 2.3, and the functions ψ_C are known as the *potential functions*, which are non-negative in order for the density to be non-negative. The fact that this factorisation is consistent with the conditional independence condition in 3.4 is provided by the *Hammersley-Clifford* theorem. We will need a small combinatorial lemma to do so.

Lemma 3.1. *(Möbius Inversion) Let V be a finite set and let Ψ, Φ be functions defined on $\mathcal{P}(V)$ taking values in an abelian group. Then, the following two statements are equivalent:*

(1) for all $a \subseteq V$: $\Psi(a) = \sum_{b \subseteq a} \Phi(b)$

(2) for all $a \subseteq V$: $\Phi(a) = \sum_{b \subseteq a} (-1)^{|a \backslash b|} \Psi(b)$

Proof. Both directions are proven analogously, using a kind of sophisticated inclusion-exclusion

idea. For (2) $\implies$ (1), take

$$\begin{aligned}\sum_{b\subseteq a}\Phi(b) &= \sum_{b\subseteq a}\sum_{c\subseteq b}(-1)^{|b\setminus c|}\Psi(c)\\ &= \sum_{c\subseteq a}\Psi(c)\left\{\sum_{b:c\subseteq b\subseteq a}(-1)^{|b\setminus c|}\right\}\\ &= \sum_{c\subseteq a}\Psi(c)\left\{\sum_{h\subseteq a\setminus c}(-1)^{|h|}\right\}\end{aligned}$$

the last sum is equal to zero except for the case where $a\setminus c=\varnothing$, since for a non-empty set there is the same amount of sets of even degree as of odd degree. □

Now we are ready to prove our factorisation theorem.

Theorem 3. *(Hammersley-Clifford) A probability distribution P with positive and continuous density p with respect to a product measure satisfies (P) with respect to a graph $\mathcal{G}$ if and only if it factorizes according to $\mathcal{G}$.*

Proof. We of course only need to prove the reverse implication, since the factorisation already implies conditional independence and thus the (P) property. To do so, first rewrite 3.8 (thanks to the positivity of the density) as

$$\log p(x) = \sum_{a\in\mathcal{C}(\mathcal{G})}\phi_a(x_a) \tag{3.9}$$

where $\phi_a(x_a)=\log\psi_a(x_a)$. We are only assuming P is pairwise Markov. Let us fix an element $x^*\in\mathcal{X}$ of the sample space and define for all $a\subseteq V$ the functions $H_a(x):=\log p(x_a,x_{a^C}{}^*)$, and also

$$\phi_a(x)=\sum_{b\subseteq a}(-1)^{|a\setminus b|}H_b(x)$$

we write these as a functions of x but of course each one only depends on x_a. Now, by lemma 3.1 applied to ϕ_a we have that

$$\log p(x)=H_V(x)=\sum_{a\subseteq V}\phi_a(x)$$

We now need to show that $\phi_a(x)\equiv 0$ when a does not induce a clique in V. To do so, let us take $\alpha,\beta\in a$ such that $\alpha\not\sim\beta$ and consider the following decomposition of ϕ_a using inclusion-exclusion:

$$\phi_a(x)=\sum_{b\subseteq c}(-1)^{|c\setminus b}(H_b-H_{b\cup\{\alpha\}}-H_{b\cup\{\beta\}}+H_{b\cup\{\alpha,\beta\}}) \tag{3.10}$$

Now, let $d = V \backslash \{\alpha, \beta\}$, we have that, using (P) and 2:

$$\begin{aligned}
H_{b\cup\{\alpha,\beta\}} - H_{b\cup\{\alpha\}} &= \log \frac{p(x_b, x_\alpha, x_\beta, x_{d\backslash b}{}^*)}{p(x_b, x_\alpha, x_\beta{}^*, x_{d\backslash b}{}^*)} \\
&= \log \frac{p(x_\alpha|x_b, x_\beta, x_{d\backslash b}{}^*)p(x_b, x_\beta, x_{d\backslash b}{}^*)}{p(x_\alpha|x_b, x_\beta{}^*, x_{d\backslash b}{}^*)p(x_b, x_\beta{}^*, x_{d\backslash b}{}^*)} = \log \frac{p(x_\alpha|x_b, x_{d\backslash b}{}^*)p(x_b, x_\beta, x_{d\backslash b}{}^*)}{p(x_\alpha|x_b, x_{d\backslash b}{}^*)p(x_b, x_\beta{}^*, x_{d\backslash b}{}^*)} \\
&= \log \frac{p(x_\alpha{}^*|x_b, x_{d\backslash b}{}^*)p(x_b, x_\beta, x_{d\backslash b}{}^*)}{p(x_\alpha{}^*|x_b, x_{d\backslash b}{}^*)p(x_b, x_\beta{}^*, x_{d\backslash b}{}^*)} \\
&= \log \frac{p(x_b, x_\alpha{}^*, x_\beta, x_{d\backslash b}{}^*)}{p(x_b, x_\alpha{}^*, x_\beta{}^*, x_{d\backslash b}{}^*)} \\
&= H_{b\cup\{\beta\}} - H_b
\end{aligned}$$

and so the expression in 3.10 is equal to 0, which is what we wanted, and only the complete subgraphs survive. □

It is noteworthy to see how much simpler everything was when we were working with DAGs (as with Bayesian Networks), as the factorisation of the density was trivial. That being said, Markov random fields are able to represent much richer structures and thus are more expressible as models, so this extra work is not in vain.

The non-negativity of the potential functions $\psi_C(\mathbf{x}_C) \geq 0$ we had in the density 3.8 makes it convenient to write them as

$$\psi_C(\mathbf{x}_C) = \exp\{-E(\mathbf{x}_C)\}$$

where E is known as the *energy* function, and now these potential functions represent the density for a *Gibbs distribution*, the name again due to its use in statistical mechanics. Intuitively, it models the fact that energy tends to flow from areas of high concentration to areas of low concentration, and so the probability of encountering states with high energy is exponentially decreasing. For us, this is key to being able to write 3.8 in a convenient and parametric way, so that we can do inference over a specific set of parameters, instead of the unmanageable set of positive measurable functions, which would require more sophisticated tools in variational inference [3].

In this way, we can leverage the exponential in the potential function to rewrite 3.8 as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C\in\mathcal{C}(\mathcal{G})} \exp\{-E(\mathbf{x}_C)\} = \frac{1}{Z} \exp\left\{ - \sum_{C\in\mathcal{C}(\mathcal{G})} E(\mathbf{x}_C) \right\} \tag{3.11}$$

Note that, crucially for training, this is an example of a PoE model 2.3, and now we have

rewritten all of the clique dependence as something that is part of the definition of the energy function. This rewriting will be very useful in the cases that we will examine as the cliques will be easy to describe and the energy will have a nice probabilistic interpretation as the canonical parameters of an exponential family. In particular, for binary variables where the expression for the energy is linear in the parameters as combinations of variables and product of pairs of variables, this model is known as a *Boltzmann machine*, and is used extensively in physics [21].

3.3 Restricted Boltzmann Machines

We will know examine a deeply studied kind of Boltzmann Machine, (and so MRF) with a strong connection to neural networks and which serve as a starting point for the model defined in the last chapter. They may also be found under the name *Harmonium*, due to the shape of the underlying graph, which was their original name in [22].

Definition 3.1. *(Restricted Boltzmann Machine) An RBM [23][22], is an undirected graphical model where $V = \{X_1, \dots, X_n, Y_1, \dots, Y_m\}$ are all binary variables, where the visible units are $\mathbf{X}$ and the hidden units are $\mathbf{Y}$, set in a bipartite graph structure, with energy function given by*

$$E(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = -\sum_{i \in N} \theta_i x_i - \sum_{j \in M} \theta_j y_j - \sum_{\{i,j\}} \theta_{ij} x_i y_j \tag{3.12}$$

Where $\boldsymbol{\theta}$ are the parameters of the model. Then, the probability of a given state is given by

$$\mathbb{P}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(-E(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})) \tag{3.13}$$

with Z a normalising constant to ensure the distribution is well defined, as seen in previous sections.

The RBM probability model with n visible and m hidden units is the set of probability distributions of the form 3.13, for all possible choices of $\boldsymbol{\theta}$. We denote this set by $RBM_{m,n}$.

Summing over the hidden units to we can easily see the expression for the marginal probability of a visible configuration, which yields another, more familiar product distribution (independent marginal distributions). To ease the notation, we will use $W := \{\theta_{ij}\}_{ij}$ the mixed product parameters, $b_i := \theta_i$ for the parameters of the visible variables and $c_j := \theta_j$ the parameters of the latent variables. As such, 3.13 becomes

$$\mathbb{P}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\{-\mathbf{b}^\mathbf{T}\mathbf{x} - \mathbf{c}^\mathbf{T}\mathbf{y} - \mathbf{y}^\mathbf{T} W \mathbf{x}\}$$

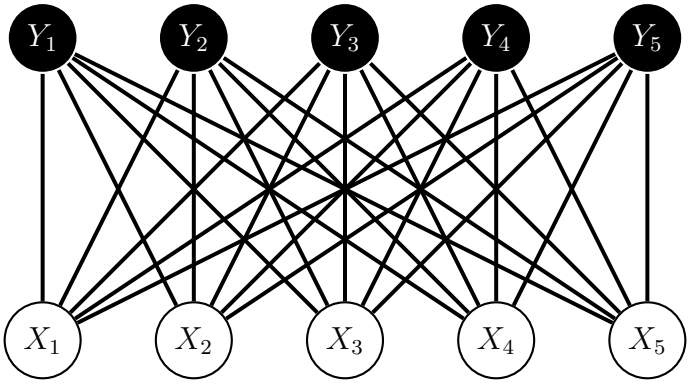


Figure 2: The graphical representation of $RBM_{5,5}$ with 5 latent and 5 observable variables.

To find the marginal distribution of the visible variables, we sum over all possible state of the hidden variables:

$$p(\mathbf{x};\boldsymbol{\theta}) = \sum_{\mathbf{y}\in\{0,1\}^m} \frac{1}{Z(\boldsymbol{\theta})} \exp\{-\mathbf{b}^\mathbf{T}\mathbf{x} - \mathbf{c}^\mathbf{T}\mathbf{y} - \mathbf{y}^\mathbf{T}W\mathbf{x}\} \tag{3.14}$$

$$= \frac{1}{Z(\boldsymbol{\theta})} \exp\{-\mathbf{b}^\mathbf{T}\mathbf{x}\} \sum_{\mathbf{y}\in\{0,1\}^m} \exp\{-\mathbf{c}^\mathbf{T}\mathbf{y} - \mathbf{y}^\mathbf{T}W\mathbf{x}\} \tag{3.15}$$

$$= \frac{1}{Z(\boldsymbol{\theta})} \exp\{-\mathbf{b}^\mathbf{T}\mathbf{x}\} \prod_{j\in M} (1 + \exp(c_j + W_j\mathbf{x})) \tag{3.16}$$

And so we get the product distribution we were looking for. The fact that this is a product distribution means that we can use 2.4 for inference, but in reality, most practitioners in practice use an even better procedure that avoids having to calculate the normalising constant, which is computationally expensive. In fact, Hinton and Welling devised a method known as Contrastive Divergence, which works by introducing bias into the MLE but accelerating the computations substantially in the style of MCMC. For more details check [24]. All of this constitutes the appeal of RBMs as a model able to break the curse of dimensionality, since despite having the structure of a mixture model, it can also be written as a PoE model which is much faster to train. For a more in-depth analysis on RBMs, check out [25].

Now that we have the joint and marginal distribution (and the two marginals of visible and latent are symmetric so the expression is almost the same), we can find the conditional distribution of a visible variable with respect to the latent variables using 3.13 and 3.14 to obtain:

$$p(X_i = 1|\mathbf{y}) = \frac{p(X_i = 1, \mathbf{y})}{p(\mathbf{y})} = \sigma\left(\sum_{j=1}^{m} W_{ij}y_j + b_i\right) \tag{3.17}$$

Where σ is the sigmoid function $\sigma(x) = 1/\left(1 + e^{-x}\right)$. This is precisely the expression we had in 2.5 for a single computation of a neuron of a neural network! This is primary interest of RBMs: they are the generative model from which feed-forward networks come about, and stacking many of them together is equivalent to adding layers to form a deep neural network. This also points to the fact of why the sigmoid function is the natural choice for an activation function in a neural network, since it arises directly from the conditional distribution of one neuron given the previous layer.

Remark 3.2. *As we can see from 3.17, the observable variables X_i are conditionally independent given the latent variables, and vice versa, that is, $\forall i \neq j$*

$$X_i \perp\!\!\!\perp X_j \,|\, \mathbf{Y}$$
$$Y_i \perp\!\!\!\perp Y_j \,|\, \mathbf{X}$$

We could have also expected these conditional independence assumptions from the separability of the variables in each layer of the RBM. This in turn provides us with proof of the conditional independence of each perceptron as a probabilistic model with respect to other perceptrons in their same layer, given the states of the previous layer.

That being said, the current state of the art in this area is *not* to use the sigmoid function, as exposed in [16], where they used the ReLU activation for ImageNet classification tasks. The intuition as to why ReLU performed better in this case is often cited as being the sigmoid is vulnerable to vanishing gradients, a machine learning phenomenon by which calculating many derivatives consecutive in backpropagation and multiplying the results, which are values smaller than 1, makes the parameter updating a very small value, but this does not seem to be completely understood.

3.4 CG Distribution for Mixed Data

Now that we have looked at the definition of graphical models and seen a key example, albeit for binary variables, we will provide a brief account of the models used for mixed data, where some variables are discrete, and some variables are continuous. To that effect, in 1989, Lauritzen and Wermuth[26] devised the *Conditional Gaussian* distribution (CG), as a way of bridging the gap between the two most prevalent areas of theoretical graphical model theory: contingency

tables, and multivariate normal distributions. The CG distribution aimed to model distributions where both options appeared together, and this explains both the particular choices of conditional distributions for each kind of variable.

Following the notation in [20], let $V = \Delta \cup \Gamma$ where Δ refers to the variables of discrete type, and Γ to the variables of continuous type. In this way, a realisation of the random vector V would be given by (h, y), where h would be the value of the realisation for the $|\Delta|$ discrete variables, which we will call H, and $y \in \mathbb{R}^{|\Gamma|}$ for the continuous realization.

Definition 3.1. *We say that the* Conditional Gaussian *distribution (CG) if*

$$p(h) = \mathbb{P}(H = h) > 0, \quad Y|H = h \sim \mathcal{N}_{|\Gamma|}(\mu(h), \Sigma(h)) \tag{3.18}$$

Which is equivalent to a joint density given by

$$f(h, y) = \exp\left\{ g(h) + G(h)^T y - \frac{1}{2} y^T K(h) y \right\} \tag{3.19}$$

for functions $g : \Delta \to \mathbb{R}$ and $G : \Delta \to \mathbb{R}^{|\Gamma|}$

We have written 3.19 as an equality and so we can think of g, G as including the normalising constant. Here if we name $|\Delta| = m$, $|\Gamma| = n$ then we have $g : \Delta \to \mathbb{R}$ and $G : \Delta \to \mathbb{R}^n$. Note that, despite the fact that this is an example of an exponential family, this is not currently written down in exponential form since it does not have the required structure, but we can see the resemblance to the density of the RBM in 3.13 with a quadratic term in y instead of just the inner product with the parameters. To see the density in its exponential form, which is rather involved, see pages 171-175 in [20].

The functions g, G can be found explicitly and depend on the latent states only, and K refers to the precision matrix ($K = \Sigma^{-1}$) which is written as depending on h, but in many cases and applications, the simplification $K(h) = K$, termed "homogeneous" by Lauritzen, is made.

Due to the conditional independences, this model can be thought of as a directed graphical model

Later, we will be considering sub-models of this general model, by breaking down the observables into conditionally independent Gaussians and by changing the conditional independence of the latent variables. To obtain the relationship between the two sets of parameters, we may take

$$\Sigma(h) = K(h), \quad \mu(h) = K(h)^{-1} g(h) \tag{3.20}$$

By using 3.20 and looking at the density in 3.19 as:

$$\begin{aligned} f(h,y) &= \exp\left\{ g(h) + G(h)^T y - \frac{1}{2} y^T K(h) y \right\} \\ &= \exp\left\{ g(h) + \left(\frac{1}{2} G(h)^T K(h)^{-1} G(h) - \frac{1}{2} G(h)^T K(h)^{-1} G(h) \right) + y^T G(h) - \frac{1}{2} y^T K(h) y \right\} \\ &= \exp\left\{ \left(g(h) + \frac{1}{2} G(h)^T K(h)^{-1} G(h) \right) - \frac{1}{2} (y - \mu(h))^T K(i) (y - \mu(h)) \right\} \\ &= \exp\left\{ g^*(h) + \frac{1}{2} (y - \mu(h))^T K(h) (y - \mu(h)) \right\} \end{aligned}$$

Where we have used that K is symmetric. Now let $z = y - \mu(h)$ and, assuming $K(h)$ is positive definite (we only need to assume full rank since $\Sigma \succcurlyeq 0$, because it is a covariance matrix), we can integrate over z to marginalize the discrete variables and obtain

$$\begin{aligned} p(h) &= \int_{z \in \mathbb{R}^n} f(h,y) dz = e^{g^*(h)} \int_{z \in \mathbb{R}^n} e^{-\frac{1}{2} z^T K(h) z} dz = e^{g^*(h)} (2\pi)^{n/2} (\det K)^{-1/2} \\ &= (2\pi)^{n/2} (\det K)^{-1/2} \exp\left\{ g(h) + \frac{1}{2} G(h)^T K(h)^{-1} G(h) \right\} \end{aligned} \tag{3.21}$$

This relation can be inverted solving for g in 3.21 to find the explicit value of g in terms of the

moment parameters

$$g(h) = \log p(h) - \frac{n}{2}\log 2\pi + \frac{1}{2}\log\det K(h) - \frac{1}{2}G(i)^T K(h)^{-1} G(h) \tag{3.22}$$

$$= \log p(h) - \frac{n}{2}\log 2\pi - \frac{1}{2}\log\det \Sigma(h)^{-1} - \frac{1}{2}\mu(h)^T \Sigma^{-1}\mu(h) \tag{3.23}$$

In this way we have found explicit ways to change the parameters of the distribution, from what Lauritzen calls the *canonical characteristics* (g, G, K) (due to them actually being equal to the canonical parameters when written in exponential form) and the *moment characteristics* (p, μ, K^{-1})

If we specify further the nature of the categorical variables, and looking to transition into the model family we will look at in later chapters, we can find a more concrete expression for $p(h)$, such as, for instance, when we have m binary variables $H_1, \ldots, H_m$ (as happens when we have a directed graphical model to represent the conditional independence of our variables). In this case we can compute the probability of a particular state h as:

$$p(h) = \Pr(H_i = h_i : i \in [m]) = \prod_{i\in[m]} {b_i}^{h_i}(1 - b_i)^{1-h_i} \tag{3.24}$$

Where it is understood that $H_i \sim \text{Bern}(b_i)$ so we have transformed a categorical variable on m states into m independent binary random variables.

This binarisation is very apt for calculations and can be thought of as being reminiscent to a sort of one-hot encoding to display the variables in a more convenient manner, but allowing more than one state to be active at once, so there are actually 2^m possible configurations. For us, this configuration is a way to reparametrise our latent space that will be useful when defining Gaussian-Bernoulli RBM models.

4 Algebraic Statistics

In the previous chapter, we defined the notion of a graphical model and looked at the two main types in terms of the type of edges of the underlying graph. In this section, we will see how these models can be defined in an algebraic way, and what can we gain from such an interpretation. At a high level, the gist is that of using tools from algebraic geometry to attain results in statistics [27]. This idea came from the seminal paper from Sturmfels and Diaconis [28], which defined the concept of a Markov basis and marked the birth of algebraic statistics as a separate field by using commutative algebra to sample from a conditional distribution.

4.1 Graphical Models

We will start by providing some definitions and then move on to how this algebraic setting translates the graphical models we defined in the previous chapter. The point of all of this is that we can express graphical models algebraically in terms of the independence ideals that are generated by the conditional independence equations. Most of the notation and style is thanks to [27].

Definition 4.1. *A* k-simplex *is a* k*-dimensional polytope given by the convex hull of* $k+1$ *(independent) vertices* $u_0, \dots, u_k$ *as in*

$$C = \left\{ \lambda_0 u_0 + \cdots + \lambda_k u_k \;\middle|\; \sum_{i=0}^{k} \lambda_i = 1,\ \lambda_i \geq 0 \right\} \tag{4.1}$$

Since our context is that of probability theory, all of our vertices will be given by the canonical basis, as in

$$\Delta_k = \left\{ \lambda \in \mathbb{R}^{k+1} : \lambda_0 + \cdots + \lambda_k,\ \lambda_i \geq 0 \right\} \tag{4.2}$$

And here we don't need to specify the use of barymetric coordinates.

These simplices are useful because they are a geometrical representation of all the probability distributions on a discrete set of $k+1$ elements. We may think of a distribution as a point $p_{i_1 \dots i_k} \in \Delta_{k-1}$. In the case of the binary distribution, n random variables take joint states in $\{0,1\}^n$, the probability of each state $a_{i_1 \dots i_n} \in \{0,1\}^n$ can be described as $p_{i_1 \dots i_n} = \Pr(H = a) = \Pr(\{H_j = i_j : j = 1, \dots n\})$. Therefore, we have that $(p_{i_1 \dots i_n})_{i_1 \dots i_n},\ i_j \in \{0,1\}$ is a $2 \times \overset{n)}{\cdots} \times 2$ tensor storing the probability of each state, and the elements lie in the simplex $\Delta_{2^n - 1}$.

Observation 4.2. *Following from the previous definition of* $p_{i_1 \dots i_n}$ *it is standard to define the*

marginal probabilities with the notation

$$p_{i+} := \sum_j p_{ij} = \Pr(X_1 = i) \tag{4.3}$$

Then, we can think that two variables are independent when the joint probability tensors factor as $p_{ij} = p_{i+}p_{+j}$*, for all* $i \in [r_1],\ j \in [r_2]$*.*

What does independence mean algebraically? The discrete case will be simpler to illustrate. Take $X = (X_1, \ldots, X_m)$ to be discrete r.v.s and let $[r_j]$ be the set of values taken by X_j, that is, $\operatorname{Im} X_j = [r_j]$, and so X takes values in $\mathcal{R} = \prod_{j=1}^m [r_j]$. Then, we shall see that conditional independence constraints correspond to a system of quadratic polynomial equations.

Proposition 4.3. *The conditional independence statement* $X_A \perp\!\!\!\perp X_B | X_C$ *holds if and only if*

$$p_{i_A,i_B,i_C,+} \cdot p_{j_A,j_B,i_C,+} - p_{i_A,j_B,i_C,+} \cdot p_{j_A,i_B,i_C,+} = 0 \tag{4.4}$$

for all $i_A, j_A \in \mathcal{R}_A,\ i_B, j_B \in \mathcal{R}_B,\ i_C \in \mathcal{R}_C$*.*

Observation 4.4. *Two random variables are marginally independent,* $X_1 \perp\!\!\!\perp X_2$*, if and only if the* $r_1 \times r_2$ *matrix* (p_{ij}) *has rank 1.*

Definition 4.5. *The* conditional independence ideal $I_{A \perp\!\!\!\perp B|C}$ *is generated by all quadratic polynomials in 4.4, and for two random variables* X_1, X_2*, it would be represented as*

$$I_{X_1 \perp\!\!\!\perp X_2} = \langle p_{i_1 i_2} p_{j_1 j_2} - p_{i_1 j_2} p_{j_1 i_2} \mid i_1, j_1 \in [r_1], i_2, j_2 \in [r_2] \rangle \tag{4.5}$$

To provide a taste of the usefulness of this theory, [29] provides us with this result using fully algebraic methods.

Proposition 4.6. *The Restricted Boltzmann Machine with 2 hidden units and 3 observables* $RBM_{2,3}$ *is described in the interior of the* Δ_7 *simplex by the union of the algebraic sets*

$$\{p_{000}p_{011} \geq p_{001}p_{010}, \quad p_{100}p_{111} \geq p_{101}p_{110}\}$$

And the other 5 *permutations of indices. These sets represent the conditional independence between variables, which we know from 3.1.*

Note that the model is over-parameterised. In this case, the model has 11 parameters but only lives in Δ_7.

The main result of [29] is concerning the tree model with a 3-state latent categorical variable and 3 binary leaves:

Theorem 4. *$\mathcal{M}_{3,3} = \overline{RBM}_{2,3}$ with $\mathcal{M}_{3,3} = RBM_{2,3}$ on the interior of the simplex.*

The proof of this result involves purely algebraic methods, and is a nice way to illustrate how this theory can be useful to find model equivalences. We will be using to find to describe the expressibility of a concrete model, but if we are able to find a model with the same algebraic description, then we can show it is equivalent to ours.

This was for categorical distributions, where it is easy to represent the entirety of a distribution as a single point. For absolutely continuous distributions, whose domains are uncountable subsets of $\mathbb{R}^n$, this would not work. However, a huge benefit of modelling our continuous variables using Gaussian distributions is that it is actually possible to describe one with only two values, by using a classical result from Marcinkiewicz:

Theorem 5. *(Marcinkiewicz 1935) X follows a multivariate Gaussian iff $K_X(t)$ is a polynomial.*

We have not yet defined what $K_X(t)$ means, but we will do so in 5.1. The point is that a Gaussian distribution is completely determined by its mean vector and covariance matrix, and all its higher order *cumulants* are 0.

Proposition 4.7. *Let $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$. The conditional independence statement $X_A \perp\!\!\!\perp X_B \mid X_C$ if and only if the covariance matrix $\Sigma_{A\cup C, B\cup C} = \mathrm{Cov}(X_{A\cup C}, X_{B\cup C})$ has rank $|C|$.*

Proof. For this proof we will need to calculate the conditional distribution of a Gaussian, which is a well known result[1]:

$$X_{A\cup B}|X_C = x_c \sim \mathcal{N}\left(\mu_{A\cup B} + \Sigma_{A\cup B,C}\Sigma_{C,C}^{-1}(x_C - \mu_C), \Sigma_{A\cup B,A\cup B} - \Sigma_{A\cup B,C}\Sigma_{C,C}^{-1}\Sigma_{C,A\cup B}\right)$$

Thanks to the variables following a multivariate Gaussian distribution, the conditional independence statement will hold if and only if the A, B components (A rows and B columns or vice-versa) of the covariance matrix are equal to zero, that is, whenever

$$\left(\Sigma_{A\cup B,A\cup B} - \Sigma_{A\cup B,C}\Sigma_{C,C}^{-1}\Sigma_{C,A\cup B}\right) = \Sigma_{A,B} - \Sigma_{A,C}\Sigma_{C,C}^{-1}\Sigma_{C,B} = 0$$

Since $\Sigma_{C,C}$ is invertible of rank $|C|$ due to being a covariance matrix, and the previous expression is the Schur Complement of the matrix

$$\Sigma_{A\cup C, B\cup C} = \begin{pmatrix} \Sigma_{A,B} & \Sigma_{A,C} \\ \Sigma_{C,B} & \Sigma_{C,C} \end{pmatrix}$$

[1] thanks to stackexchange for the full derivation

Hence, we have that $\text{rank}(\Sigma_{A\cup C,B\cup C}) = |C|$. □

Therefore, since the condition of having a particular rank k corresponds to a set of equations being equal to 0 (the $k+1$ minors of the matrix) then this conditional independence condition may also be expressed as an algebraic constraint.

As before, this allows us to define an independence ideal like in 4.5, which Sullivant calls *The Gaussian conditional independence ideal* $J_{A\perp\!\!\!\perp B|C}$:

$$J_{A\perp\!\!\!\perp B|C} = \langle((|C|+1) - \text{minors of } \Sigma_{A\cup C,B\cup C}\rangle \tag{4.6}$$

And for a collection of conditional independence statements over our variables given by $\mathcal{C} = \{X_{A_1} \perp\!\!\!\perp X_{B_1}|X_{C_1}, X_{A_2} \perp\!\!\!\perp X_{B_2}|X_{C_2}\dots\}$ we may define the Gaussian conditional independence ideal as

$$J_{\mathcal{C}} = J_{X_{A_1}\perp\!\!\!\perp X_{B_1}|X_{C_1}} + J_{X_{A_2}\perp\!\!\!\perp X_{B_2}|X_{C_2}} + \dots \tag{4.7}$$

As we have seen, we may use the independence ideals to turn the conditional independence constraints of a given graphical model into algebraic ideals, such the conditions (P) and (G) of Markov Random Fields using either 4.5 for discrete models or 4.6 for Gaussian graphical models.

The challenge is that these conditions are easily defined when all the variables are discrete, as with the RBM, or when they are all Gaussian, where we can look at zeros in the precision matrix (equivalently vanishing minors of the covariance matrix). For a mixed model, the task would be quite hard, but since our particular mixed model will be such that the latent variables will be conditionally binary and the observables will be conditionally Gaussian, we are able to sidestep a lot of the complexity using mixture models.

4.2 Mixture Models

Even though we have not gone into much depth on the issue of hidden variables, simply stating that they were not observed and so they were not available to calculate the MLE, they suppose a challenge because standard asymptotic theory does not apply to them. The solution is for us, since we will be taking discrete hidden variables, is to model them using *mixture models*. This is because given a state of the latent variables, we will have a different observable distribution, and the marginal distribution will be a mixture of the different observables.

Definition 4.1. *Let V, W be two parameter sets, their* mixture model $\text{Mixt}(V,W)$ *is given by*

$$\text{Mixt}(V,W) = \{\lambda v + (1-\lambda)w : v \in V, w \in W, \lambda \in [0,1]\}$$

If you recall the factor analysis model discussed in the literature review, it could be modelled as a Gaussian graphical model with a bipartite structure, and with some of the conditional independence conditions that we had for the RBM and will want for our conditional Gaussian model in the next section. As such, it is a good example to illustrate what we are looking for.

The factor analysis model with m latent variables $(H_1, \dots, H_m)$ and n observables $(X_1, \dots, X_n)$ denoted $\mathcal{F}_{m,n}$, satisfies the conditional independence constraints

$$X_1 \perp\!\!\!\perp X_2 \perp\!\!\!\perp \dots \perp\!\!\!\perp X_n \mid (H_1, \dots, H_m)$$

Where the variables follow a joint multivariate Gaussian distribution. Note that as a graphical model, this would be given by a DAG with the edges pointing towards the visible variables. Thanks to the conditional Gaussian assumption, we can parametrise the model as the space of vectors $\boldsymbol{\mu} \in \mathbb{R}^n$ for the mean and covariance matrix Σ lying in the cone:

$$\mathcal{F}_{m,n} = \left\{ \Psi + \Lambda\Lambda^T \in \mathbb{R}^{n \times n} : \Psi \succ 0 \text{ diagonal}, \Lambda \in \mathbb{R}^{n \times m} \right\} \tag{4.8}$$

Which is a semi-algebraic set. To see this, note that the

$$\operatorname{Cov} \begin{pmatrix} X \\ H \end{pmatrix} = \begin{pmatrix} \Sigma & \Lambda \\ \Lambda^T & \Phi \end{pmatrix}$$

Which is an $(n+m) \times (n+m)$ matrix, and by 4.6 we have that the conditional independence assumptions of the model translate to the vanishing of the $m+1$ minors of the covariance matrix containing all the terms for the hidden variables, since those are the variables we are conditioning on. Assuming $i \neq j$ we obtain:

$$\det \begin{pmatrix} \sigma_{ij} & \Lambda_{i*} \\ \Lambda_{j*}^T & \Phi \end{pmatrix} = \det(\Phi) \cdot \left(\sigma_{ij} - \Lambda_{i*} \Phi^{-1} \Lambda_{j*}^T \right) = 0$$

By the Laplace expansion on the first entry. Hence, since $\det(\Phi) \neq 0$ it will be invertible and we can define the Schur complement $\Psi = \Sigma - \Lambda\Phi^{-1}\Lambda^T$ and now solving for the covariance we get $\Sigma = \Psi + \Lambda\Phi^{-1}\Lambda^T$, and taking $\Phi \succcurlyeq 0$ then we can use its Cholesky decomposition and we get the algebraic description for the covariance in equation 4.8.

And we have an algebraic description of the Factor Analysis model. This is only valid for a joint Gaussian distribution of all variables, which is not the case for the model we will be looking at, only the observable will follow a conditional Gaussian.

For our mixture model, let's assume a hidden categorical variable H with state space $[s]$, and for each $j \in [s]$, we have a distribution of the observable random variable X, given by P_j. Let

$\pi_j = \mathbb{P}(Y = j)$ so that $\pi \in \Delta_{s-1}$. In this way, we've made the marginal distribution of X a mixture distribution. If we let $p^{(j)} \in \mathcal{P}$, then we may write our distribution as

$$\text{Mixt}^s(\mathcal{P}) = \left\{ \sum_{j=1}^{s} \pi_j p^{(j)} \ : \ \pi \in \Delta_{s-1}, p^{(i)} \in \mathcal{P} \right\} \tag{4.9}$$

In the following chapter, we will be taking $p^{(j)}$ be conditionally independent Gaussians and analyse the model's properties.

5 Gaussian-Bernoulli RBM

Now that we have defined graphical models and seen some of their properties as related to their factorisation, inference and general computations, as well as seen how to model mixed variables through the Conditional Gaussian, we are ready to look at a class of models that have seldom been studied in statistical circles. Their use has been limited to the machine learning community for feature extraction and classification tasks in high-impact articles such as [30] or [31] and so results about them are more centered around their performance with respect to specific tasks or data sets, rather than having results on their theoretical/statistical foundations.

Therefore, in this main section we will provide a motivated and sound definition for the model, properly define it in the context of probabilistic graphical models, we will examine its statistical properties, and refrain from going into the machine learning discussions concerning hyperparameter selection or adequateness for a specific learning task, for which there already is extensive literature.

Building upon the Restricted Boltzmann Machine from 3.13, we will now consider the case where the observable variables are continuous and normally distributed, and we will consider the effect of different categorical latent structures, which will allow us model continuous data while keeping the simple discrete latent structure as before.

In this sense, our model is suited for mixed data, but with the requirement that the discrete variables be latent while the observables be continuous (which we will simplify as Gaussian).

For this reason, and as seen in the literature review, this model has a Gaussian mixture structure when marginalising over the visible variables but at the same time will retain the product structure that eases training

5.1 Model Definition and Parameters

As previously mentioned, our model is based on the bipartite structure of the Restricted Boltzmann Machine, with m conditionally binary latent variables $\mathbf{H} = (H_1, \dots, H_m)$ and n conditionally Gaussian observable variables $\mathbf{Y} = (Y_1, \dots, Y_n)$.

$$\begin{aligned} Y_i \perp\!\!\!\perp Y_j \mid \mathbf{H} = \mathbf{h} \\ H_i \perp\!\!\!\perp H_j \mid \mathbf{Y} = \mathbf{y} \end{aligned} \tag{5.1}$$

Definition 5.1. *The Gaussian-Bernoulli graphical model with m latent variables and n observables, denoted $GRBM_{m,n}$ is defined by the conditional independence rules in 5.1 and the conditional*

distributions

$$Y_i \mid \mathbf{H} = \mathbf{h} \sim \mathcal{N}\left(\mu(\mathbf{h}), \Sigma(\mathbf{h})\right)$$
$$H_i \mid \mathbf{Y} = \mathbf{y} \sim \text{Bern}\left(b(\mathbf{y})\right)$$

We will find the explicit expressions for the moment parameters b, μ, Σ later on. In fact, the parameters of the model will be written following convention as $\boldsymbol{\theta} = (\mathbf{a}, \mathbf{b}, \Sigma, W)$, where $\Sigma(h) = \Sigma$ so there will no dependence on the latent state to simplify the model, and since we will assume conditional independence by the graphical model structure, this matrix will also be diagonal. Then, $\mathbf{a}$ will affect the mean of the Gaussians and W will be the interaction term as in 3.1.

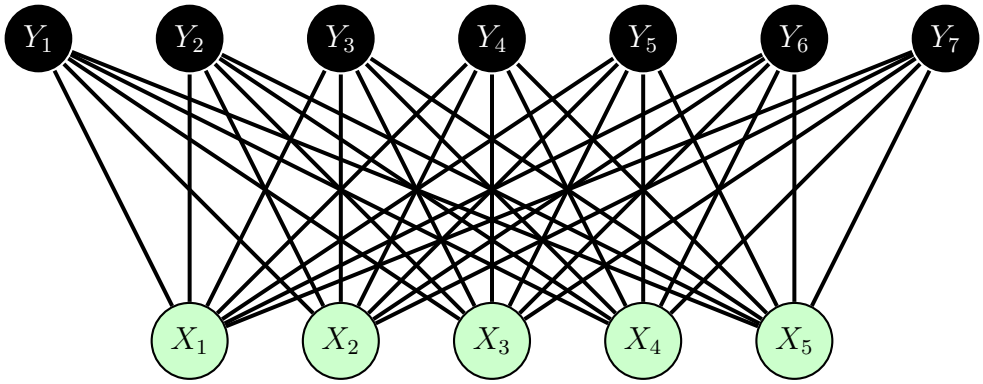


Figure 3: The graphical representation of $GRBM_{7,5}$ with 7 latent and 5 observable variables.

We may leverage theorem 3 along with the fact that the maximal cliques in a bipartite graph are just the edges to directly construct the density of the $GRBM$, which will be very similar to the RBM.

$$\begin{aligned} p(\mathbf{h}, \mathbf{y}; \boldsymbol{\theta}) &= \frac{1}{Z(\boldsymbol{\theta})} \exp\{-E(\mathbf{h}, \mathbf{y})\}, \quad \theta = (W, \Sigma, \mathbf{a}, \mathbf{b}) \\ E(\mathbf{h}, \mathbf{y}) &= -\frac{1}{2}(\mathbf{y} - \mathbf{a})^T \Sigma^{-1} (\mathbf{y} - \mathbf{a}) + \mathbf{h}^T \mathbf{b} + \mathbf{y}^T W \Sigma^{-1} \mathbf{h} \end{aligned} \tag{5.2}$$

Note the term highlighted in blue is quadratic in $\mathbf{y}$ and will provide us with the conditional Gaussian distribution of the observable variables.

Proposition 5.2. *The density in 5.2 indeed yields the properties promised in 5.1.*

Proof. To check the conditional distributions, we may compute directly from 5.2, where the integral is rather direct as a Gaussian integral, we later complete the square, and the only problem is the

somewhat hairy notation.

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{h}) &= \frac{p(\mathbf{h},\mathbf{y})}{\int_{\mathbf{y}} p(\mathbf{h},\mathbf{y})\,\mathrm{d}\mathbf{y}} \\
&= \frac{\exp\left\{-\frac{1}{2}(\mathbf{y}-\mathbf{a})^T\Sigma^{-1}(\mathbf{y}-\mathbf{a}) + \mathbf{h}^T\mathbf{b} + \mathbf{y}^T W\Sigma^{-1}\mathbf{h}\right\}}{\int_{\mathbf{y}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\mathbf{a})^T\Sigma^{-1}(\mathbf{y}-\mathbf{a}) + \mathbf{h}^T\mathbf{b} + \mathbf{y}^T W\Sigma^{-1}\mathbf{h}\right\}\mathrm{d}\mathbf{y}} \\
&= \frac{\prod_{i=1}^{n} \exp\left\{\frac{-(y_i-a_i)^2}{2\sigma_i} + \mathbf{b^T h} + \frac{y_i}{\sigma_i}\sum_{j=1}^{m} W_{ij}h_j\right\}}{\prod_{i=1}^{n}\sigma_i\sqrt{2\pi}\exp\left\{\frac{1}{2}\left(\sum_{j=1}^{m} W_{ij}h_j\right)^2 + \mathbf{b^T h} + \frac{b_i}{\sigma_i}\sum_{j=1}^{m} W_{ij}h_j\right\}} \\
&= \prod_{i=1}^{n}\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left(-\frac{1}{2{\sigma_i}^2}\left(y_i - a_i - \sigma_i\sum_{j=1}^{m} W_{ij}h_j\right)^2\right)
\end{aligned}
$$

Which we can see is the density of a multivariate Gaussian with diagonal covariance matrix. For the conditional density of a single latent with respect to the observables:

$$
\begin{aligned}
p(h_j = 1|\mathbf{y}) &= \frac{\sum_{h_{k\neq j}} p(h_j = 1, \mathbf{h}_{k\neq j}, \mathbf{y})}{\sum_h \exp\{-E(\mathbf{h},\mathbf{y})\}} \\
&= \frac{\exp\left\{\sum_{i=1}^{n}\frac{y_i}{\sigma_i}W_{ij} + b_j\right\}\sum_{h_{k\neq j}}\exp\{-E\left(h_j = 0, \mathbf{h}_{k\neq j}, \mathbf{y}\right)\}}{\sum_{h_{k\neq j}}\exp\{-E(h_j = 0, \mathbf{h}, \mathbf{y})\} + \sum_{h_{k\neq j}}\exp\{-E(h_j = 1, \mathbf{h}, \mathbf{y})\}} \\
&= \frac{1}{1 + \exp\left\{-\sum_{i=1}^{n}\frac{x_i}{\sigma_i}W_{ij} + b_j\right\}}
\end{aligned}
$$

□

Corollary 5.1. *We can now see the explicit expression for the parameters of the conditional distributions:*

$$
\begin{aligned}
\mathbf{Y} \mid \mathbf{H} = \mathbf{h} &\sim \mathcal{N}\left(\mathbf{a} + \Sigma W\mathbf{h}, \Sigma\right) \\
H_j \mid \mathbf{Y} = \mathbf{y} &\sim \mathrm{Bern}\left(S\left(\sum_{i=1}^{n}\frac{y_i}{\sigma_i}w_{ij} + b_j\right)\right)
\end{aligned}
$$

Where $S(x)$ denotes the sigmoid function, which again makes an appearance, of a similar nature as in the RBM case. We can also infer the conditional independence between variables so that the

distribution is faithful to the graph.

$$Y_1 \perp\!\!\!\perp \ldots \perp\!\!\!\perp Y_n \mid \mathbf{H}$$
$$H_1 \perp\!\!\!\perp \ldots \perp\!\!\!\perp H_m \mid \mathbf{Y}$$

For the marginal distribution of the observable variables, given by $\sum_h \exp\{-E(\mathbf{h}, \mathbf{y})\}$, we may also write it using its factorisation with the conditional Gaussian density $p(\mathbf{y}|\mathbf{h})$

$$p(\mathbf{y}) = \sum_{\mathbf{h}} p(\mathbf{y}|\mathbf{h})p(\mathbf{h}) = \sum_{\mathbf{h}} p(\mathbf{h})\mathcal{N}\left(\mathbf{a} + \Sigma W\mathbf{h}, \Sigma\right) \tag{5.3}$$

With a subtle abuse of notation, but we can see how this is a mixture of *independent* Gaussians, since $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$ is diagonal. Therefore this marginal distribution is a mixture of 2^m independent Gaussians with means lying on the corners of a parallelepiped, since they are given by $\mathbf{a} + W\mathbf{h}$, where the $W\mathbf{h}$ is common to all Gaussian observables. This makes modelling with this model not as trivial as with general Gaussian Mixture Models, more details in [32].

5.2 Latent Tree Model

A first approximation is that of making the graph a tree model, which allows us to benefit from the existing literature on Latent Tree Models, such as [33], and the graph theory results on trees. This simplification also includes the naive Bayes and Factor Analysis models from 3.1.4 and 2.2. If we take a latent tree DAG version, with the induced topological ordering, we could use the Bayesian Network factorisation for the density to simplify computations, but by standard results on on Markov equivalence of DAGs, which we can see by comparing the factorisations of the two models when the underlying graph is a tree, a Bayesian network on a tree is equivalent to the Latent Tree Model (an undirected tree) over the same tree $\mathcal{T}$ with undirected edges. The tree case is trivial, but for more complex equivalences between Bayesian Networks and Markov Random Fields, see [34].

A distribution that can be represented by a tree with n leaves and m internal vertices will have a number of parameters given by:

- $n + n$ total, coming from the pseudo-means and variances of the Gaussians (like $\mathbf{a}$ and Σ from 5.1)
- m total, coming from the binary variables (like the $\mathbf{b}$ parameters)
- $n + m - 1$ total, one for each edge of the tree (like the W parameters).

Which gives a grand total of $3n+2m-1$ parameters. That being said, this is *over-parametrised* since there are different sets of parameters which yield the same joint distribution. This follows from the fact that there always exists (for trees with more than one vertex) a relabeling of the nodes that yields the same graphical model. This is a classical result by Erdös and Rényi in [35], where after Theorem 4 they make a neat argument about the existence of such relabelings. This implies that the model is not identifiable.

Definition 5.1. *We say a parametric model is* identifiable*, if the map from the parameter space Θ to the model is injective. That is, for a statistical model $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$, it is the case that:*

$$P_\theta = P_{\theta'} \implies \theta = \theta'$$

And so for in the limit where the number of random samples goes to infinity, we can recover the parameters of the model.

Therefore, we can only recover the parameters up the tree automorphisms that respect the graphical model distribution. As an example, in figure 4, by switching around the labels Y_1, Y_2, Y_3, the model remains the same.

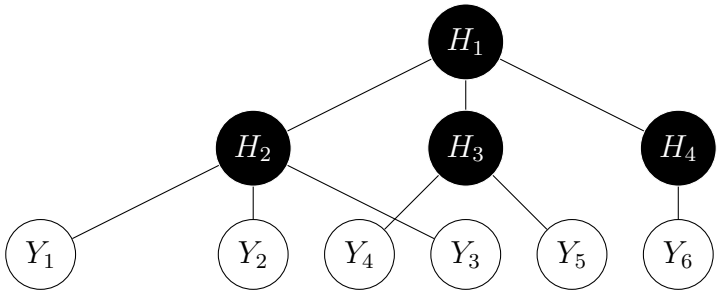


Figure 4: An example of a latent tree model with 6 Gaussian observable variables and 4 binary latent variables

This lack of complete identifiability, for the same reason, will extend to the $GRBM$ model, only the automorphisms of a bipartite graph are very simple to state, since the group of automorphisms of a complete bipartite graph $\mathcal{K}_{n,m}$ is isomorphic to $S_n \times S_m$ for $n \neq m$, where S_n is the permutation group on n elements. Of course if $n = m$ then we get another extra symmetry and the graph automorphism group becomes $S_n \times S_m \times \mathbb{Z}_2$. Other than this trivial label swapping, [36] used algebraic methods to show the generic identifiability in Markov models, when taking into account the relabeling.

In terms of the expressibility of the model, it is clear that by changing the value of the $\mathbf{a}$ parameters, any mean in $\mathbb{R}^n$ is attainable, so it is fully expressible in this sense. In fact, it is also

easy to study the expectation because it is a *linear* function of the latent variables, since every observable Y_i will have a single parent H_j, and the expectation will be given by:

$$\mathbb{E}[Y_i|H_j] = \mu^{(i)}_{H_j=0}(1-H_j) + \mu^{(i)}_{H_j=1}H_j \tag{5.4}$$

And the covariance between two leaves Y_{i_1}, Y_{i_2} with the same parent $H_j \sim \text{Bern}\,(\alpha)$ will be given by

$$\begin{aligned}\text{Cov}(Y_{i_1}, Y_{i_2}) &= \text{Cov}(\mathbb{E}[Y_{i_1}|H_j], \mathbb{E}[Y_{i_2}|H_j]) \\ &= \mathbb{E}[\mathbb{E}[Y_{i_1}|H_j]\,\mathbb{E}[Y_{i_2}|H_j]] - \mathbb{E}[Y_{i_1}]\,\mathbb{E}[Y_{i_2}] \\ &= \alpha(1-\alpha)\left(\mu^{(i_1)}_{H_j=1} - \mu^{(i_1)}_{H_j=0}\right)\left(\mu^{(i_2)}_{H_j=1} - \mu^{(i_2)}_{H_j=0}\right)\end{aligned} \tag{5.5}$$

which is a linear function of $\text{Var}(H)$. In terms of the covariance, there is a limitation in its possible range of values, given the nature of the second order moments in a tree, as given by the following result for the correlation of Gaussian r.v.s on all nodes from [33].

Proposition 5.2. *For any two nodes $X_i, X_j \in \mathcal{T}$ in a latent tree model, we have that their correlation ρ_{ij}*

$$\rho_{ij} = \prod_{(u,v)\in\overline{ij}} \rho_{uv} \tag{5.6}$$

where $\overline{ij}$ is the (unique) path between X_i and X_j.

Proof. The result follows directly from the calculation of $\text{Cov}\,(X_i|H_j)$ where X_i and H_j are neighbours, using the law of total covariance.

$$\begin{aligned}\text{Cov}(X_i, H_j) &= \mathbb{E}[\text{Cov}(X_i, H_j|H_j)] + \text{Cov}(\mathbb{E}[X_i|H_j], \mathbb{E}[H_j|H_j]) \\ &= \cancel{\mathbb{E}[\text{Cov}(X_1, H|H)]} + \text{Cov}((1-H_j)\mu_{i0} + H\mu_{i1}, H_j) \\ &= \text{Cov}(\mu_{i0} + H(\mu_{i1}-\mu_{i0}), H_j) = (\mu_{i1}-\mu_{i0})\alpha(1-\alpha)\end{aligned}$$

where we used $\mu_{ik} = \mathbb{E}\,[X_i|H_j = k]$ and $\alpha = \mathbb{P}(H_j = 1)$. From this we can infer the relation

$$\text{Cov}(X_1, H)\,\text{Cov}(X_2, H) = \text{Cov}(X_1, X_2)\,\text{Var}(H)$$

Where we have only used the fact that H_j was binary, so we can extend this for the whole path, where dividing by the standard deviations gives us the correlation, thus yielding the product result we wanted. □

Corollary 5.2. *From this previous result on correlations we obtain a sort of triangle inequality for the correlations of the the leaves of the tree: for any triplet of leaves Y_i, Y_j, Y_k we have that:*

$$\rho_{ij} \geq \rho_{ik}\rho_{jk}$$

And so the space of correlations of three variables is the semi-algebraic variety given by the intersection of the sets:

$$\begin{cases} x \geq yz \\ y \geq zx \\ z \geq xy \\ |x|, |y|, |z| \leq 1 \end{cases}$$

Plotted in figure 5. Regarding the triangle inequality statement made, by taking $d_{ij} = -\log|\rho_{ij}|$ we have a distance metric on the tree, with many applications in phylogenetics and others (see Zwiernik in [34]).

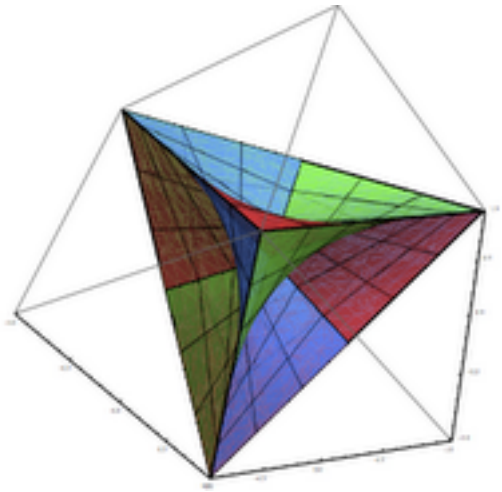


Figure 5: Correlation space for three Gaussian variables from the cover of [33]

Those are all the correlations a Gaussian-Bernoulli latent tree model can attain, as the interior of the semi-algebraic variety in the previous figure 5, which is contained in the tetrahedron with vertices given by $(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)$ but with the faces curving inwards.

5.3 Learning and Sampling Procedures

5.3.1 Gibbs Sampling

Sampling from a distribution is essential to carry out a lot different procedures related to a distribution, like for instance to simulate data, to compute approximate moments when these are not explicitly available, or to compute expectations with respect to this distribution. Sampling for general graphical models can be a difficult task due to the dependence between variables, so a naive Monte Carlo sampling is not appropriate. However, we may use a Markov Chain Monte Carlo method, which aims to construct a Markov chain whose limiting distribution is given by the distribution we wish to sample from, and take the samples from it. The theory of MCMC methods is full of beautiful ideas but too vast to go into for this project; for an introduction, check out [37].

As a matter of fact, using 5.1 and the conditional independence structure of the variables, we can sample from the *GRBM* using an especially simple kind of Metropolis-Hastings MCMC procedure known as *Gibbs Sampling*, which works as follows:

Algorithm 1 Gibbs Sampling

Initialize $(\mathbf{h}^0, \mathbf{y}^0)$ ▷ With some generic point
for $s = 1, 2, \ldots$ **do** ▷ As many samples as required
  for $i \in \{1, \ldots, n\}$ **do**
    Sample ${y_i}^s \sim p(y_i|\mathbf{h}^{s-1})$ ▷ Directly from 5.1
  end for
  for $j \in \{1, \ldots, m\}$ **do**
    Sample ${h_j}^s \sim p(h_j|\mathbf{y}^{s-1})$ ▷ Again from 5.1
  end for
end for

This algorithm makes the sampling procedure completely trivial, thanks to having explicit and easy conditional distributions. Note that a usual problem with MCMC procedures though, is that they take some time to converge to the actual distribution, so some practitioners simply discard some of the first samples, sometimes up to the first third [37].

5.3.2 Expectation-Maximisation Algorithm

In order to learn the parameters of the model, again we run into the problem that the latent are not observed, and so we cannot find the maximum likelihood directly. As such, we can adapt the EM procedure from 3.5 to this case:

- E-Step:

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) &:= \mathbb{E}_{\mathbf{H}\sim p(\cdot|\mathbf{Y},\boldsymbol{\theta}^{(t)})} \left[\ln p(\mathbf{H}, \mathbf{Y}; \boldsymbol{\theta})\right] \\ &= \sum_{\mathbf{H}} \ln p(\mathbf{H}, \mathbf{Y}; \boldsymbol{\theta}) p(\mathbf{H}|\mathbf{Y}; \boldsymbol{\theta}^{(t)}) \end{aligned}$$

- M-Step:

$$\boldsymbol{\theta}^{(t+1)} := \arg\max_{\boldsymbol{\theta}} \{Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})\}$$

Where we have explicit expressions for the conditional $p(\mathbf{H}|\mathbf{Y}; \boldsymbol{\theta}^{(t)})$ and the joint density $p(\mathbf{H}, \mathbf{Y}; \boldsymbol{\theta}^{(t)})$, but the expectation is calculated over all $\mathbf{H}$ states, of which there are 2^m, and each iteration requires the recalculation of the normalising constant $Z(\boldsymbol{\theta})$, which makes it intractable. This makes this procedure viable for a small number of latent states, but for a large number the computational cost becomes too large, especially if we don't know in advance how many latent variables we want, and take the number as a hyperparameter to be selected. In order to avoid this intractability, we may use the Contrastive Divergence procedure defined in [24], which introduces bias to avoid having to calculate the expectation with an exponential number of terms.

This procedure has been "enormously valuable" [25] in practical applications and is based, in broad strokes, on approximating the maximum likelihood in the E-Step of the EM algorithm using Gibbs sampling, but instead of sampling until convergence, we just run the sampling for a small number of iterations, initialising at the points provided from the data, and use that to estimate the function $Q(\boldsymbol{\theta}, \boldsymbol{\theta^{(t)}})$ [24].

5.4 Expressibility

By the expressibility of the model we mean what possible distributions can be represented by it. When looking at the expressibility of a model, there are two main ways that interest us:

Question 1. *Are there any convergence results of our model towards some given fixed distribution?*

Question 2. *Can our model approximate the moments of a fixed distribution arbitrarily well?*

Regarding the first question, it is often known in the machine learning literature as a *Universal Approximation* result, of which we can provide an almost positive answer.

Regarding the second question, we propose an algebraic formulation in order to conjecture a possible positive result. These two questions are expanded upon in the following subsections.

5.4.1 Approximation Properties

Recall we had a Universal Approximation result in theorem 1 for neural networks with arbitrary width and height. We are interested in whether there is a similar result for our GRBMs. Intuitively, we could have a positive result given that the marginal distribution of the GRBM is given by a Gaussian mixture, and Gaussian Mixtures Models are universal approximators, as mentioned in page 65 of [38].

That being said, the GRBM marginal is unlike the general Gaussian mixture since in our case we have a mixture of 2^m *independent* Gaussians, and the means of each component of the mixture are structured in a parallelepiped, as explained in 5.3. For this reason, as already mentioned in the definition, modelling with these GRBMs has its complications due to the restrictions in the model structure [32]. In fact, the universal approximation of GRBMs is still an open problem, but when replacing the sigmoid activation with a ReLU, it was shown in 2022 [39] that the model becomes a universal approximator. On top of that, the paper also showed that the stacking of two latent layers as if increasing the depth of a neural network, which is often termed the *Gaussian-Bernoulli Deep Belief Network* or GB-DBN, increases the representational power with respect to our GRBM enough to be able to prove a universal approximation result. A depiction of this extension with three hidden layers and directed conditionals, with the undirected RBM component on top, is provided in figure 6.[2]

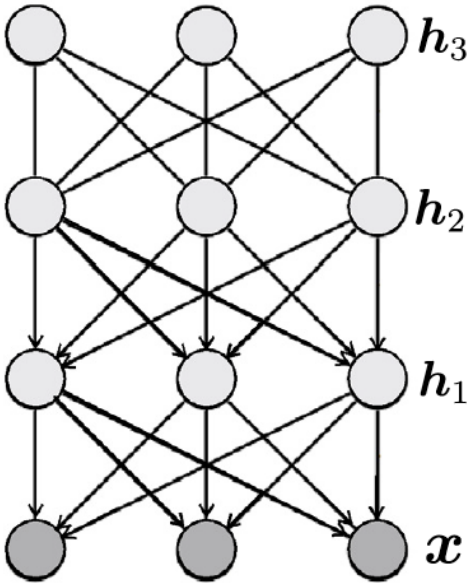


Figure 6: Illustration of the GB-DBN extension of the GRBM, from [39]

[2]In case one wishes to look further, this type of graph with mixed edges is known as a *chain graph*.

For results on the approximation rate, from the machine learning perspective we have the same article from 2022 [39] providing the convergence rate of the GB-DBN model. In this extension of the model, they provide the result of $O(\epsilon^{-2})$ hidden units per layer for a double latent layer, meaning that, module a constant value, we would need to quadruple the the number of hidden units in each of the two layers to half the approximation error.

5.4.2 Model Cumulants

Now, moving onto the second question, we shall be looking at the possible moments that we can attain with our model. That being said, we wont be looking exactly at the moments but at the cumulants of our model, defined as follows.

Definition 5.1. *The* nth-order cumulant κ_n *of a random variable* X *is given by the* n*th term of the power series expansion of the* cumulant generating function*, which is defined in terms of the moment-generating function as*

$$K_X(t) = \log M_X(t) = \log \mathbb{E}[e^{tX}] \tag{5.7}$$

Thus, the nth order cumulant κ_n *is given by* $K^n(0)$ *in*

$$K(t) = \sum_{n=1} \frac{\kappa_n}{n!} = \kappa_1 t + \kappa_2 \frac{t^2}{2} + \kappa_3 \frac{t^3}{6} + \cdots + \kappa_k \frac{t^k}{k!} + \ldots \tag{5.8}$$

Remark 5.1. *The cumulant generating function of a joint distribution* $X = (X_1, \ldots, X_k)$ *is given by*

$$K_X(t_1, \ldots, t_k) = \log \mathbb{E}[\exp\left(\sum_j t_j X_j\right)]$$

The reason for us to prefer working with cumulants instead of moments is due them having some very desirable statistical and combinatorial properties.

Proposition 5.2. *Some basic properties of cumulants are as follows:*

(i) For $n > 1$ *and constant* c, $\kappa_n(c + X) = \kappa_n(X)$ *(translation invariant).*

(ii) For constant c, $\kappa_n(cX) = c^n \kappa(X)$ *(homogeneous of degree* n*).*

(iii) For $X_1, \ldots, X_m$ *independent r.v.s we have* $\kappa_n(X_1 + \cdots + X_m) = \kappa_n(X_1) + \cdots + \kappa(X_m)$

The first and second order cumulant of X are, conveniently, the expectation $\mathbb{E}[X]$ and the covariance $\mathrm{Cov}(X, X)$. On top of that, we had Marcinkiewicz's result from 5, concerning the

cumulants of the Gaussian distribution. Another strong reason to prefer cumulants is a somewhat obscure result by Brillinger generalising the law of total covariance known as the *Law of Total Cumulance* [40]:

Theorem 6. *(Law of total cumulance)*

$$\kappa(X_1, \ldots, X_k) = \sum_{\pi} (\kappa(\kappa(X_i : i \in B|Y) : B \in \pi))$$

Where Y is the random variable we are conditioning on and π are partition blocks of $\{1, \ldots, k\}$.

Using this result, we may easily find the higher order cumulants of any set of observables (without repetitions) from our model thanks to the independence between them when conditioning on the latent variables. For instance, consider the covariance between any two variables $Y_i, Y_j, i \neq j$ from our model:

$$\begin{aligned}
\operatorname{Cov}(Y_i, Y_j) &= \mathbb{E}(\operatorname{Cov}(Y_i, Y_j|H)) + \operatorname{Cov}\left(\mathbb{E}(Y_i|H), \mathbb{E}(Y_j|H)\right) \\
&= \cancel{\mathbb{E}(\operatorname{Cov}(Y_i, Y_j|H))} + \operatorname{Cov}\left(\mathbb{E}(Y_i|H), \mathbb{E}(Y_j|H)\right) \\
&= \mathbb{E}\left(\mathbb{E}(Y_i|H)\,\mathbb{E}(Y_j|H)\right) - \mathbb{E}\left(\mathbb{E}(Y_i|H)\right)\mathbb{E}\left(\mathbb{E}(Y_j|H)\right) \\
&= \mathbb{E}\left[\left(\sum_h \mu_h^{(i)} \mathbb{1}_{\{H=h\}}\right)\left(\sum_h \mu_h^{(j)} \mathbb{1}_{\{H=h\}}\right)\right] - \mathbb{E}(Y_i)\,\mathbb{E}(Y_j)
\end{aligned} \tag{5.9}$$

Where we used $\mu_h^{(i)}$ for the conditional mean of Y_i given the latent state $H = h$, and $\mathbb{1}_A$ for an indicator function of the event A. We can see from the previous expression how we could characterise the covariance of our model in terms of a multinomial product For a general n-th order cumulant we would have:

$$\begin{aligned}
\kappa(Y_{i_1}, \ldots, Y_{i_r}) &= \kappa\left(\kappa(Y_{i_1}|H), \ldots, \kappa(Y_{i_r}|H)\right) \\
\mathbb{E}[Y_i|H] &= \sum_h \mu_h^{(i)} \mathbb{1}\{H = h\}
\end{aligned} \tag{5.10}$$

Unlike the tree case where we had the simple linear expressions for the expectation and the covariance, as in equations 5.4 and 5.5, now these will not be linear. That being said, there is still little to say about the expectation of the model, as we can trivially attain any expectation we wish by moving the $\mathbf{a}$ parameters for the $\mathbf{H} = \mathbf{0}$, and the others will be given by the edges of the parallelepiped given by $W\mathbf{h}$. With the law of total expectation, we may describe the mean of any observable Y_i as:

$$\mathbb{E}[Y_i] = \mathbb{E}[\mathbb{E}[Y_i|H]] = \sum_h \mu_h^{(i)}\, \mathbb{P}(H = h) \tag{5.11}$$

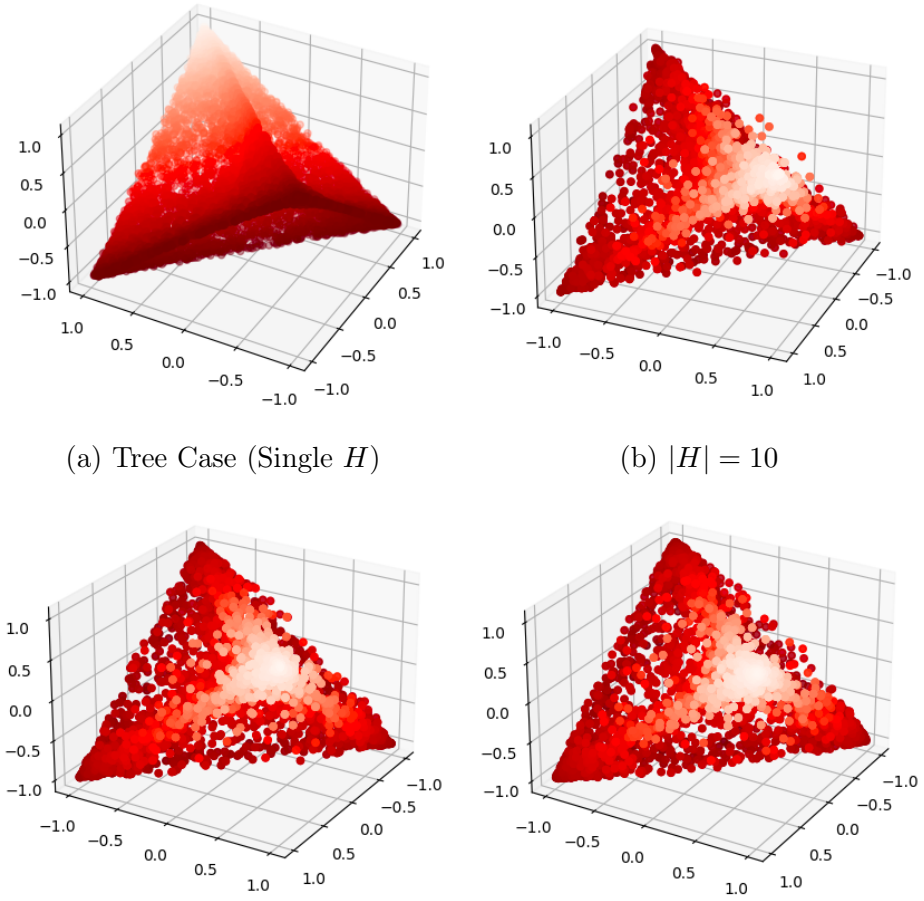
As the combination of the means for each of the 2^m latent states. Concerning the second cumulant, the covariance, things were not so trivial. As with the tree case, we will want to look at the normalised covariance (correlation) to make things more visual. Recall that for three variables, the correlation matrix is given by:

$$R = \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix}$$

Whose determinant is therefore $\det(R) = 1 - x^2 - y^2 - z^2 + 2xyz$ and to the set of points (x, y, z) where this is positive, we may call it the *correlation variety*. We saw that with the tree model there were many different correlations that weren't attainable, but with the $GRBM$ this is no longer the case. In figure 7 we have the results of some simulations using uniformly generated parameters, with one of the three correlation values on each axis, thanks to the fact that we can use Gibbs Sampling to generate samples of our distribution, where we increased the number of hidden variables.

Even though the images represent a noisy version of the actual correlation space, we can clearly appreciate how increasing the number of hidden variables increases the space of correlations attainable by our model, increasing the volume spanned by the samples correlations produced, and we conjecture that the space described by our model can be made to be uniformly and arbitrarily close to that of the equation $\det(R) = 1 - x^2 - y^2 - z^2 + 2xyz,\ |x|, |y|, |z| \leq 1$.

For the code used to generate the samples in `Python3`, check out this repository.



(a) Tree Case (Single H)

(b) $|H| = 10$

(c) $|H| = 50$

(d) $|H| = 100$

Figure 7: Simulated correlations of a GRBM model with 3 observable variables and different numbers of latent variables

Moving on to higher order cumulants, the parametrisation in 5.11 we found to be rather cumbersome, and so there was a tensor notation that we found was more illuminating; in many instances it is more useful to talk about the means with another tensor basis, noting that $H = (H_1, \ldots, H_m)$ can be re-written using $\overline{H}_i = (1, H_i)$, and then

$$\overline{H} = \left(\overline{H}_1 \otimes \overline{H}_2 \otimes \cdots \otimes \overline{H}_m\right) : \mathbb{R}^{2\times\cdots\times 2} \longrightarrow \mathbb{R}$$

And using this notation we can think the conditional mean as consisting of a base term corresponding to $H_1 = \cdots = H_m = 0$, and some offsets that are added when certain latent variables are "active" giving us a tensor $a_{i_1 i_2 \cdots i_m}$, $i_j \in 0, 1$:

$$A^{(i)} \in \mathbb{R}^{2\times\cdots\times 2}, \quad A^{(i)} \star \overline{H} = \mathbb{E}\left[Y_i | H\right]$$

Where we define the $\star$ operation on tensors so as to combine them to give what we describe in the following expression of our conditional expectation:

$$\begin{aligned}\mathbb{E}[Y_i|H] = a_{0\ldots0} + a_{10\ldots0}H_1 + \ldots \\ + a_{0\ldots01}H_m + \cdots + a_{1\ldots1}H_1 \cdots H_m\end{aligned} \tag{5.12}$$

Unlike the tree case, where the conditional mean was a linear function of the latent variables, here we have the conditional mean is a polynomial function of the latent variables, and so we need to incorporate higher order tensors.

In this way, the arbitrary-order cumulants we had in 5.10 could be algebraically manipulated using the properties of cumulants to get:

$$\kappa\left(Y_{i_1}, \ldots, Y_{i_r}\right) = \kappa\left(A^{(i_1)} \star \overline{H}, \ldots, A^{(i_r)} \star \overline{H}\right) = (A^{(i_1)}, \ldots, A^{(i_r)}) \star \kappa(\overline{H}, \ldots, \overline{H})$$

We already saw for the covariance how this ended up being a multinomial product of two terms, and in general we will have a multinomial product of r terms for the r-th order cumulant. And so our problem of calculating these cumulants reduces to calculating multinomial products and the the cumulants of binary random vectors, which to the best of our knowledge is a combinatorial open problem. Therefore, in order to progress in this direction, combinatorial tools must be developed and applied in this realm, and the cumulants of this conditional Gaussian model with fall with them.

6 Conclusion

As a conclusion, this work has been an overview of probabilistic models with recent developments and applications, how to formalise them as statistical models and, hopefully, an interesting way to motivate why they work the way they do. The main constructions of Restricted Boltzmann Machines and our GRBM model have been provided, and an review of their strengths and weaknesses with respect to concerns of identifiability, parameter estimation, expressibility, and relation to other known models has been covered. The parametrisation provided for the higher order cumulants of the GRBM model serve as a possible starting point for further research, and demonstrate once again the power of tensor algebra to advance our knowledge of statistics.

In addition, we gave background on the use of the somewhat mysterious sigmoid function, usually given in machine learning courses as a simple non-linear transformation of the outputs, and how it ties together the conditional density in harmonium structures.

There are also several open problems concerning these data modelling challenges which touch upon purer areas of mathematics such as algebraic geometry and especially combinatorics, and further work in this direction would help to better understand the backbone of the machine learning models used in the cutting-edge technologies and state-of-the-art algorithms that dominate the computer vision, natural language processing and reinforcement learning paradigms. Some work has been done on general exponential family distributions over the nodes in a harmonium, but again this work, while recent, mostly comes from the machine learning community and so the results are more practical than theoretical in nature, so this is definitely another possible future extension.

What the author also finds really fascinating about this area is how easily you can find yourself down deep rabbit holes that go in completely different directions. For instance, when looking at graphical models one can ask questions about sampling theory and MCMC, or inquire about how the graph properties determining the conditional independence structure can be used to model causality, or even probe into the depths of exponential families and their rich information-theoretic properties. That without even mentioning the vast jungle of numerical methods and operations research algorithms used to train learning algorithms. At every junction there is a possibility for new explorations, and despite the difficulties this posed to the author's focus, this was an invaluable introduction into such a plethora of topics for further study.

References

[1] Karl Pearson and Olaus Magnus Friedrich Erdmann Henrici. "III. Contributions to the mathematical theory of evolution". In: *Philosophical Transactions of the Royal Society of London. (A.)* 185 (1894), pp. 71–110. DOI: 10.1098/rsta.1894.0003. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1894.0003. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1894.0003.

[2] Carlos Améndola, Marta Casanellas, and Luis David García-Puente. "Tapas of Algebraic Statistics". In: *Notices of the American Mathematical Society* (2018).

[3] Christopher M. Bishop. *Pattern recognition and machine learning*. Information Science and Statistics. Springer, New York, 2006. ISBN: 0387310738. DOI: 10.1007/978-0-387-45528-0.

[4] R. Fletcher. *Practical Methods of Optimization*. A Wiley-Interscience publication. Wiley, 2000. ISBN: 9780471494638.

[5] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[6] Michael JD Powell. "Radial basis functions for multivariable interpolation: a review." In: *Algorithms for the Approximation of Functions and Data.* (1985).

[7] Carlos Améndola, Alexander Engström, and Christian Haase. "Maximum number of modes of Gaussian mixtures". In: *Information and Inference: A Journal of the IMA* 9.3 (June 2019), pp. 587–600. DOI: 10.1093/imaiai/iaz013. URL: https://doi.org/10.1093/imaiai/iaz013.

[8] Karl Pearson F.R.S. "LIII. On lines and planes of closest fit to systems of points in space". In: *Philosophical Magazine Series 1* 2 (1901), pp. 559–572.

[9] Michael E. Tipping and Christopher M. Bishop. "Probabilistic Principal Component Analysis". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622. ISSN: 13697412, 14679868. URL: http://www.jstor.org/stable/2680726.

[10] A.T. Basilevsky. *Statistical Factor Analysis and Related Methods: Theory and Applications*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN: 9780471570820.

[11] Geoffrey E Hinton. "Training products of experts by minimizing contrastive divergence". In: *Neural computation* 14.8 (2002), pp. 1771–1800.

[12] Iqbal Sarker. "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions". In: *SN Computer Science* 2 (Aug. 2021). DOI: 10.1007/s42979-021-00815-1.

[13] Kai Fong Ernest Chong. "A closer look at the approximation capabilities of neural networks". In: *International Conference on Learning Representations*. 2020.

[14] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert L. White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2 (1989), pp. 359–366.

[15] Moshe Leshno et al. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(05)80131-5.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012.

[17] Yann LeCun. "Who is afraid of non-convex loss functions?" NIPS. 2007. URL: https://cs.nyu.edu/~yann/talks/lecun-20071207-nonconvex.pdf.

[18] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2017. ISBN: 978-0-262-03731-0. URL: https://mitpress.mit.edu/books/elements-causal-inference.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38. ISSN: 00359246.

[20] S.L. Lauritzen. *Graphical Models*. Oxford Statistical Science Series. Clarendon Press, 1996. ISBN: 9780191591228.

[21] R.J. Baxter. *Exactly Solved Models in Statistical Mechanics*. Dover books on physics. Dover Publications, 2007. ISBN: 9780486462714.

[22] Paul Smolensky. "Information processing in dynamical systems: Foundations of harmony theory". In: *Parallel Distributed Process* 1 (Jan. 1986).

[23] Yoav Freund and David Haussler. "Unsupervised Learning of Distributions of Binary Vectors Using 2-Layer Networks". In: *NIPS*. 1991.

[24] Miguel Á. Carreira-Perpiñán and Geoffrey Hinton. "On Contrastive Divergence Learning". In: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. Ed. by Robert G. Cowell and Zoubin Ghahramani. Vol. R5. Proceedings of Machine Learning Research. Reissued by PMLR on 30 March 2021. PMLR, Jan. 2005, pp. 33–40.

[25] Guido Montúfar. "Restricted Boltzmann Machines: Introduction and Review". In: *ArXiv* abs/1806.07066 (2018).

[26] S. L. Lauritzen and N. Wermuth. "Graphical Models for Associations between Variables, some of which are Qualitative and some Quantitative". In: *The Annals of Statistics* 17 (1989), pp. 31–57. DOI: 10.1214/aos/1176347003. URL: https://doi.org/10.1214/aos/1176347003.

[27] Mathias Drton, Bernd Sturmfels, and Seth Sullivant. *Lectures on Algebraic Statistics*. Vol. 39. Oberwolfach Seminars. Springer, 2009. DOI: 10.1007/978-3-7643-8905-5.

[28] Persi Diaconis and Bernd Sturmfels. "Algebraic algorithms for sampling from conditional distributions". In: *The Annals of Statistics* 26.1 (1998), pp. 363–397. DOI: 10.1214/aos/1030563990. URL: https://doi.org/10.1214/aos/1030563990.

[29] Anna Seigal and Guido Montufar. "Mixtures and products in two graphical models". In: (2017). DOI: 10.48550/ARXIV.1709.05276. URL: https://arxiv.org/abs/1709.05276.

[30] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[31] Jan Melchior, Nan Wang, and Laurenz Wiskott. "Gaussian-binary restricted Boltzmann machines for modeling natural image statistics". In: *PLOS ONE* 12.2 (Feb. 2017), pp. 1–24. DOI: 10.1371/journal.pone.0171015.

[32] Oswin Krause et al. "Approximation properties of DBNs with binary hidden units and real-valued visible units". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, June 2013, pp. 419–426.

[33] Piotr Zwiernik. "Latent tree models". In: *Handbook of graphical models*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, 2019, pp. 265–288. URL: https://books.google.ca/books?id=g7J5DwAAQBAJ.

[34] M. Maathuis et al. *Handbook of Graphical Models*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, 2018. ISBN: 9780429874239.

[35] Paul Erdos and Alfréd Rényi. "Asymmetric graphs". In: *Acta Math. Acad. Sci. Hungar* 14.295-315 (1963), p. 3.

[36] "Full reconstruction of Markov models on evolutionary trees: Identifiability and consistency". In: *Mathematical Biosciences* 137.1 (1996), pp. 51–73. ISSN: 0025-5564. DOI: https://doi.org/10.1016/S0025-5564(96)00075-2.

[37] N. Chopin and O. Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer Series in Statistics. Springer International Publishing, 2020. ISBN: 9783030478452.

[38] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016. ISBN: 9780262035613.

[39] Linyan Gu, Lihua Yang, and Feng Zhou. "Approximation properties of Gaussian-binary restricted Boltzmann machines and Gaussian-binary deep belief networks". In: *Neural Networks* 153 (2022), pp. 49–63. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2022.05.020.

[40] David Brillinger. "The calculation of cumulants via conditioning". In: *Annals of the Institute of Statistical Mathematics* 21 (Dec. 1969), pp. 215–218. DOI: 10.1007/BF02532246.