LEARNING GRAPHICAL MODELS WITH LIMITED OBSERVATIONS OF HIGH-DIMENSIONAL DATA

by

Erdem Yörük

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

July, 2011

Abstract

In a variety of computational domains, the number of samples available for learning remains relatively small as compared to increasing data dimensions. This is common in computational biology and vision, posing even greater challenges when combined with complex interactions among variables. Consequently, the bias-variance trade-off requires one to invest model parameters with the utmost care for robust learning. In the small sample context, we argue for incorporating the available prior knowledge, and introducing carefully chosen biases to reduce variance. Motivated by this philosophy and particular problems from biology and vision, we propose two new generative approaches within a graphical model formalism: (a) A comprehensive statistical model for analyzing cell signaling networks, and (b) A restricted family of latent variable forest models for discovery of complex dependencies.

The first method is particular to protein signaling networks, which play a central role in transcriptional regulation and the etiology of many diseases. With known molecular connections, our model is anchored to a pre-defined core signaling topology. It has a limited complexity due to parameter sharing and uses expression data of

ABSTRACT

target genes as the only observable components. Specifically, we account for cell heterogeneity and a multi-level process, representing signaling as a Bayesian network at the cell level, modeling measurements as ensemble averages at the tissue level and incorporating patient-to-patient differences at the population level. We applied our method to the RAS-RAF network using a breast cancer study. We demonstrated robust statistical inference, established reproducibility through simulations and the ability to recover receptor status from available microarray data.

Our second method addresses the deeper endeavor of model selection. We propose a restricted family of forest structured distributions which are Markov with observed leaf variables regulated hierarchically by non-terminal latent variables. With a nested design, our model family allows a well-principled stepwise discovery of dependencies via sequential aggregations of pending substructures. Using particular parametric choices, we prove identifiability of our models and exact inference via dynamic programming. We apply our generative approach to synthesis and classification of handwritten digits, and to phenotype prediction from microarray data, with performances comparable to the state-of-the-art discriminative methods.

Primary Reader: Dr. Donald Geman

Secondary Reader: Dr. Laurent Younes

Acknowledgments

First and foremost, I would like to express my immense appreciation of my two dear advisors *Dr. Donald Geman* and *Dr. Laurent Younes*.

I extend my deepest gratitude to my primary advisor *Dr. Donald Geman* for his terrific mentorship in all these years. His glittering wisdom, scientific prominence and admirable farsightedness provided me the brightest north star throughout this long journey. I am humbled by his seamless intellectual and financial support, his embracing friendship and his never-ending trust in my potential. He has been and always will be an irreplaceable role model for years to come.

I am also immensely grateful to my co-advisor *Dr. Laurent Younes*, who nourished me with his vast knowledge, limitless intelligence and an unmatched rigor. He was always there when I needed, always ready to share with me the countless hours of pondering; and all this time, he was never short of illuminating the path to a solution. Without his tremendous help and guidance, it would not be possible to complete this dissertation.

Thus, I had the great privilege of experiencing the brilliant leadership and phi-

losophy of both Don and Laurent. Hoping to be worthy of their legacy, I can simply and unequivocally say they are my "founding fathers".

Secondly, I would like to express my heartfelt appreciation to *Dr. Michael Ochs*, for his comradely support, his diligent role as both my fellow author and examiner, and more importantly for his priceless expertise with bright ideas and precious data that not only realized a huge portion of this thesis, but also opened up a very exciting new chapter in my research statement.

Thirdly, I would like to thank *Dr. John Goutsias* for his constructive critique, inspiring questions and valuable comments as my defense chair. Additionally, I feel very fortunate to have taken his "Computational and Functional Genomics" class, which has been the most influential course in my graduate life while also motivating a great portion of this thesis.

I would also like to express my gratitude to my department chair *Dr. Dan Naiman* for his genuine support and interest in my progress; and under his successful management, to the entire family of the *Department of Applied Mathematics and Statistics*, which provided me a very rigorous graduate education. I am grateful to our academic program coordinator Kristin Bechtel for all her help throughout these years, and particularly to my professors *Dr. Carey Priebe*, *Dr. Jim Fill* and *Dr. Bruno Jedynak* who taught and elevated me towards a prestigious degree.

Concurrently, it was a great honor and joy to be a part of the *Center For Imaging Science* community. CIS provided me not just the most innovative and colorful re-

search environment, but also my second home here in Baltimore. I am thankful to all the faculty and members of CIS, especially to *Dr. Tilak Ratnanather* and *Dr. René Vidal* and to our dear administrators *Dawn Kilheffer* and *Maura Vonasek*.

The overwhelming period of grad-school would not be bearable without the cheerful support of friends. My special thanks go to my office-mates *Dr. Felipe Arrate* and *Bahman Afsari* for their sustained assistance and exchange of ideas in endless conversations; to *Francisco Sanchez* and *David Simcha* for their very valuable input and collaboration, to my fellow countryman *Ertan Çetingül* for his precious companionship, and many other fellow grads in CIS, who shared the same roof and dedication with me. I should also acknowledge the support from countless other friends including *Genco Güralp*, *Özge Gürcanlı*, *Burak Gürel*, *Orcan Ögetbil*, *Ender Konukoğlu*, *Atilla Yılmaz*, *Anıl Yazıcı* and *Gönenc Yücel*, who are my fellow passengers in the same long academic journey.

Unquestionably, my family has been the greatest source of support in all my endeavors, since I was born. My biggest appreciation goes to my wonderful dad and best friend *Ali Yörük* for his irreplaceable lifelong guidance, to my beloved deceased mom *İnci Yörük* for her constant spiritual presence that keeps alive the pride of fulfilling her dreams, and to my dear step-mom *Tülin Yörük* for her endless cheer and encouragement. I am blessed with the world's most caring brothers, *Onur Yörük* and *Can Yörük*, who were always there whenever I needed their companion and sense of humor. I am also very humbled by the constant love and support of my parents-

in-law *Zafer Göksun* and *Selva Göksun* and my lovely sister-in-law *Hande Göksun*.

Definitely more than anything else, I would like to thank my soul-mate and beloved wife *Dr. Tilbe Göksun-Yörük* for completing me in every aspect imaginable. Her beautiful mind, brilliant intellect and affectionate assistance have always embraced and lifted me, as I confronted the hardships of Ph.D. Every page of this dissertation bears another fond memory of her constant and amazing support, which fuels brand new chapters yet to come.

Dedication

This thesis is dedicated to my wonderful wife Tilbe.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Preface

Statistical learning is the algorithmic study of inferring knowledge about a phenomenon from a limited number of quantified observations. The term "learning" refers to the goal that the knowledge acquired should generalize and improve with experience, that is, as more examples are seen and can be analyzed. One of the earliest published papers in this field dates back to 1936, when Fisher laid out a quadratic function to optimally decide the membership of a sample, when there are records from two normally distributed alternative populations. Ever since, research in statistical learning has matured to become one of the drivers in many areas of modern science. Not surprisingly, this exciting progress owes its origin to the revolutionary arrival of computers, which launched the reign of information and made possible the

development of many novel algorithms. Now, thanks to seamless technological breakthroughs, vast amounts of data are being accumulated, posing brand new statistical challenges that constantly grow in size and complexity.

The science of learning plays a key role in domains ranging from engineering to social sciences, finance and biology. Related research in those fields has even paved the way for new disciplines concentrated on heavily studied particular problems. For example, teaching the machines how to semantically interpret images has given rise to *computer vision*, whereas the challenge of understanding cellular networks and their functions from measured molecular counts have led to *computational biology*.

Regardless of the subject matter, the general learning problem usually takes the following mathematical form: Given N independent and identically distributed (i.i.d.) realizations $\mathcal{L} = \{(x^{(1)}, y^{(1)}), ..., (x^{(N)}, y^{(N)})\}$ from the joint probability distribution of a D-dimensional random input vector $X \in \mathcal{X}$, and a corresponding random output variable $Y \in \mathcal{Y}$, what is the underlying function $g : \mathcal{X} \mapsto \mathcal{Y}$ that maps X to Y?

Here, the entries of X are called *features*, which are quantifiable measures of a phenomenon, whereas Y is the *outcome* or *label* of that phenomenon. The sought-after mapping g is estimated by some *predictor* f based on patterns inferred from the *learning set* $\mathcal{L}$. The central challenge here is to design an f as accurate as possible for new examples outside $\mathcal{L}$. This general performance is reflected by the so called *generalization error*, which quantifies the expected discrepancy between predictor's response and the true output.

Face detection in still images is a typical computer vision example of such a learning problem. In this particular task, X is a list of image features, such as raster scanned pixel gray levels or localized indicators of image discontinuities, while Y is a binary label that takes on the value "1" on images actual faces, and "0" on non-face images. Accordingly, the learning set $\mathcal{L}$ is a collection of images or image patches of both faces and background, where true labels "face" or "not face" are already assigned manually.

In the general formulation above, the learning set $\mathcal{L}$ includes recordings of the output variable Y, which guide the construction of f. Thus, the corresponding procedure is categorized as *supervised* learning. In particular, if Y takes on finitely many values, then the task is called *classification*. Face detection is an example for this, where each image is classified as one of the two outcomes, i.e. face or background. For real valued Y, the problem is referred as *regression*, in which case one accordingly fits a continuous function to the observations, for instance, to forecast the future behavior of some time series data.

Sometimes, the problem is posed without an explicit output variable Y, or the realizations for Y are absent in the data $\mathcal{L}$. In that case, the goal is to describe the associations and patterns among recorded features. Learning in such situations is therefore done in an *unsupervised* manner. Related tasks include *density estimation*, and finding statistically relevant groups of similar data points, namely *clustering*.

In all of the cases discussed above, the learning strategy mostly involves some in-

tuitive assumptions that are required to translate the overall problem to some form of tractable optimization. For example, in the case of classification, one may hypothesize normal densities for the features X, given the label Y, such that their joint distribution $p(x, y)$ is represented by a mixture of Gaussian densities. Assuming each class is equally likely, class conditional means and covariances deduced empirically from data $\mathcal{L}$ will suffice to formulate a maximum likelihood classifier $f(x) = \arg\max_y p(x|y)$ that assigns a feature vector x to the class, under which it achieves the largest probability. As in this example, the assumptions induce a series of mathematical rules and relations, namely a *model*, and in turn, the problem boils down to estimating the *parameters* of that model.

In general, we can distinguish between *discriminative* and *generative* models. Discriminative models are mostly suited to classification problems. They are aimed at producing optimal decision surfaces in the feature space that separate instances of one particular class from those of other classes as cleanly as possible. Examples for such models include linear discriminant analysis, support vector machines and neural networks. On the other hand, generative models are targeted at estimating data generating probability distributions. Learning is usually harder with such models, but their utility is broader. They can be used for understanding the underlying generative processes, validating hypotheses on variable interactions, inferring latent causes that lead to observations and classifying instances based on their class-conditional likelihoods. In this thesis, our focus will be on generative models.

Model complexity is a crucial concept in learning, and it is quantified by the number of parameters to be estimated. In fact, one can always do better at explaining data that are already seen during training simply by replacing a simple model by a more general and complex one. This is also true for the above example: switching from normal densities to mixtures of normals would result in a better classification of training instances. But, what matters is the general behavior on unseen samples, where the same trend does not necessarily apply. This fundamental question of which model to choose for learning, or how to devise the search for an appropriate model, brings us to the well known issues of *model selection*, *bias* and *variance*.

To be concrete, let us briefly explain these concepts again within the supervised learning paradigm, where we imagine several different, but equally representative learning sets at our disposal, say $\mathcal{L}_1$, $\mathcal{L}_2$ etc. Let $\mathcal{M}$ denote a model fixed a priori for our learning strategy, and let $f_k : \mathcal{X} \mapsto \mathcal{Y}$ be the predictor based on $\mathcal{M}$ that is estimated from the k^{th} learning set $\mathcal{L}_k$. Then, in simple terms, $\mathcal{M}$ is said to be *biased* for a particular input x, if $f_1(x)$, $f_2(x)$ etc. are always incorrect in their predictions of the true output for x, even when training sample sizes $|\mathcal{L}_1|$, $|\mathcal{L}_2|$ etc. tend to infinity. On the other hand, $\mathcal{M}$ is said to have high *variance* for an input x, if $f_1(x)$, $f_2(x)$ etc. deviate significantly from each other.

In this setting, the viability of $\mathcal{M}$ is reflected by its generalization error, which can be shown to decompose as a sum of bias and variance terms (similar decompositions into bias and variance also hold in contexts other than the supervised problem).

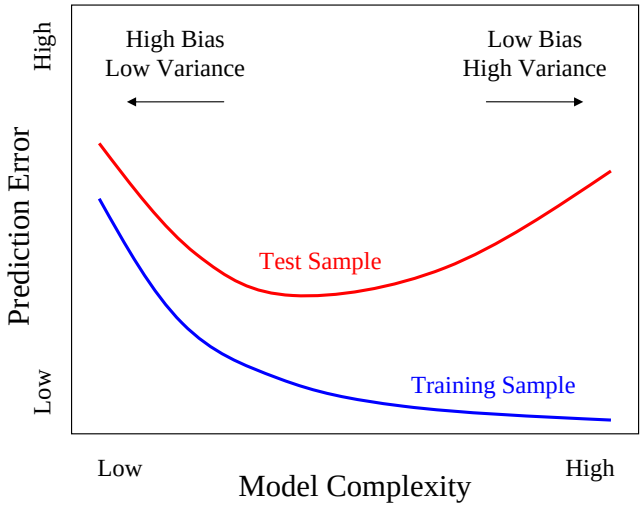

Figure 1.1: Bias-variance tradeoff in relation to model complexity

Clearly, the ideal situation is when these two components are both small. But, in general, there is a tradeoff between bias and variance, which can be controlled by model complexity (see Figure 1.1). The intuition behind this is as follows: A simple model is "rigid", thus it exhibits low variance, but tends to have a high bias. On the other hand, a complex model is "flexible", better accommodating the real situation, hence having a small bias, but in turn a high variance, since it adapts itself excessively and therefore differently to each possible data set.

The bias-variance tradeoff should be wisely handled and incorporated into one's learning strategy. It becomes especially crucial in certain circumstances. In particular, difficulties commonly arise due to

- Small N: Scarcity of samples available for training

- Large D: High dimensional feature space $\mathcal{X}$
- Complex interactions among input variables in X

which are indeed prevalent in most of the statistical problems around.

In computational biology, the "small N, large D" dilemma can reach an extreme. Difficulties in that regard are well-documented in the domain [1–4]. A typical example is the analysis of *DNA microarrays*, which is a revolutionary technology used to quantify activation levels of genes. DNA microarrays can provide simultaneous measurements for thousands of different molecule species, but the number of observed feature vectors per experiment is minuscule, mostly less than one hundred. Similarly, in computer vision, interesting problems, such as semantic scene interpretation require descriptions of high order spatial patterns among a very large number of image features. In some cases, such as representations using large collections of multi scale cues [5, 6], input dimensionality can easily go up to thousands even for images of ordinary resolution, yet compared to the length of these descriptors, training sample sizes remain relatively small.

Moreover, in these examples, variables tend to have complex interactions with each other. In biology, these are due to the sophisticated regulation in the living organism, thus, for a biologist, their discovery and interpretation is itself a core issue. Similarly, in vision, the complexity of interactions go beyond variations of illumination or geometric transformations, essentially determining object categories and their configurations in the scene.

Clearly, having only a small number of training samples for high-dimensional features make learning severely prone to high variance. When the "small N, large D" situation is further exacerbated by complex variable interactions, exact learning becomes impossible in practice [7]. Consequently, one is forced to impose strong assumptions to address the bias-variance problem.

In this thesis we are motivated by those challenges in the particular realms of biology and vision. To gain control over the variance component of learning, we broadly argue for

- Incorporating prior domain knowledge
- Introducing carefully chosen structural biases

In computational biology, making use of prior knowledge can be especially helpful. For example, when analyzing biological networks of molecules in the cell, one can identify known interactions between certain components a priori. This valuable information can be readily incorporated into the representation of dependencies between corresponding variables. Similarly, in vision, one can utilize prior knowledge about special structures of the image world, such as spatial relations of localized features and invariance to photometric and geometric transformations that preserve semantic categories.

Nonetheless, the amount of prior knowledge is rarely sufficient to circumscribe the difficulties. Consequently, the huge computational scale of model search and suscep-

tibility to high variance makes further attempts necessary to confine and organize possible explanations. Usually, this is done by penalizing model complexity, i.e., favoring simpler explanations, but learning then tends to be biased towards the absence of interactions, namely towards independence, blocking the discovery of high-order relationships among the variables of interest. We argue that it can be more effective to ease the penalties while adopting proper biases and severely restricting the search to a relatively small class of models, in which case learning complex dependencies becomes feasible due to variance reduction. Next we discuss in detail particular challenges from biology and vision, which motivate this general philosophy.

1.2 Motivations from Biology

Biological data are now being accumulated in huge amounts thanks to recent technological breakthroughs. Consequently, their analysis has become one of the most challenging and interesting statistical problems today [8]. A good portion of current computational effort is being invested to shed light on fundamental questions like how cells communicate with their internal and external environments, and how they process incoming signals and make decisions based on them. At the heart of these questions lies the intricate machinery called *gene regulation*, which is the ability of a cell to modulate its genomic activity and produce protein when needed.

Proteins are essential building blocks of any living organism and their chemical "blueprint" is contained in particular segments of the DNA, namely the *genes*. The production of proteins is carried out by reading this blueprint, which basically involves the following two important steps (see Figure 1.2):

- *Transcription*: Synthesis of intermediate molecules, called the *messenger RNA* (mRNA) from the DNA of protein encoding gene.

- *Translation*: Synthesis of proteins at specific production sites based on the sequence information delivered by the mRNA

Proteins participate in virtually every cellular process such as catalyzing enzymatic reactions, determining cellular states, mediating signaling between and within the cells, and many more. Thus, their synthesis is tightly coordinated, most extensively

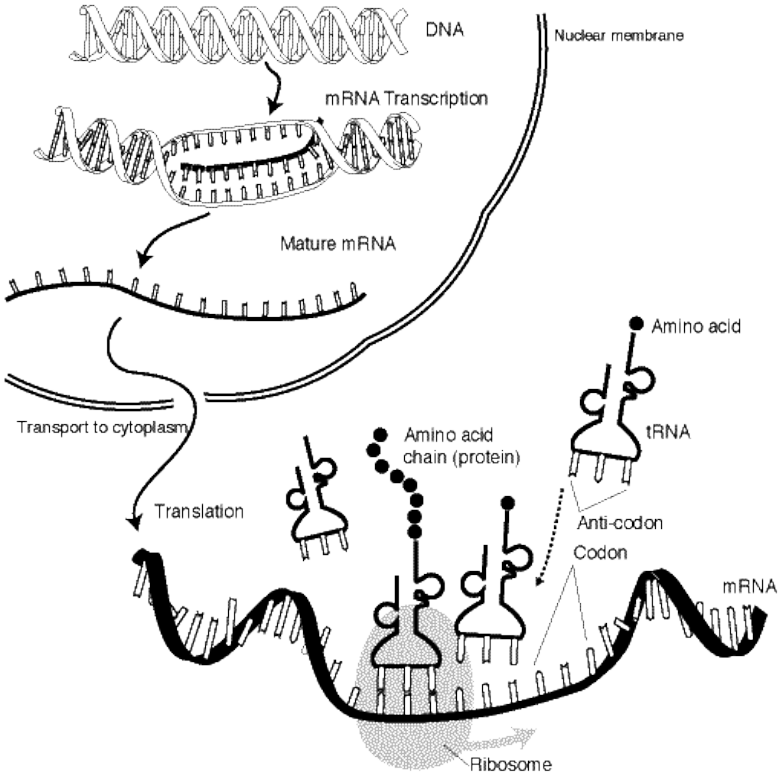

Figure 1.2: Protein Synthesis

at the transcription stage, through regulation of the responsible gene. This makes the amount of the transcribed mRNA an important source of informaion, which not only indicates the gene's activity, namely its *expression*, but also provides a valuable means to understand structural and statistical properties of cellular processes.

DNA microarrays are revolutionary tools that can achieve just this. Basically, they are arrayed series of thousands of printed microscopic spots, called the *probes*, which are short DNA sequences specific to known genes. In a typical microarray experiment, mRNA is first extracted from the analyzed cell sample, and, using an enzyme, it is

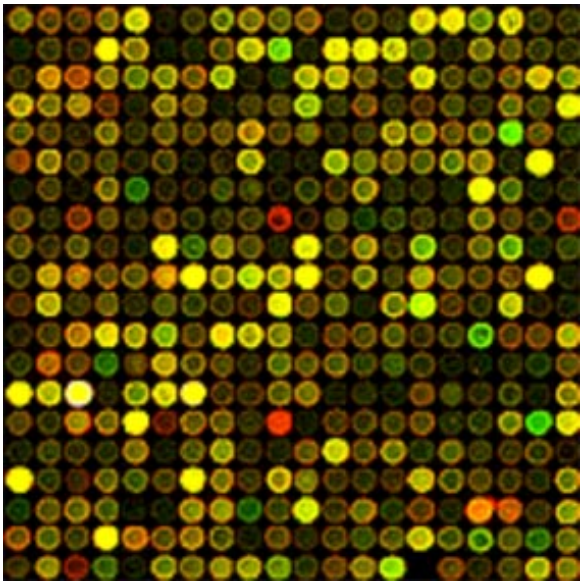

Figure 1.3: An example portion of a DNA microarray. Fluoroscopic intensity at each spot reflects the relative expression of the corresponding gene.

copied to the more stable complementary DNA (cDNA), which is also labeled with fluorescent dye. Then, this cDNA, called the *target*, is *hybridized* to the microarray slide, where target sequences bind to their complementary probe sequences. After hybridization, data are read by measuring the fluorescent intensity at each spot, which reflects the relative experimented transcript abundance for the corresponding gene, namely its expression level in the analyzed sample (see Figure 1.3 for an example). Since an array can contain tens of thousands of probes, it can measure expression levels of many genes simultaneously [9,10]. However, the number of experiments (e.g., patients) per study remains quite small, usually fewer than one hundred [11, 12].

DNA microarrays provide rich input for various statistical questions. Typical examples are gene clustering (e.g., identifying *co-regulated* genes) and *genotype-to-*

phenotype prediction (e.g., what are the specific expression patterns that determine a certain cancer type). More ambitious tasks include revealing *gene regulatory networks*, that is, elucidating rich and complex interactions among the genes.

Clearly, the huge feature dimensionality combined with very limited sample sizes makes learning from microarrays a difficult task. Even so, DNA microarrays can provide a good means to approach interesting problems, like the one discussed below, which motivates a large part of this thesis.

1.2.1 Inferring Protein Signaling Networks

Cells communicate with their micro-environment to coordinate their basic activities. This ability to perceive and respond to surrounding *signals* is very crucial for proper conduct of many cellular processes including development, tissue repair and immunity. Diseases such as cancer, autoimmunity, and diabetes usually occur when this communication is impaired. Thus, understanding cell signaling is very important in the pursuit of effective treatment.

Signaling proteins are the basic units that enable information flow between and within the cells. They can modify their behavior based on conformational changes induced by other signaling proteins, and thereupon can cause similar changes in others, hence forming *pathways* and *networks* of signaling interactions (see Figure 1.4 for an illustration).

Protein signaling networks play a central role in transcriptional regulation. They

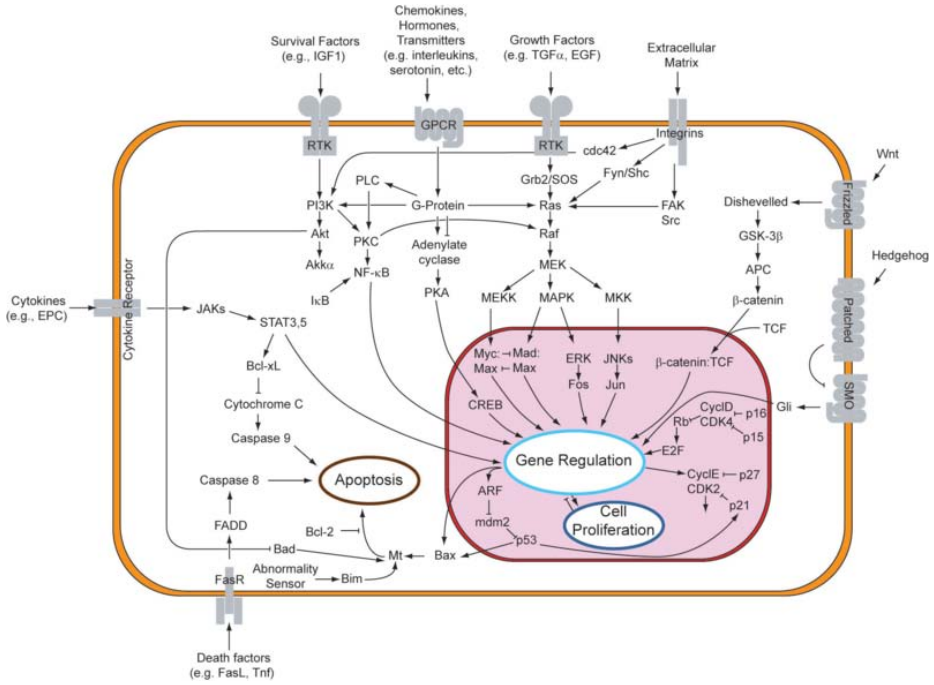

Figure 1.4: An overview of signaling networks (courtesy of wikipedia.org).

provide an efficient structure for information flow, allowing cells to alter gene expressions after sensing changes in their environment. A simplified explanation of how signaling regulation works can be given as follows: Consider a pathway from the cell surface towards the DNA. A signal gets initiated at a *cell receptor*, which is a special protein at the cell membrane, capable of receiving outside stimuli. Then, the signal propagates along causal cascades of protein interactions and eventually affects a final target protein called the *transcription factor*. This protein, if activated, binds to the DNA, and, as its name suggests, initiates mRNA transcription of a particular gene. As a consequence, the presence and strength of the arriving signal determines the expression level of this gene.

Statistical methods have been widely used to understand cell signaling [13–17], but mostly for extensively studied non-human species, called *model organisms*, and with a primary objective of uncovering structures of signaling pathways. Extensions to mammalian systems have not yielded compelling results, due likely to greatly increased difficulty posed by small sample sizes, the necessity to discover the underlying pathway topology, and limited protein measurements.

More interesting questions arise when analyzing cell signaling in the context of a human disease. In particular, being able to infer signaling abnormalities from data is of great clinical value. If such protein aberrations can be successfully identified, they can provide potential targets for therapeutic intervention. But again, without incorporating prior information, such as known interactions between signaling proteins, there is little hope for learning when only gene expressions are observed.

1.3 Motivations from Vision

The limitations and challenges in computer vision are well known and well documented. In particular, the current state of artificial vision systems in interpreting images, such as recognizing instances of generic object categories in cluttered scenes, falls well short of human performance. Consider again, for example, the problem of detecting and localizing faces in static gray level scenes. Despite intense study and major steps towards real-time processing [6, 18], all methods, whether generative or

discriminative, incur a considerable number of false positives at high detection rates, even for frontal views and with high resolution image data. Extending the same example to more detailed descriptions such as recognizing faces and facial expressions, is yet farther beyond current capabilities, and the situation is qualitatively the same in other areas of scene analysis, for example indexing medical image databases and identifying scene context.

Solid algorithmic and theoretical advances have been made in statistical learning using new tools, such as boosting [19], support vector machines [20], multi layered neural networks [21], and decision trees [8, 22]. However, achieving state-of-the-art performance with such advanced learning techniques comes at the expense of massive training, for instance exploiting huge numbers of examples in order to narrow down false positives and learn about background clutter.

This argues for a model based global strategy by which one can organize computation, incorporate prior knowledge about invariance (e.g., geometric and photometric transformations that preserve semantic categories) and can accommodate situations that are never seen during training. Still, the space of features is typically very high dimensional and the dependency structure, i.e., the interactions among the semantic tokens involved in an interpretation, are highly complex. This again leads directly to the problem of small sample structure discovery.

In that regard, we are primarily motivated by the growing body of work, in which learning, modeling and search are addressed by *hierarchical representations* of fea-

tures. The literature on this theme include research related to hierarchical matching [23, 24], hierarchical modeling [25], feature sharing [26–29], cascades [6, 30], and coarse-to-fine search [18, 31–33]. Among generative approaches of that spirit, compositional vision systems [27, 34] are designed to account for context and the hierarchical nature of the physical world, but in the expense of intensive manual modeling. Another inspirational study on hierarchical representations for object recognition involves the concept of "decomposable events" [18], where discriminative features are designed in an agglomerative and data-driven approach by recursively combining significant subsets of binary events that tend to co-occur in the target object class. Motivated by the promise of those methods, we intend to extend similar ideas to generative learning applicable to challenges in artificial vision.

1.4 Problem Statement

We argued for incorporating prior information and/or proper biases, for difficult learning scenarios laid out in section 1.1. Towards this end, we concentrate on two particular challenging questions, for which those ideas will be especially useful:

- Given scarce and noisy data of gene expressions and advanced knowledge about molecular connections, how can we formulate a robust way to infer signaling activities of unobserved proteins? In particular, how can we reliably detect protein aberrations and reverse engineer phenotypic attributes of signaling from microarrays?

- More generally, when data are scarce and high-dimensional with little prior knowledge, what is a feasible generative approach to uncover and represent existing interactions among variables? In particular, how should one design a good set of candidate explanations that are (a) versatile enough to accommodate such complex interactions; and (b) confined enough to manage the susceptibility to high variance while also enabling a tractable search strategy?

These problems are closely related in spirit, but can be treated as two stand-alone projects given their scope of applications. One is particular to cell signaling analysis, whereas the other poses the deeper question of model discovery from a more general orientation. Accordingly, we will develop them in separate segments of this thesis.

Before going into our proposed methodology, we first need to lay out the general mathematical framework to operate with. Note that the questions stated above, as well as the related ones motivating this thesis, have common technical aspects. For example, when available, prior information is in the form of known associations among components, such as interacting proteins. Secondly, existing complex dependencies are, or can be reasonably assumed to be, hierarchical and decomposable into interactions within smaller subsets of components. A third common property is the involvement of latent elements. These are either actual but unobservable components of the underlying generative process, or their hypothesis is convenient for efficient representation. A rigorous framework to properly handle each of these points is provided by *graphical models*, which we briefly review next.

1.5 Graphical Models

Graphical models are probabilistic models that can compactly encode multivariate distributions via an annotated graph of variables. In a graphical model, each node stands for an individual random variable, and the (lack of) edges encode conditional independence assumptions. That is, whenever an edge is absent between any two variables, then these are rendered conditionally independent given the rest of the variables. This *Markov property* enables a very effective representation of complex dependencies via conjunctions and hierarchies of local interactions.

To better elaborate on this, consider the following classical example due to [35] (see Figure 1.5). Here nodes "Cloudy" (C), "Rain" (R), "Sprinkler" (S) and "Wet Grass" (W) represent binary random variables, each being either true or false. As the graph depicts, the event "Grass is wet" (W=true), can have two possible causes: either the sprinkler is on (S=true), or it is raining (R=true). Similarly, the usage of sprinkler as well as the rain are both affected by the weather being cloudy or not. Then, by the chain rule of probability the joint probability of all the events is given by

$$P(C, R, S, W) = P(C) \times P(R|C) \times P(S|R, C) \times P(W|C, R, S).$$

Using conditional independence relationships we can rewrite this more compactly. For instance the event "Grass is wet" is conditionally independent of the weather being cloudy, given that it rains, or that the sprinkler is on. Thus, we can reduce

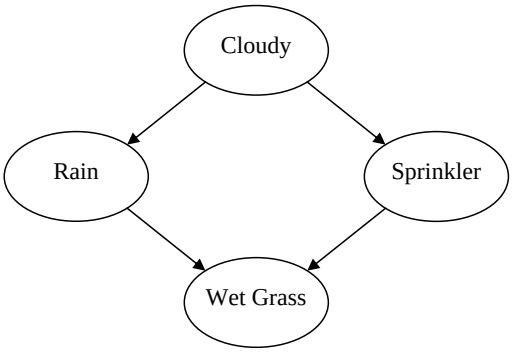

Figure 1.5: A simple graphical model on 4 nodes.

$P(W|C,R,S) = P(W|R,S)$ and similarly $P(S|R,C) = P(S|C)$, to obtain a simpler representation

$$P(C,R,S,W) = P(C) \times P(R|C) \times P(S|C) \times P(W|R,S).$$

Although savings in complexity are small in this particular example, they can be drastic in general: Over D variables, the full joint would require $O(2^D)$ parameters, whereas using a sparsely connected graph with at most $K < D$ connections, (i.e., direct dependencies) per node, the corresponding graphical model would only need $O(D2^K)$ parameters for its factored representation.

Thanks to this interpretable interface, graphical models can readily incorporate structural biases as well as prior knowledge about variable interactions; they can accommodate hierarchies and latent components, and with their well understood algorithmic foundations they can be efficiently learned. These appealing properties have made them common statistical tools in a variety of fields ranging from physics,

biology to speech and image analysis. Many of the established statistical learning methods fall within the graphical model formalism. Examples include mixture models, factor analysis, hidden Markov models and Kalman filters.

Depending on the properties of the graphs employed, there are two common types of graphical models: those with undirected edges, called Markov random fields (MRFs) or Markov networks, and those with directed edges, namely Bayesian networks (BNs) or belief networks. In this thesis, we will be mainly interested in networks of hierarchical nature, which involve "causal" relationships and therefore can be represented with directed acyclic graphs (DAGs). Consequently, our focus will be on the use of directed graphical models, namely Bayesian networks, in particular those with hidden variables. We give below a brief review on how learning is generally achieved with Bayesian networks. A detailed tutorial can be found in [36, 37].

1.5.1 Learning with Graphical Models

In the context of graphical models, learning may refer to estimating the underlying graph, namely the dependency structure, or the parameters of local conditional distributions; and occasionally both. Another important distinction can be made based on whether or not all the represented variables are observed. The latter case may be due to limitations in the data acquisition process leading to incomplete data, as well as due to the invention latent variables for modeling purposes. Based on observability and the knowledge about the graph structure, there can be four distinct

Table 1.1: Typical learning algorithms for different graphical model settings

	Full Observability	Partial Observability
Known Structure	Closed Form	EM
Unknown Structure	Local Search	EM + Local Search

learning scenarios as listed in Table 1.1.

In the case of known topology (first row), one only has to estimate the parameters. This is called model identification and corresponds to finding parameter values that maximize the log-likelihood of data. Since the likelihood decomposes into conditional distributions of individual nodes given their parents, model identification is mostly straightforward in Bayesian networks. With full observability, it is done in closed form. If there are missing data, or the model involves latent variables, parameter estimation is achieved iteratively by expectation maximization (EM) algorithm [38] or its stochastic variants [39].

In the case of unknown topology (second row), one has to learn the underlying connectivity simultaneously with associated parameters. Usually, competing models are scored according to some metric and the one with best score is eventually chosen, or a posterior distribution over the models/parameters is returned. By introducing priors for possible structures and/or parameters, the score is usually evaluated as the posterior probability of the underlying graph given the recorded observations. Setting the priors appropriately, complex models are penalized more to control variance and

thereby to avoid over-fitting the data.

If all variables are observable, the Markov assumption allows the model score to decompose into a product of local terms, such that the search through the large space of possible architectures can be efficiently done locally, i.e., by adding and removing edges one by one. On the other hand, if there are latent variables, the marginal likelihood of observations becomes an integral over the hidden part, thus it no longer decomposes into local terms. In this hardest case, the model score can be theoretically approximated [36, 40] by Akaike Information Criterion (AIC) [41], Bayesian Information Criterion (BIC) [42] or similar other metrics defined for different situations. These quantities usually involve maximum likelihood parameter estimates given the current structure, such that one has to run EM at every step of the search, which may be computationally expensive. A faster alternative is to embed the search for optimal topology into the M-step of EM, resulting in the so called structural EM algorithm [43].

1.6 Proposed Methodology

Using the general ideas of incorporating prior information and/or proper biases, and the rigorous formalism provided by probabilistic graphical models, we now propose two new and robust generative approaches for the challenging problems stated in Section 1.4:

(a) A comprehensive statistical model for analyzing cell signaling networks from microarray expression data

(b) A restricted class of nested forest models designed to learn and represent complex dependency structures

Our first method is particular to cell signaling networks, while the second one addresses the more general endeavor of model discovery, with broader potential for applications ranging from classification to density estimation and clustering. Both methods are proposed and designed for robust learning from small samples, and both involve directed graphical representations with only terminal nodes as observable variables. The assumption of latent structures is due to unmeasurable actual components, namely protein activities, in the case of (a), whereas in (b) it provides the means to effectively represent high-order dependencies. Accordingly, our methodology and findings can be detailed in the following two segments.

1.6.1 A Comprehensive Statistical Model for Cell Signaling

In Chapter 2, we present our comprehensive statistical model for cell signaling, with an emphasis on inferring unobserved activities of signaling proteins from only microarray data of their pathway targets. We argue for fixing a core signaling diagram a priori, based on known protein interactions and documented pathway structures in the biology literature. This serves as the underlying directed graph of our latent variable model, which involves hidden protein activities as internal nodes and gene expressions at the terminal nodes, i.e., target leaves. Thus, from a learning standpoint, the method here falls within the "known structure, partial observability" category of Table 1.1, where, for inference, we use a stochastic variant of EM, namely stochastic approximation EM (SAEM) algorithm [39].

Our model has several appealing properties and novelties. It has a limited complexity due to parameter sharing among similar types of signaling interactions. This provides robustness against network size and applicability to different studies. We also account for the realistic concept of *cell heterogeneity* and take a *multi-level* approach, where we incorporate cell-to-cell differences within the experimented tissue and study the problem comprehensively in multiple stages of the process leading to observations. In particular, we consider a Bayesian network model at the *cell level*, which we base on the core signaling graph. Then, at the *tissue level*, we represent

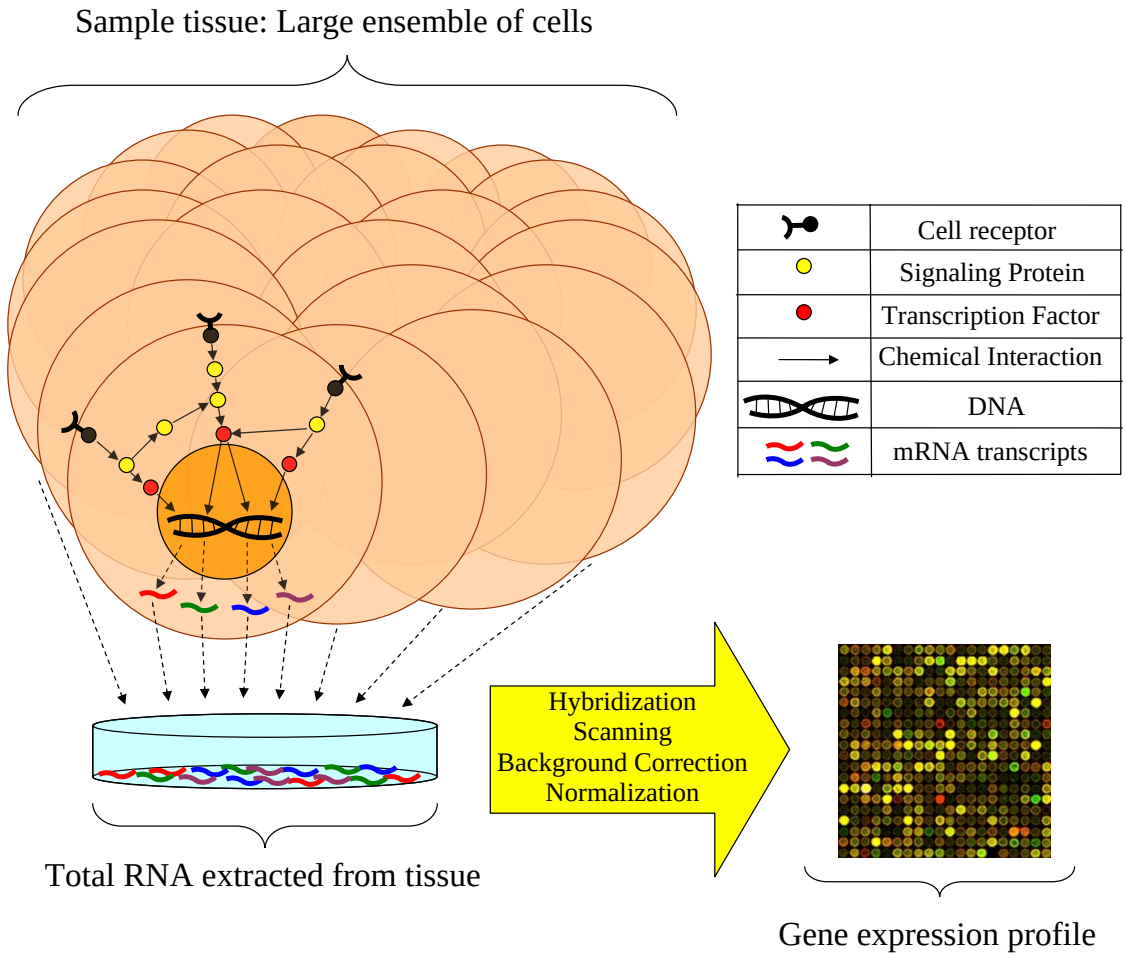

Figure 1.6: An overview of data generation at multiple stages from signaling process to microarray measurements, which motivates our multi-level model.

measurements as functions of ensemble averages over experimented cells. Finally, at the *population level*, we account for patient-to-patient differences by also treating phenotypic attributes as random variables. A broad overview of both the signaling and measurement processes, which motivates our approach, is given in Figure 1.6.

With the goal of identifying individual protein abnormalities as potential therapeutical targets, we apply this comprehensive model to the RAS-RAF signaling

network using a breast cancer study with 118 cancer patients. We can achieve robust statistical inference given the large proportion of missing data, small sample sizes and known limitations of the microarray measurement process. Our findings demonstrate reproducibility even with small samples, robustness against varying amounts of measurement noise, invariance to realistic modifications of the network topology, and most importantly, the ability to accurately recover hidden receptor statuses.

1.6.2 A Nested Family of Latent Variable Forest Models

In Chapter 3, we analyze the deeper question of model discovery from undersampled, high dimensional data, and in the presence of complex variable interactions. For such challenges, we construct a nested class of hierarchical latent variable forest models, for which we can lay out an efficient learning algorithm both for parameters and dependency structures. Our models have various potential applications, including density estimation, maximum likelihood based classification, clustering and dimensionality reduction.

Latent variables are a major ingredient of our design, which, in this case, are invented to encode dependencies. In particular, they are arranged hierarchically in binary, balanced and Markovian tree structures to represent joint multivariate distributions of observable variables at the terminal nodes (see Figure 1.7 for an example

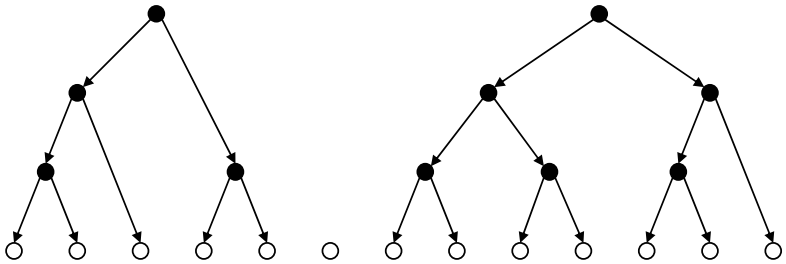

Figure 1.7: An example dependency structure from our proposed model class. White nodes (leaves) are observable variables, whereas black ones are latent.

graphical structure employed in our models). In that regard, the overall process of learning corresponds here to the hardest case, namely "unknown structure, partial observability" of table 1.1.

The model class we introduce is severely restricted with tailored structural biases, such as tree topologies that are binary and decomposable into smaller trees of comparable size. This nested design allows a well principled and greedy model selection strategy, which is based on sequential aggregations of pairs of pending substructures. This corresponds to a stepwise pursuit of significant dependencies, starting from detected pairwise correlations between observed variables, and leading to more complex interactions among larger and larger subsets of features. In this way, we can manage a feasible learning strategy with low variance in the small sample context, while retaining the ability to recover rich dependencies that may exist.

In particular, our learning algorithm has two components: Maximum likelihood parameter estimation, namely *model identification* under a fixed structure, which can be done with EM in an efficient and recursive formulation; and structure learning,

namely *model selection* realized as local search moves exploring candidate fusions of current substructures. More precisely, each such fusion replaces two independent laws by a joint distribution implemented for the combined set of terminal variables. This new joint distribution and thereby the corresponding merge of structures is justified and selected based on the Bayesian information criterion (BIC), which is the net gain in data likelihood but penalized with the additional complexity and the log of sample size. Consequently, through sequential discovery of the most significant interactions, which progressively get more and more complex, our algorithm achieves a nested refinement within the proposed family of forest structured latent variable models, where a bounded parametric growth is maintained.

We present the proposed model class and the corresponding estimations with two particular parametric choices: binary variables with local conditionals given as Bernoulli distributions, and real-valued variables with linear Gaussian interactions. In both scenarios, we give detailed proofs for *parametric identifiability*, which is an important property for inference to be possible, requiring that different parameter values must generate different probability distributions over the observable variables. Exploiting the tree topologies, we provide, under both parametric cases, explicit formulations for exact inference by a dynamic programming approach based on the Belief propagation algorithm, and lay out the details of EM parameter estimation.

We entertain various applications of our model family as described in Chapter 4. Our experiments include density estimation and classification of handwritten digits,

as well as predicting phenotypes from microarray data (e.g., "cancer" vs. "healthy" or different subtypes of cancer). Using our generative approach, we can achieve performances comparable to, and in some cases even better than the state-of-the-art discriminative methods.

1.7 Related Work and Our Contributions

Since the literature overlap is limited, We find it more convenient to discuss the previous work leading to this thesis in two segments, again corresponding to our two proposals, one for the analysis of cell signaling networks, and the other on latent variable model discovery.

1.7.1 Modeling Signaling Networks

Cell signaling networks have a central role in transcriptional regulation, thus, their analysis is key to understanding many diseases and the development of possible targeted therapeutics. Despite the recent arrival of technologies providing expression data, protein signaling has already been studied in wide variety of ways. Attempts that address this problem from a statistical standpoint are rather new. Among those, Bayesian networks have gained considerable interest, due to their well understood algorithmic foundations, their ability to handle measurement noise, to describe the process from locally interacting components, and more importantly a characterization that complies with the notion of *causal influence* [36, 44].

Friedman *et al.* introduced the first principled work on applying Bayesian networks (BNs) to gene expression data [13]. They applied their method to cell cycle expression patterns from *S. cerevisiae* and were able to draw biologically relevant conclusions. This established probabilistic graphical models as a promising tool for in-

ferring biological networks. Subsequently, extensions of this framework were studied, with broader applications to identifying co-expressed genes, clustering genes of similar function and elucidating gene regulatory networks from expression data [45–48]. In those studies, the common goal is the discovery of existing causal relations between actual physical components, namely the connectivity among a small number of genes, as well as those between proteins and genes. Statistical validation is done within the standard Bayesian network formalism by giving confidence levels on the discovered connectivity or evaluating the biological significance of the hypothesized structure with Bayesian scoring metrics.

Similar probabilistic approaches have attempted to gain further insight about the causality of signaling and regulatory pathways by analyzing perturbations of genomic activity [49, 50]. Alternatively, Bayesian methods have been applied to estimate gene regulatory networks from microarray data [14], and from microarray data coupled with biological information such as protein-protein interactions [51].

In more recent studies, signaling networks on small numbers of proteins have also been reconstructed from limited measurements of protein state and abundance, such as from flow cytometry [17], or from prior data on beliefs of connectivity [52]. Concurrently, the utility of Bayesian networks are further extended beyond traditional physical connections, for example, by the inclusion of the effects of clinical variables on outcome, while still relying on molecular data [53]; or by viewing the underlying mechanism from a phenomenological perspective, where direct molecular causative

agents are abstracted away, but predictive relationships between measured variables are retained [15, 16].

Other approaches to creating robust models have also been attempted. Ordinary differential equation (ODE) models, such as modeling of ERBB signaling response [54], can capture great complexity, but they rely on large numbers of poorly determined parameters. This can limit their verifiability, since large ranges of parameter values on many components must be explored. For inference on larger cell signaling networks, a number of alternative methods have been attempted. Matrix factorization has been used to determine activity on components of networks or in biological processes from microarray data, such as through Network Component Analysis [55] and other methods reviewed in [56].

In this previous body of work, the statistical network modeling approach, in particular, graphical Markov models, such as Bayesian networks, has prevailed due to the promise for analyzing large-scale systems and for identifying specific nodes as optimal therapeutic targets. Yet, in most of these studies, the applications are limited to model organisms with small sized networks and with a primary objective of uncovering connectivity rather than inferring protein aberrations. Extensions to mammalian systems have not yielded compelling results due to small sample sizes, the necessity to discover the underlying pathway topology in the absence of prior knowledge, and limited protein measurements *in vivo*.

In contrast to those studies, our model is intended for the analysis of cell signaling

at larger scales and with an emphasis on estimating activities of unobserved signaling proteins from transcriptional changes that can be measured routinely and globally. Our methodology differs from the majority of the previous work in that we incorporate prior knowledge about biological wiring diagrams and anchor our Bayesian network model to a fixed topology, while sharply reducing model complexity by parameter-sharing.

Our other significant contribution is that we introduce an ensemble of cell models that captures biological heterogeneity. In particular, we consider the experimented tissue as a large assembly of individual cells, each comprising a Bayesian network. Recent evidence on TRAIL[1] induced cell death supports this assumption suggesting that variability in protein concentrations between even clonal cells can lead to phenotypic variation that homogeneous models cannot address [57]. Finally, our approach is more comprehensive than previous attempts, involving a realistic multi-level approach with different statistical constructions at the cell, tissue, measurement and population levels.

1.7.2 Latent Variable Models

The use of latent variables in explaining complex phenomena is well recognized in a variety of applications. In particular, latent tree models, i.e., graphical models on Markovian tree structures with observable variables at the leaves, have gained

[1]TNF-related apoptosis-inducing ligand: a protein functioning as a ligand that induces the process of cell death called apoptosis

considerable attention due to their tractability. Thus, the relevant literature is vast, and we give here the main directions of research in this area.

A simple version of latent tree models was first introduced by Lazarsfeld and Henry for discrete data [58], where observable variables were represented as conditionally independent given a single latent variable on K possible discrete states. Denoted *latent class* (LC) models, these have the same graphical structure as the naive Bayes classifier, except that the root, which is latent in the former is actually the observed class variable in the latter.

The potential of such latent tree models as useful Bayesian networks was first pointed out by Pearl. In Section 8.3, of his book [59], he studied the recovery of latent tree structures on binary and Gaussian variables from their pairwise statistics. The idea relies on the fact that the correct tree configuration on four observable and two latent variables can be uniquely decided from relationships among pairwise correlations of the observable ones. But this argument requires precise knowledge of the correlations and that the underlying true distribution should be tree decomposable.

More practical hierarchical extensions of these ideas have later been introduced. Connolly proposed the first algorithm for learning hierarchical latent trees [60], which inspired the field but lacked a firm statistical framework. The algorithm constructs a latent tree model using mutual information based similarities between groups of variables, while latent components are learned with a conceptual clustering algorithm [61] rather than EM.

Recently, Zhang explored LC models in a more principled way, and under the terminology *hierarchical latent class* (HLC) models [62]. In this approach, cardinalities of internal latent variables are simultaneously estimated while the search through possible structures is executed in a greedy fashion. Model discovery is laid out as a series of local moves (e.g., introducing one hidden node or replacing an edge), which is accelerated with heuristic approximations in a subsequent paper [63]. A more recent extension of this work is applied to approximate inference on Bayesian networks (BN), where the strategy is to learn a latent tree model from samples generated from a ground truth Bayesian network [64]. Though computationally expensive in its learning phase, HLC models provided advantages in statistical inference thanks to their tree structures as compared to more general DAG representations.

In order to reduce running times, alternative methods have been proposed, which narrow down the combinatorial scale of model selection by confining the class of candidate structures. For example, in a recent work on HLC models [65], the authors restrict the search to binary tree structures[2], and employ the recursive layerwise learning approach previously applied to deep belief networks [66]. With notions from agglomerative clustering, their method performs sequential fusions of variables (latent or observed) using mutual information as the metric for node similarity. But with binary valued variables represented on binary trees, which is mostly the case in their applications, the method does not guarantee parametric identifiability, which makes

[2]In that regard, the authors' particular construction and EM based learning algorithm looks very similar to ours, but our method has been developed independently and in a more principled way, with substantially different applications in mind.

their similarity measure an unreliable choice when computed between latent variables.

Another recent study [67] points to this issue and the general lack of consistency of HLC models, and suggests relaxing the convention of placing observable variables at the leaves. In particular, by allowing some internal nodes to be also visible, the authors manage to limit the number of hidden nodes and provide consistency, which again holds when data are generated from their proposed class of models. Stability of learning is also addressed in the context of EM, especially when the class of explored models is rich. Attempts to alleviate relevant problems like large computational cost and getting trapped in local optima, have led to development of algorithms like structural EM [43] and information bottleneck-EM [68].

Reconstruction of hierarchical latent trees is a common task in probabilistic data clustering and mixture representations. Although related studies (e.g., [69, 70]) consider leaf nodes as data points rather than individual observed variables, their ties to generative latent tree models become obvious when the roles of variables and samples are transposed. Exploiting this, Kemp and Tenenbaum [71] proposed a general setting that compares structures of many different forms including latent trees and various others, according to their merits in organizing or describing a given data set. In the case of latent trees, they use Gaussian representations and a divisive, rather than agglomerative, approach. Similar intersecting work, though occasionally not probabilistic in nature, is found in phylogenetic studies such as [72, 73], where unknown evolutionary trees of existing species are inferred from their DNA or protein

sequences (see [74] for a thorough review).

Our proposed family of models have several important differences and advantages when compared to related previous work. First of all, our approach is primarily motivated by the central challenges confronted by a variety of current statistical problems, where samples for learning are limited, feature dimensionalities are high and interactions among the variables are rich. Thus, our core objective is to introduce proper biases to make learning robust, which we achieve by severely restricting the model space of interest to a relatively small nested class. In this way, we gain control over the complexity and implement a feasible search strategy.

Secondly, unlike most of the methods discussed above, we thoroughly analyze and establish identifiability of our models and thereby we can lay out our structure learning algorithm rigorously. In particular, when the latent tree is constructed in an agglomerative way, which is the usual bottom-up approach, previous methods mostly perform the grouping of variables based on some form of a statistical distance or a similarity metric. In some cases, this metric is mutual information, in others more heuristic choices are used. As the hierarchy grows, these distances between latent variables are needed in the construction, and the quality of their estimation heavily relies on the stability of inference, and therefore on the identifiability of model parameters. But this latter important property has been rarely analyzed and in some cases it is not satisfied. In contrast, here we prove identifiability for each of the particular parametric choices we entertain, and guide our model discovery directly by

the likelihood gains achieved in representing data at the leaves.

Thirdly, in most of the previous studies, the merits of the generative approach are not assessed in a comprehensive way. Common quality metrics are limited to the relevance of variables that are grouped together from real data, running times of the algorithm, and the estimated accuracy of structure recovery from simulations. Usually the first property is judged in a somewhat ad hoc way, whereas, in some cases, the latter two criteria remain the only means of validation. Moreover, extensions of these generative models to more challenging applications, for example, detecting generic objects in cluttered scenes, remain to be explored (except possibly for compositional non-Markovian models [27] that require intensive manual interference). As a matter of fact, methods that employ latent hierarchies and target vision problems, are currently mostly discriminative (e.g., [75]). In our case, we accommodate such applications within our scope of generative learning, and we report experiments in maximum likelihood classification of handwritten digit images, phenotype prediction from microarray profiles, as well as density estimation and reconstruction of digit shapes.

Chapter 2

A Comprehensive Statistical Model for Cell Signaling

2.1 Introduction

A central component of the cellular machinery is the set of protein signaling networks, which permit a cell to sense its chemical micro-environment and respond by altering metabolism and gene activity. Signaling networks comprise interacting signaling pathways, with each pathway containing a number of individual signaling proteins. These are the building blocks of the information flow within and between the cells. They can modify their chemical behavior based on conformational changes induced by other signaling proteins, and thereupon can cause similar changes in others, hence realizing a cascade of chemical interactions namely signal transduction.

Usually, this occurs in the cytosol of the cell via chains of phosphorylation. In that case, the components involved are kinases, which are proteins capable of adding a phosphate group to other proteins. After the interaction, the phosphorylated protein undergoes a conformational change and switches on its own kinase activity, which leads to a similar reaction with another target protein. In addition to kinases, there are also phosphatases that remove phosphate groups and thereby reverse the signal from a kinase and shut down the activity of their targets. Also, for certain signaling proteins, activity is generated by cleavage of a parent protein or dimerization, which is especially common for receptor tyrosine kinases that reside on the cell membrane and respond to cues outside the cell, such as hormones or growth factors.

In a typical case, the signal eventually targets a transcriptional regulator (i.e., a transcription factor or co-factor), activating or suppressing its binding to DNA, which directly leads to modifications in gene expression. Consequently, the transcribed mRNA, which can be measured by DNA microarrays, bears information about the data generating mechanism of underlying signaling processes that are currently not directly observable to the extent and scope we intend to analyze here.

As discussed in Chapter 1, microarray data comprise very small number of samples for learning, as compared to their large input dimensionality. Moreover, they are an outcome of complex and generally hierarchical molecular relationships; and they involve measurement noise combined with a certain extent of randomness inherent to biological interactions. Consequently, graphical Markov models, have become the

prevalent statistical approach for elucidating signaling networks from gene expression data. In particular, Bayesian networks have gained considerable interest lately, since they can comply with the notion of causal influence, accommodate complex hierarchical structures with a reduced complexity invested to represent local interactions. They can also provide an intuitive interface to incorporate prior domain knowledge, and a rigorous framework to infer and identify nodes as optimal therapeutical targets. Not surprisingly, there is already a large body of related previous work on the use of Bayesian networks for analyzing cell signaling networks (see Chapter 1 for a review).

However, despite their early promise, few major insights have emerged from such modeling efforts, at least for mammalian data. It is likely many of the shortcomings are due to the high dimensionality of the data and, in the case of reverse engineering regulatory networks, from the necessity of learning both the underlying connectivity and estimating the corresponding statistical parameters. As a result, methods designed to reduce what needs to be learned from data by incorporating prior knowledge have come into use. They are even more indispensable when, like in the present study, small sample sizes come in combination with a large proportion of unobservable components in the process of interest. In particular, in the work described here, in order to apply graphical Markov models to learning signaling networks, we utilize existing knowledge about biological wiring diagrams as well as sharply reduce complexity by parameter-sharing. In addition, we account for cell heterogeneity by modeling the observed expression data as cell averages.

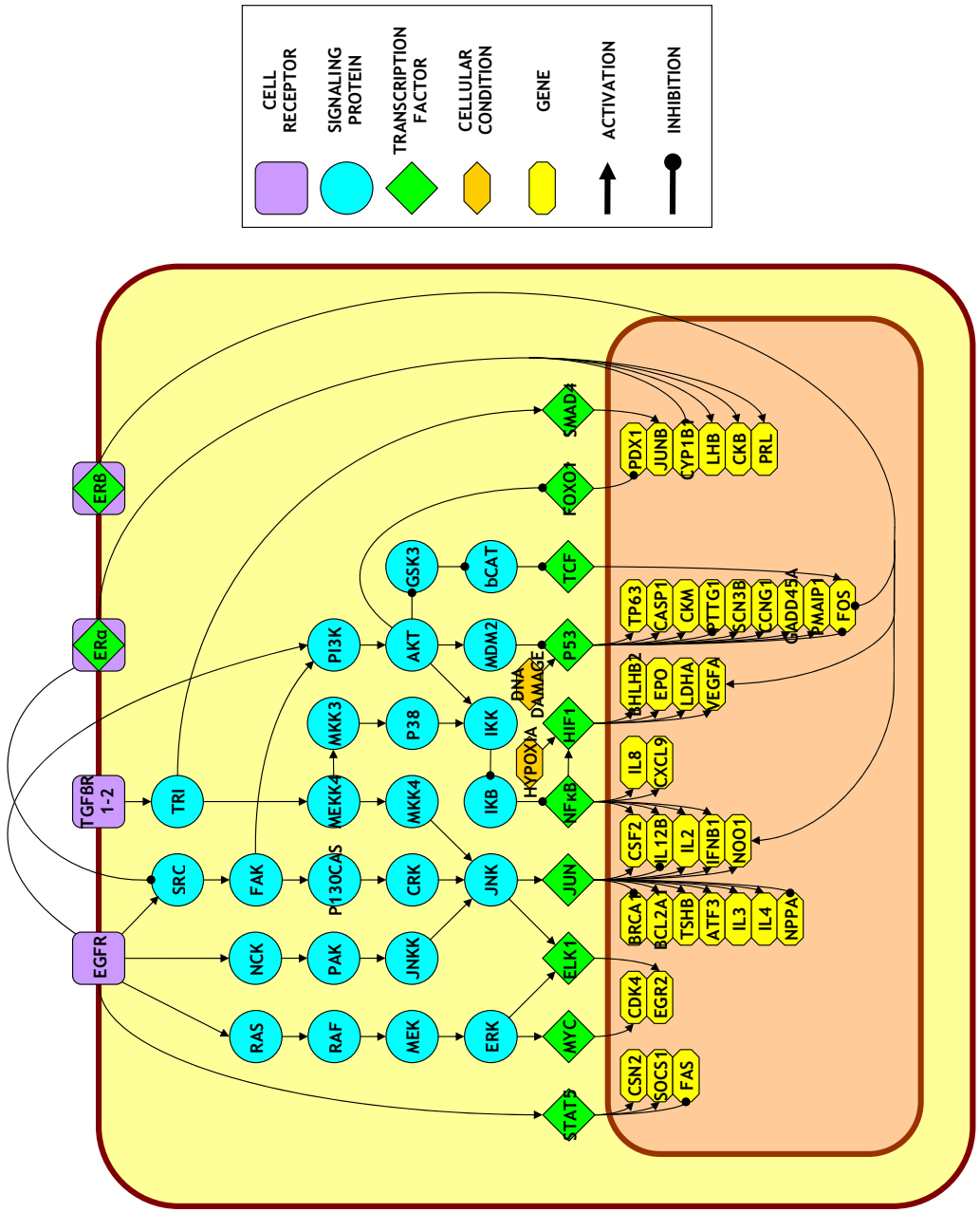

Figure 2.1: Graphical representation of the signaling network of interest. Cell receptors (purple rounded squares) and cellular conditions (orange hexagons) are roots on top of the hierarchy as initiators of downstream signaling. Signaling proteins (blue circles) followed by transcription factors (green diamonds) are in the middle whereas genes (yellow octagons) are leaves appear at the bottom as final targets of transcription. Causal interactions are depicted with arcs directed from parent to child; arrow and round heads stand for activation and inhibition, respectively.

2.1.1 Pre-defined Wiring Diagram

Unlike standard Bayesian network approaches, which attempt to learn a wiring diagram in addition to statistical parameter estimates, we begin with a defined core topology, thus eliminating the combined problem of insufficient sample size and of hidden components for determining parameters for our statistical models.

A number of the core pathways of protein-protein interactions have been detailed, especially those affecting disease, for example in cancer studies [76, 77]. Since these pathways play critical roles across many organisms, there is a substantial knowledge base [78]. For any given system, the core pathways need to be modified in terms of specific cell types, which is presently best done through review of the literature [53]. In this way a core signaling network can be created for a system of interest, with the pathways considered critically linked to transcriptional regulators.

More specifically, studies on mutation in breast cancers have verified driver mutations of key signaling components in multiple pathways that lead to breast cancer development [79]. Both the RAS-RAF proliferation pathway and the PI3K cell fate pathways have multiple driver mutations, suggesting these are excellent targets for studies aimed at developing a method suitable for identifying targets for therapeutic intervention. With such applications in mind, we constructed a network based on the core signaling processes in breast cancer. The resulting network is shown in Figure 2.1, with a hierarchical layout, where cell receptors (rounded squares) are on top as initiators of signaling; signaling proteins (circles) and transcription factors (diamonds)

are in the middle; and genes (octagons) are at the bottom as final targets.

We then identified a public domain microarray data set from a breast cancer study that included phenotypic information on receptor status [80]. This data set was collected using the Affymetrix U133A GeneChip and deposited in ArrayExpress (TABM158) [81]. We annotated the network in Figure 2.1 for the targets of the transcription factors from TRANSFAC Professional v11.4 [82] using our annotation pipeline, associating Affymetrix probes with Unigene clusters for gene identification [83]. These data will be used to learn the parameters of our Bayesian network and to validate the learned model by deducing the status of upstream signaling proteins in the form of probabilities of activation, comparing the estimated activation levels with ground-truth obtained from the clinical status measurements provided in [81].

2.1.2 Multi-Level Generative Process

Applying Bayesian networks directly to the graph in Figure 2.1 is not straightforward for several reasons. First, this ignores an important component of the data acquisition process, which is that the measured transcript levels are averaged over large ensembles of cells. Taking this into account in the model induces notable differences compared to what would correspond to a single cell model. In a proper tissue-level model, each observation arises from a large group of networks, each representing a cell. Second, the status of the signaling proteins is not observed. The only observed variables are tissue-level (hence cell-averaged) gene expression levels.

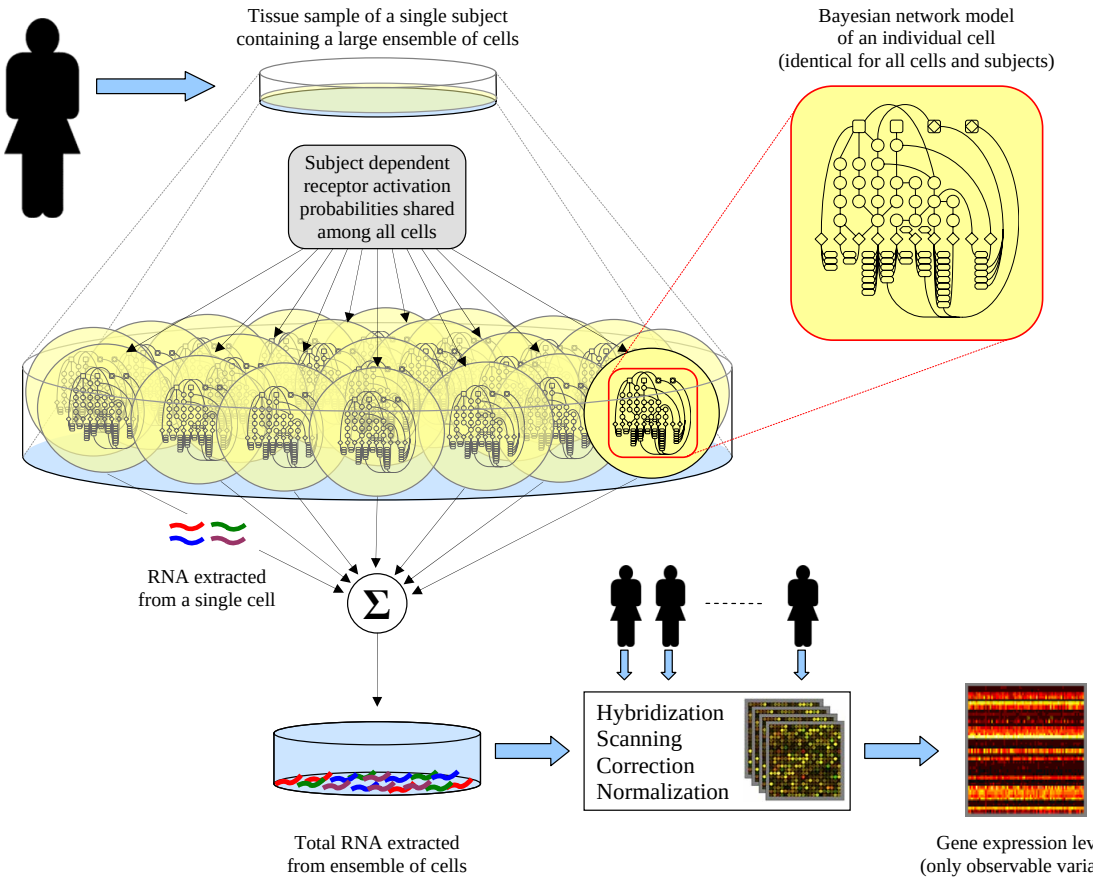

Figure 2.2: Illustration of a microarray experiment. A tissue sample obtained from a test subject is assumed to contain a large ensemble of cells. Signaling in each cell is modeled by the same Bayesian network that generates gene specific mRNA, independently of other cells given the patient's phenotypic receptor activation probabilities. The mRNA accumulated from all cells is processed through hybridization, scanning etc. to yield the final gene expression readouts, which constitute the only observable variables. Thus, the overall process motivates our multi-level approach and the assumption that measured gene expression levels are proportional to their single cell conditional expectations given patient-dependent root activation probabilities.

Despite the averaging and hidden variables, we are still able to predict the receptor status given the observed transcript levels.

Our model is organized on two levels, the first one incorporating cell-dependent variables, and the second one including factors that are common to large cell assemblies (tissues), but are subject-dependent. A detailed overview is presented in Figure 2.2.

At the cell level, we model signaling pathways as Bayesian networks in which the information is flowing from receptors (which constitute the roots of the networks to which are added certain cellular conditions, such as hypoxia) to genes. This process is assumed to be working within each cell, independently of the others. With an additive noise component, a gene expression measurement is modeled as the logarithm of a linearly increasing function of total gene-specific RNA abundance summed over a large population of cells. Final transcript readouts constitute the only observable components in our model.

The parameters of the Bayesian network at the cell level are assumed to be identical within each subject. This implies that conditioned on the subject, the measurements stem from sums of independent and identically distributed random variables. Most of these parameters are also assumed to be identical across subjects, with the exception of the cell receptor activation probabilities. These probabilities are subject-dependent and assumed to be randomly generated. Putting things in a generative order, we model the process leading to a micro-array measurement as the following

sequence of operations, performed independently for each subject:

(i) Specify the receptor activation probabilities. These are shared by all cells in the analyzed tissue.

(ii) For each cell, let the gene expression be obtained from the state of the terminal nodes in a Bayesian network that models the signaling pathway.

(iii) For each gene, define the total expression to be the sum of the gene expressions over a large population of cells.

(iv) The final expression measurement is modeled as the logarithm of a linearly increasing function of this total expression with some additive observation noise.

2.2 A Comprehensive Model

We analyze the signaling process and its measured outcomes in multiple levels, incorporating biological heterogeneity in the tissue and subject dependent phenotypic components. Next we elaborate each of those levels in detail.

2.2.1 Individual Cell Model

Interacting signaling pathways of an individual cell are modeled as a Bayesian network over a pre-determined directed acyclic graph $\mathbb{G} = (V, E)$, where V is the set of nodes (or vertices) and E is the set of oriented edges. The graph used in this paper is depicted in Figure 2.1. Some nodes $v \in V$ represent a protein which participates in signal transduction, namely a cell receptor, intermediate signaling protein or transcription factor (TF). Other nodes stand for a cellular condition, such as DNA damage and Hypoxia, and the terminal nodes (those with no children) represent genes, the final targets of signal transduction.

A directed edge $(u, v) \in E$, from u towards v $(u, v \in V)$, represents a potential functional interaction between u and v. Each such edge is labeled with the type of regulation, either activating (up-regulating) or inhibitory (down-regulating). Let $pa(v) = \{u : (u, v) \in E\}$ denote the set of v's parents, i.e. nodes that have an edge towards v. Accordingly, let A_v and I_v denote the disjoint subsets of $pa(v)$ consisting of the parents that activate and inhibit v, respectively.

We denote by $R \subset V$ the set of roots of the network, i.e., the nodes with no parents, which can be either cell receptors or certain cellular conditions which initiate downstream signaling. Also, let $G \subset V$ stand for the terminal nodes of $\mathbb{G}$; clearly $G \cap R = \emptyset$ (since there are no isolated nodes). While v will usually denote a generic node, we will use whenever possible r to denote a receptor node and g to represent a gene.

Each node $v \in V$ carries a random variable X_v, which quantifies the signaling activity of node v in the network. We will use small case letters (e.g. x_v) for realizations of random variables, and we will write X_B to indicate the set of random variables $\{X_v, v \in B\}$. For example, X_G is the set of variables associated with genes. These random variables are interpreted as follows. For each gene $g \in G$, X_g stands for the expression level of gene g in the cell, i.e., the amount of transcribed mRNA. All other variables $X_v, v \in V \setminus G$ are binary, and represent the state of signaling at node v, where $X_v = 0$ means "off" and $X_v = 1$ means "on," interpreted as the presence of signal at site v, ready to propagate down. *The stochastic process $X_V = \{X_v : v \in V\}$ is our representation of signaling activity in a single cell, and we encode the joint distribution in a Bayesian network.* Therefore, the probability that the whole system is in state $x_V = \{x_v, v \in V\}$ is

$$P(X_V = x_V) = \prod_{r \in R} p_r(x_r) \prod_{v \in V \setminus R} p_v(x_v | x_{pa(v)}).$$

Turning to the parametrization of the model, consider first the root nodes $r \in R$; since X_r is binary, there is one parameter per node, denoted $\phi_r = p_r(1) = P(X_r =$

1). For modeling signal transitions, to each $v \in V$, we attribute a function $\phi_v : \{0,1\}^{|pa(v)|} \to [0,1]$ which quantifies the net effect of the collection of signals $x_{pa(v)}$ from the parents of v. The extreme values, 0 and 1, correspond to pure inhibition and pure activation, respectively. More precisely,

- If v is neither a root nor a terminal node,

$$\phi_v(x_{pa(v)}) = E[X_v | X_{pa(v)} = x_{pa(v)}] = P(X_v = 1 | X_{pa(v)} = x_{pa(v)})$$

  which completely specifies the transition probability at v. They are "hard wired" in our model.

- If $g \in G$, the only property of the distribution of mRNA abundance X_g that will be needed is the conditional expectation given the parent TFs. We then introduce a scaling coefficient $a_g > 0$ and take

$$E[X_g | X_{pa(g)} = x_{pa(g)}] = a_g \phi_g(x_{pa(g)}). \tag{2.1}$$

  We can interpret this as follows: transcription is either "on" or "off" with probability $\phi_g(x_{pa(g)})$. When it is "on", the mean is a_g and when it is "off" the abundance is zero.

A possible choice, if v is not a root, is to take

$$\phi_v(x_{pa(v)}) = \frac{\sum_{u \in pa(v)} x_u \mathbf{1}\{u \in A_v\} + (1 - x_u)\mathbf{1}\{u \in I_v\}}{|pa(v)|}. \tag{2.2}$$

It is easy to see that ϕ_v is linearly increasing in the difference between the number of active up-regulating and down-regulating parents of v, which is clearly an over-simplified model of "transcriptional synergy", at least in the case of "competing" parents. More complex forms could be considered which are more faithful to the chemical interactions, perhaps even accounting for TF binding energies. However, in our particular network given in Figure 2.1, only a relatively small portion of nodes have competing parents and our choice like (2.2) has the major advantage of being parameter-free, allowing us to pre-compute certain quantities which appear repeatedly during parameter identification. Moreover, as we will discuss later, simulating the Bayesian network is significantly more efficient under the assumption of linearity in Equation (2.2).

2.2.2 Tissue Model

At the patient level, the measured abundance of mRNA for each gene on the microarray originates from a very large ensemble of cells contained in the sample tissue. Let $\mathcal{C}$ denote this ensemble of cells, with size $C = |\mathcal{C}|$, and let $x_{g,c}$ be amount of transcribed mRNA for gene $g \in G$ in cell $c \in \mathcal{C}$. The total abundance is denoted by $x_{g,\mathcal{C}} = \sum_c x_{g,c}$. By the law of large numbers, assuming that the Bayesian networks for the cells are independent and identically distributed within the subject, we have

$$\frac{x_{g,\mathcal{C}}}{C} \approx E[X_g | a_g, \phi_R]$$

where a_g and $\phi_R = \{\phi_r : r \in R\}$ are the model parameters that affect X_g. In addition, due to the Markov property of the network,

$$E[X_g|a_g, \phi_R] = E\big[E[X_g|a_g, X_{pa(g)}]\big|\phi_R\big] = E[a_g\phi_g(X_{pa(g)})|\phi_R].$$

Writing

$$\xi_g(\phi_R) = E[\phi_g(X_{pa(g)})|\phi_R], \tag{2.3}$$

for the expected transcription rate of gene g given the root activation probabilities ϕ_R, and dropping the approximation above, the transcript abundance in the tissue is

$$x_{g,\mathcal{C}} = a_g C\xi_g(\phi_R). \tag{2.4}$$

2.2.3 Population Level

It is not realistic to assume that the activation rates of the receptors and cellular conditions at the roots of the network are the same for every subject. Consequently, the final component of the model is to consider these rates to be subject-dependent, in fact random variables at the population level. That is, there is a random variable $\Phi_R^{(n)} = \{\phi_r^{(n)}, r \in R\}$ for each patient $n = 1, ..., N$. These variables are assumed independent and identically distributed across patients, for a given tissue type. Each component $\phi_r, r \in R$, independently follows a Beta prior

$$\phi_r \sim \beta(a_r, b_r).$$

with parameters a_r and b_r.

2.2.4 Measurement Model

It is well known that the actual measurement process, i.e., the steps leading up to what is actually recorded for each gene and patient, is complex. Thus, we should take into account the various stages of a microarray experiment including hybridization, scanning, background correction and normalization. As reported by numerous authors [84], [85], [86], we assume a linear relationship between scanned intensities of expression and actual RNA abundances. In particular, after undergoing all these steps, we consider the final log-expression reading $y_g^{(n)}$, obtained for gene $g \in G$ and subject $n = 1, ..., N$, to be the logarithm of a linearly increasing function of the corresponding tissue mRNA abundance $x_{g,\mathcal{C}}^{(n)}$, say

$$y_g^{(n)} = \log(b_g^{(n)} + c_g^{(n)} x_{g,\mathcal{C}}^{(n)}). \tag{2.5}$$

The gain parameter $c_g^{(n)}$ represents the net factor, that comes between patient n's actual molecule count for gene g and its processed probe intensity, before being transformed to log-scale. It involves the multiplicative measurement noise and accounts for experimental effects like hybridization efficiency, scanner gain and normalization. On the other hand, the additive term $b_g^{(n)}$ stands for the part of the intensity, that does not stem from the experimented mRNA, but rather effects like unspecific hybridization, detector offset etc.

As argued in [84, 87], we assume that the additive part $b_g^{(n)}$ of Equation (2.5) is negligibly small due to background correction applied by the scanner's imaging

software. Secondly, again proposed by the same authors, we consider a multiplicative decomposition for the gain factor $c_g^{(n)}$

$$c_g^{(n)} = \frac{d_g}{D^{(n)}} \exp\left\{\eta^{(g,n)}\right\}. \tag{2.6}$$

We interpret each component as follows: Combined with the scanner gain, d_g represents the background corrected hybridization efficiency of the probe set assigned to gene g. The quantity $D^{(n)}$, on the other hand, stands for the normalization constant applied across all probes of patient n. We take d_g to be gene specific and fixed for all subjects, whereas $D^{(n)}$ is subject dependent and the same for all genes. We further assume that $D^{(n)}$ is proportional to the number $C^{(n)}$ of cells contained in subject n's experimented tissue, since it is usually set to the total intensity captured from the corresponding array. Finally, the random component in $c_g^{(n)}$ is attributed to a multiplicative error term given as an exponential, whose argument is modeled i.i.d. for all genes and subjects, and realized as $\eta^{(g,n)}$ for gene g and patient n.

Combining Equations (2.4), (2.5) and (2.6), we obtain our measurement model for log expression of gene g and patient n

$$y_g^{(n)} = \lambda_g + \log \xi_g(\phi_R^{(n)}) + \eta^{(g,n)}$$

where, probe specific parameters and the ratio $C^{(n)}/D^{(n)}$, which is constant by assumption, are absorbed by the final readout offset $\lambda_g = \log \frac{a_g d_g C^{(n)}}{D^{(n)}}$, that is specific to gene g and independent of patients. In log scale, measurement noise η becomes additive and described as a zero mean Gaussian random variable with variance σ^2,

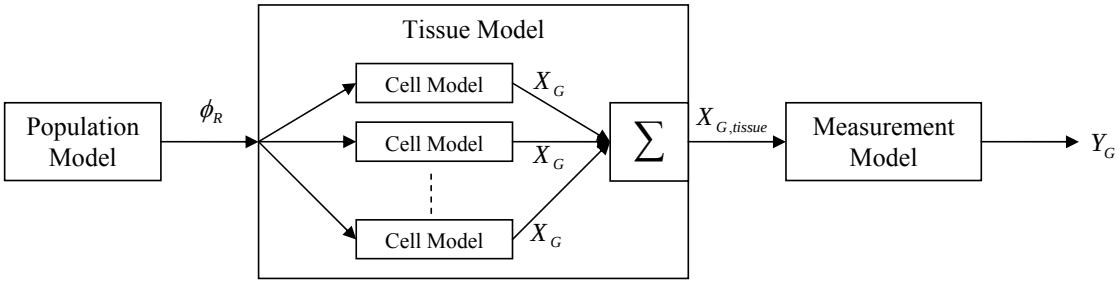

Figure 2.3: Overall model with individual levels put into generative order.

which is modeled the same for all genes and subjects.

In summary, our overall model, as illustrated in Figure 2.2 and shown as a block diagram in Figure 2.3, incorporates the entire process in a generative order from patient-to-patient differences in receptor activation, to the Bayesian network modeling of individual cell signaling, and to log-expression readouts at the population level. As a result, the final observation made for gene g for a given patient is modeled as a Gaussian random variable Y_g with conditional mean $\lambda_g + \log \xi_g(\phi_R)$, parametrized by the gene-dependent offset λ_g and subject-dependent root activation rates $\phi_R = \{\phi_r : r \in R\}$, and each ϕ_r has a Beta distribution with node-specific parameters a_r and b_r. The level of transcriptional regulation $\xi_g(\phi_R)$ for target g is evaluated using the single-cell Bayesian network model. Finally, the variance σ^2 accounts for the variation in measurement error, which is the same for all genes.

2.2.5 Expected Transcription Rate Function

Recall from Equation (2.3) that for each gene $g \in G$, $\xi_g(\phi_R)$ represents the cell-level average transcription rate of g, where we interpret this equation as a conditional expectation given the root activation rates are fixed to be ϕ_R. Let R_g denote the set of roots which are ancestors of g, so that $\xi_g(\phi_R)$ only depends on ϕ_{R_g}. Since these root variables are binary and independent,

$$P(X_{R_g} = x_{R_g}) = \prod_{r \in R_g} \phi_r^{x_r}(1-\phi_r)^{1-x_r}.$$

Consequently,

$$\xi_g(\phi_R) = \sum_{x_{R_g} \in \{0,1\}^{|R_g|}} E\big[\phi_g(X_{pa(g)}) | X_{R_g} = x_{R_g}\big] \prod_{r \in R_g} \phi_r^{x_r}(1-\phi_r)^{1-x_r}. \tag{2.7}$$

It will be important in the following to have a quick access to the value of $\xi_g(\phi_R)$ for any given choice of the root activation rates. One possibility is to pre-compute all the coefficients of the above polynomial expression (i.e., all the $E[\phi_g(X_{pa(g)})|X_{R_g} = x_{R_g}]$), which are parameter-free, and evaluate the polynomial when needed. This is tractable as long as $2^{|R_g|}$ remains manageable, which is the case with our network where $|R_g|$ does not exceed 5. The pre-computation of the conditional expectations has to be done only once. It can be done exactly for small networks (including, again, our case), or for specific topologies. In the general case, approximate (and often good) values can be computed using belief propagation methods, or Monte-Carlo sampling. When $|R_g|$ is too large for this strategy to be tractable, it is still possible to compute

or approximate $\xi_g(\phi_R)$ for a given ϕ_R using belief propagation each time its value is needed (without pre-computation).

Finally, we notice that the computation of ξ_g can be done very efficiently when, for each $v \in V$, the function ϕ_v depends linearly on the states $x_{pa(v)}$ of the parents. This property is true in particular in the model proposed in (2.2). In that case, ξ_g can be evaluated using dynamic programming along the network's top-down hierarchy, thanks to the following proposition.

Proposition 2.2.1. *Suppose that for all $v \in V$,*

$$E[X_v|X_{pa(v)}] = c_v + \sum_{u \in pa(v)} c_{uv} X_u$$

for some coefficients c_v and c_{uv}. Then for all $v \in V$,

$$E[X_v|\phi_R] = d_v + \sum_{r \in R} d_{rv} \phi_r$$

for the coefficients d_v and d_{rv} determined by the recursions $d_v = c_v + \sum_{u \in pa(v)} c_{uv} d_u$ and $d_{rv} = \sum_{u \in pa(v)} c_{uv} d_{ru}$.

Proof. The result follows from the Markov property of the process and the linearity of the expectation. If $v \in R$, the claim holds by definition, with $d_v = 0$ and $d_{rv} = \delta(r, v)$.

Otherwise, suppose the claim holds for all parents of v; then by induction we have

$$\begin{aligned}
&E[X_v|\phi_R] = E\big[E[X_v|X_{pa(v)}]\big|\phi_R\big] \\
&= E\left[c_v + \sum_{u\in pa(v)} c_{uv}X_u\middle|\phi_R\right] = c_v + \sum_{u\in pa(v)} c_{uv}E[X_u|\phi_R] \\
&= c_v + \sum_{u\in pa(v)} c_{uv}\left(d_u + \sum_{r\in R} d_{ru}\phi_r\right) \\
&= \underbrace{\left(c_v + \sum_{u\in pa(v)} c_{uv}d_u\right)}_{d_v} + \sum_{r\in R}\underbrace{\left(\sum_{u\in pa(v)} c_{uv}d_{ru}\right)}_{d_{rv}}\phi_r
\end{aligned}$$

□

Thus, in the case of Proposition 2.2.1, it suffices to pre-compute coefficients d_{rg} for $r \in R$ and $g \in G$ to ensure a computation of $\xi_g(\phi_R)$ in a time which is linear in the number of roots.

2.3 Learning Algorithm

Our model has both observed and hidden variables. The observed ones are the gene expression levels $\mathbf{y}_G = \{y_g^{(n)} : g \in G, n = 1, ..., N\}$ over N subjects. All other variables are unobserved. Among these, we are particularly interested in the root activation rates $\Phi_R = \{\phi_r^{(n)} : r \in R, n = 1, ..., N\}$, which constitute the hidden phenotypic information about the individuals in the population. The joint density of gene expression values and activation rates is given by

$$f_{GR}(y_G, \phi_R|\theta) = f_{G|R}(y_G|\phi_R; \theta) f_R(\phi_R|\theta) = \prod_{g \in G} f_{g|R}(y_g|\phi_R; \theta) \prod_{r \in R} f_r(\phi_r|\theta) \qquad (2.8)$$

where θ refers to the parameters of the model. For each gene $g \in G$, the conditional density of corresponding log-expression Y_g is Gaussian

$$f_{g|R}(y_g|\phi_R; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ - \frac{\big(y_g - \lambda_g - \log \xi_g(\phi_R)\big)^2}{2\sigma^2} \right\},$$

with an offset λ_g and noise variance σ^2. For each root node $r \in R$, the corresponding activation rate ϕ_r has a standard beta prior

$$f_r(\phi_r|\theta) = \frac{\phi_r^{a_r - 1}(1 - \phi_r)^{b_r - 1}}{B(a_r, b_r)},$$

with shape parameters (a_r, b_r), where $B(a, b) = \int_0^1 x^{(a-1)}(1 - x)^{(b-1)} dx$ is the beta function.

In summary, the parameters to be inferred are

$$\theta = \{\lambda_g, a_r, b_r, \sigma^2 : g \in G, r \in R\},$$

which includes the shape parameters (a_r, b_r) of the beta priors on receptor activation rates, the variance σ^2 of additive measurement noise, and the offset parameters λ_g for gene expressions. The beta parameters are specific to each individual receptor, but constant across the patient population. The noise variance is constant for all genes and patients. Gene offsets are specific to each individual gene but constant across subjects.

Model identification, i.e. learning θ is based on observed expression data $\mathbf{y}_G$, where we assume each Y_g to be conditionally independent of expressions $Y_{G\setminus\{g\}}$ of other genes, given activation rates ϕ_R. Hidden components that we represent in the cell level, namely the signaling proteins and transcription factors, as well as their wiring, appear implicitly in functions ξ_g $(g \in G)$.

The standard method for learning such a latent variable model is the expectation maximization (EM) algorithm [38]. Briefly, EM provides an improving sequence $\big(\widehat{\theta}^{(t)}\big)_{t\geq 1}$ of parameter estimates by iteratively maximizing the conditional expectation of the complete data log-likelihood, given i.i.d. incomplete observations. In particular, each iteration t of EM involves (i) an E-step which requires computing the missing data posterior, $f_{R|G}\big(\Phi_R|\mathbf{y}_G;\widehat{\theta}^{(t)}\big)$, in order to evaluate the current objective function

$$Q\big(\theta|\widehat{\theta}^{(t)}\big) = E\big[\log f_{GR}\big(\mathbf{Y}_G, \Phi_R|\theta\big)\big|\mathbf{y}_G;\widehat{\theta}^{(t)}\big] \tag{2.9}$$

and (ii) an M-step, in which one solves for the new parameter estimates

$$\widehat{\theta}^{(t+1)} = \arg\max_{\theta} Q\big(\theta|\widehat{\theta}^{(t)}\big) \tag{2.10}$$

by maximizing the objective function. This procedure is repeated until convergence is evident.

Evaluating (2.9) is usually simplified when the likelihood of the complete data model (including both hidden and observed variables) belongs to an exponential family, which is indeed the case here, since we can write

$$\log f_{GR}\big(\mathbf{y}_G, \Phi_R|\theta\big) = -\Lambda(\theta) + \langle \Pi(\theta), \mathbf{S}\big(\mathbf{y}_G, \Phi_R\big)\rangle \tag{2.11}$$

where $\langle \cdot, \cdot \rangle$ denotes the vector scalar product, Λ and Π are scalar and vector-valued functions of θ given by

$$\Lambda(\theta) = N\bigg(\sum_{r\in R} \log B(a_r, b_r) + |G| \log\sqrt{2\pi}\sigma + \frac{1}{2\sigma^2}\sum_{g\in G}\lambda_g^2\bigg),$$

$$\Pi(\theta) = N \begin{pmatrix} \begin{pmatrix} a_r - 1 \\ b_r - 1 \end{pmatrix}_{r\in R} \\ \\ \frac{1}{2\sigma^2}\begin{pmatrix} 2\lambda_g \\ -1 \\ -2\lambda_g \\ -1 \\ 2 \end{pmatrix}_{g\in G} \end{pmatrix}$$

and $\mathbf{S}\big(\mathbf{y}_G, \Phi_R\big)$ is a vector-valued complete data sufficient statistic

$$\mathbf{S}\big(\mathbf{y}_G, \Phi_R\big) = \frac{1}{N}\sum_{n=1}^{N} \begin{pmatrix} \begin{pmatrix} \log \phi_r^{(n)} \\ \log\big(1-\phi_r^{(n)}\big) \end{pmatrix}_{r\in R} \\ \begin{pmatrix} y_g^{(n)} \\ \big[y_g^{(n)}\big]^2 \\ \log \xi_g\big(\phi_R^{(n)}\big) \\ \big[\log \xi_g\big(\phi_R^{(n)}\big)\big]^2 \\ y_g^{(n)} \log \xi_g\big(\phi_R^{(n)}\big) \end{pmatrix}_{g\in G} \end{pmatrix}. \tag{2.12}$$

Parameters that maximize the complete data log-likelihood in (2.11), can be expressed as a function of this sufficient statistic, i.e., in the form

$$\mathbf{S} \mapsto \widehat{\theta}_{\mathrm{ML}}(\mathbf{S}).$$

To be explicit, given $\mathbf{S}$ as in (2.12), let $\mathbf{s}_r = \big[s_r^{(i)}\big]_{i=1}^{2}$ and $\mathbf{s}_g = \big[s_g^{(i)}\big]_{i=1}^{5}$ denote its two- and five-dimensional sub-vectors, corresponding to the receptor $r \in R$ and the gene $g \in G$, respectively. Then, the maximum likelihood estimator $\widehat{\theta}_{\mathrm{ML}}(\mathbf{S})$ is characterized as follows:

- For each receptor $r \in R$, $\widehat{a}_r$ and $\widehat{b}_r$ of the beta prior f_r will satisfy

$$\psi\big(\widehat{a}_r\big) - \psi\big(\widehat{a}_r + \widehat{b}_r\big) = s_r^{(1)} \tag{2.13}$$

$$\psi\big(\widehat{b}_r\big) - \psi\big(\widehat{a}_r + \widehat{b}_r\big) = s_r^{(2)} \tag{2.14}$$

where $\psi(x) = \Gamma'(x)/\Gamma(x)$ is the digamma function. A closed-form solution to that system does not exist, but, similar to standard maximum likelihood parameter estimation of a beta density, $\widehat{a}_r$ and $\widehat{b}_r$ are obtained numerically.

- For each gene $g \in G$, the estimate for the offset parameter is given by

$$\widehat{\lambda}_g = s_g^{(1)} - s_g^{(3)} \tag{2.15}$$

- Finally, the estimate for noise variance is obtained by

$$\widehat{\sigma}^2 = \frac{1}{|G|} \sum_{g \in G} s_g^{(2)} - 2s_g^{(5)} + s_g^{(4)} - \widehat{\lambda}_g^2. \tag{2.16}$$

Since (2.9) can be rewritten as

$$Q(\theta|\widehat{\theta}^{(t)}) = -\Lambda(\theta) + \langle \Pi(\theta), E\big[\mathbf{S}|\mathbf{y}_G; \widehat{\theta}^{(t)}\big] \rangle$$

the E-step can be reduced to computing the conditional expectation of the sufficient statistic, namely

$$\mathbf{S}^{(t+1)} = E\big[\mathbf{S}|\mathbf{y}_G; \widehat{\theta}^{(t)}\big]. \tag{2.17}$$

Then, (2.10) of the M-step can be expressed as

$$\widehat{\theta}^{(t+1)} = \widehat{\theta}_{\mathrm{ML}}(\mathbf{S}^{(t+1)}) \tag{2.18}$$

However, due to the marginal beta distribution of Φ_R, there is no simple closed form for the the computation of (2.17) in the E-step and straightforward EM is intractable here. Instead, we will consider a stochastic variant, the Stochastic Approximation EM (SAEM) algorithm, wherein the E-step is approximated with Monte

Carlo sampling. Under mild conditions [39, 88], SAEM converges to (local) maxima of the objective function if the complete data log-likelihood belongs to a curved exponential family, which is the case in our model. Basically, SAEM replaces the E-step of conventional EM with a stochastic approximation running in parallel, involving the *simulation* of missing data Φ_R. In its simple form, the SAEM algorithm makes an iterative approximation of $\mathbf{S}^{(t+1)}$ by defining

$$\widehat{\mathbf{S}}^{(t+1)} = \widehat{\mathbf{S}}^{(t)} + \gamma^{(t)}\big(\mathbf{S}(\Phi_R^{(t)}, \mathbf{y}_G) - \widehat{\mathbf{S}}^{(t)}\big) \tag{2.19}$$

where $\big(\gamma^{(t)}\big)_{t\geq 1} \in [0,1]$ is a decreasing sequence of positive step sizes starting with $\gamma^{(1)} = 1$, and $\Phi_R^{(t)}$ is a simulated sample of Φ_R, drawn conditionally to $\mathbf{y}_G$ for the current parameter $\theta^{(t)}$. The M-step is then given by

$$\widehat{\theta}^{(t+1)} = \widehat{\theta}_{\mathrm{ML}}(\widehat{\mathbf{S}}^{(t+1)}). \tag{2.20}$$

In principle, in order to ensure the convergence of the SAEM algorithm, one should take $\sum_{t=1}^{\infty} \gamma^{(t)} = \infty$ and $\sum_{t=1}^{\infty} (\gamma^{(t)})^2 < \infty$.

So the SAEM algorithm replaces computing conditional expectations by sampling from the conditional distribution which is most of the time much more feasible. Moreover, variants of this algorithm allow for coupling the iterations with Markov chain Monte-Carlo sampling when direct sampling is not feasible or not efficient (which is the case for our model). One can also use more than one sample $\Phi_R^{(t)}$ at each step, using a sample average in (2.19). The explicit implementation of the variant we have used is described in the next sections, for a single iteration t.

2.3.1 Simulation

Given the current parameter values $\widehat{\theta}^{(t)}$ and observed expression data $\mathbf{y}_G$, we generate $M^{(t)} \geq 1$ realizations $\Phi_R^{(t,m)} = \left\{\phi_r^{(n,t,m)} : r \in R, n = 1, ..., N\right\}$, $(m = 1, ..., M^{(t)})$ of missing data under their joint posterior $f_{R|G}\big(\cdot\,|\mathbf{y}_G; \widehat{\theta}^{(t)}\big)$. For this, we use the Gibbs sampling algorithm, which sequentially produces an instance for each ϕ_r, from its univariate conditional given the observations and already sampled current states of other root variables $\phi_{R\setminus\{r\}}$. The resulting sequence $\left(\phi_R^{(n,t,m)}\right)_{m\geq 1}$ of realizations will then constitute a Markov chain, whose stationary distribution is the sought-after posterior $f_{R|G}$.

For each $r \in R$, let G_r be the set of genes which are descendants of r and let R_r be the set of root nodes other than r which have a descendant in G_r. Then, using Bayes rule and the Markov property, we can write the conditional density of the activation rate ϕ_r of root $r \in R$, given the realizations $(y_G, \phi_{R\setminus\{r\}})$ of remaining variables, as

$$\begin{aligned}\frac{f_{GR}(y_G, \phi_r, \phi_{R\setminus\{r\}}|\theta)}{\int_0^1 f_{GR}(y_G, \tilde{\phi}_r, \phi_{R\setminus\{r\}}|\theta)d\tilde{\phi}_r} &= \frac{f_R(\phi_r, \phi_{R\setminus\{r\}}|\theta) f_{G|R}(y_G|\phi_r, \phi_{R\setminus\{r\}};\theta)}{\int_0^1 f_R(\tilde{\phi}_r, \phi_{R\setminus\{r\}}|\theta) f_{G|R}(y_G|\tilde{\phi}_r, \phi_{R\setminus\{r\}};\theta)d\tilde{\phi}_r} \\ &= \frac{f_r(\phi_r|\theta)\prod_{g\in G_r} f_{g|R}(y_g|\phi_r, \phi_{R_r};\theta)}{\int_0^1 f_r(\tilde{\phi}_r|\theta)\prod_{g\in G_r} f_{g|R}(y_g|\tilde{\phi}_r, \phi_{R_r};\theta)d\tilde{\phi}_r}.\end{aligned}$$

Terms $\prod_{r'\in R\setminus\{r\}} f_{r'}(\phi_{r'}|\theta)\prod_{g\in G\setminus G_r} f_{g|R}(y_g|\phi_{R\setminus\{r\}};\theta)$ that do not involve anything indexed with r cancel each other in the first line, yielding the final expression, which only depends on realizations at nodes $G_r \cup R_r$, i.e. the "Markov blanket" of r in $G \cup R$ (see Figure 2.4). Thus, we denote ϕ_r's univariate conditional given the rest of the variables by $f_{r|G_rR_r}(\cdot|y_{G_r}, \phi_{R_r};\theta)$.

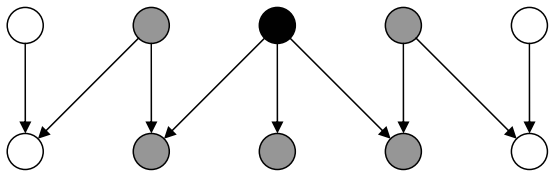

Figure 2.4: A simple DAG with 5 roots and 5 leaves. The Markov Blanket of the black node is the set of gray nodes.

Then, the m^{th} realization $\phi_R^{(n,t,m)}$ of missing root variables for subject n and iteration t of SAEM is produced by Gibbs sampling as follows:

(i) Set

$$\phi_R^{(n,t,m)} := \begin{cases} \phi'_R \sim U\big([0,1]^{|R|}\big), & \text{if } t = 1 \text{ and } m = 1; \\ \phi_R^{(n,t-1,m)}, & \text{if } t > 1 \text{ and } m = 1; \\ \phi_R^{(n,t,m-1)}, & \text{otherwise.} \end{cases}$$

(ii) Visit the root nodes in some fixed order and, for each $r \in R$, set

$$\phi_r^{(n,t,m)} := \phi'_r \sim f_{r|G_r R_r}(\cdot|y_{G_r}^{(n)}, \phi_{R_r}^{(n,t,m)}; \widehat{\theta}^{(t)}).$$

Simulating instances from $f_{r|G_r R_r}$ at step (ii) is still not straightforward but can easily done with factored sampling. Since

$$f_{r|G_r R_r}(\cdot|y_{G_r}, \phi_{R_r}; \theta) \propto f_r(\cdot|\theta) \prod_{g \in G_r} f_{g|R}(y_g|\cdot, \phi_{R_r}; \theta),$$

generating K samples $\{z^{(1)}, ..., z^{(K)}\}$ from the prior beta density $f_r(\cdot|\theta)$ and selecting $z^{(i)}$ $(i = 1, ..., K)$, with probability

$$\pi^{(i)} = \frac{\prod_{g \in G_r} f_{g|R}(y_g|z^{(i)}, \phi_{R_r}; \theta)}{\sum_{j=1}^{K} \prod_{g \in G_r} f_{g|R}(y_g|z^{(j)}, \phi_{R_r}; \theta)},$$

as the new realization for ϕ_r, will approximate a variable from $f_{r|G_r R_r}$, as K tends to be large.

Drawing samples from standard beta priors is straightforward with available statistical packages, and so is evaluating weights $\pi^{(i)}$. Also, with a reasonable K, one does not have to wait for Gibbs sampler to mix within every single execution of the *simulation* step, since last samples returned from a given iteration of SAEM, are already used to initialize the chain for the next iteration.

2.3.2 Stochastic Approximation

We update the sufficient statistic according to

$$\widehat{\mathbf{S}}^{(t+1)} = \widehat{\mathbf{S}}^{(t)} + \gamma^{(t)} \left(\frac{\sum_{m=1}^{M^{(t)}} \mathbf{S}(\Phi_R^{(t,m)}, \mathbf{y}_G)}{M^{(t)}} - \widehat{\mathbf{S}}^{(t)} \right), \tag{2.21}$$

which generalizes the simple form in (2.19) via taking an average over $M^{(t)} \geq 1$ simulated versions of complete data.

2.3.3 Maximization

We compute $\widehat{\theta}^{(t+1)} = \widehat{\theta}_{\mathrm{ML}}(\widehat{\mathbf{S}}^{(t+1)})$ using the stochastic approximation in (2.21) as input to the maximum likelihood estimator described by Equations (2.13)-(2.16). In summary, the model parameters are efficiently learned by keeping track of complete data sufficient statistics, which are improved with new realizations of missing data.

2.3.4 Root Activation Probabilities

The sequences $\left(\Phi_R^{(t,m)}\right)_{t\geq 1, m\geq 1}$ that are generated by the SAEM algorithm can also be used to estimate subject-dependent expected root activation probabilities given the corresponding gene expression levels. That is, the conditional expectation $E\left[\phi_R | y_G^{(n)}; \widehat{\theta}^{(t)}\right]$ can be recursively approximated by

$$\widehat{\phi}_R^{(n,t)} = \widehat{\phi}_R^{(n,t-1)} + \gamma^{(t)} \left(\frac{\sum_{m=1}^{M^{(t)}} \phi_R^{(n,t,m)}}{M^{(t)}} - \widehat{\phi}_R^{(n,t-1)} \right) \tag{2.22}$$

which, at SAEM's convergence is returned as patient n's phenotype estimate $\widehat{\phi}_R^{(n)}$.

2.4 Experiments on RAS-RAF Network

In this section we present experiments in learning the network, measuring the stability of model identification, and estimating the states of the hidden variables, especially the activation states of the receptors. Our data consist of gene expression levels measured for 38 genes and collected from 118 breast cancer patients. The observed genes, i.e. the targets of the signaling network of Figure 2.1, are again listed in Table 2.1, together with their known transcription factors and associated type of regulation. The data set also contains complete measurements for the ERα status of patients.

2.4.1 Validating Identifiability of Model

Before discussing experiments with real patient data, we first verify that the model can be accurately identified from artificial gene expressions simulated with known parameters. Given the parameter vector θ, we generate subject-dependent receptor activation rates $\Phi_R = \{\phi_r^{(n)} : r \in R, n = 1, ..., N\}$ according to their beta priors $f_R(\cdot|\theta)$; and, conditioned on these rates, we sample gene expressions $\mathbf{y}_G = \{y_g^{(n)} : g \in G, n = 1, ..., N\}$ from $f_{G|R}(\cdot|\Phi_R; \theta)$. We then evaluate the fit between the true parameter vector θ and the estimate $\widehat{\theta}$ that is learned by applying the algorithm to the simulated observations $\mathbf{y}_G$. In particular, since the SAEM algorithm also returns predictions $\widehat{\Phi}_R$ of receptor activation rates, we can compare those subject-specific

Table 2.1: List of observed genes and their parent transcription factors (also shown in Figure 2.1 bottom); the type of regulation (activating or inhibiting) is indicated by the arrows

Gene	Parent TF(s)	Gene	Parent TF(s)
CSF2	NFkB↑, JUN↑	NQO1	ERβ↑, JUN↑
TP63	p53↑	TSHB	JUN↑
EGR2	Elk1↑	ATF3	JUN↑
CYP1B1	ERα↓	SCN3B	p53↑
LHB	ERα↑	SOCS1	STAT5↑
BHLHB2	Hif1↑	IL8	NFkB↑
CKB	ERα↑	IL12B	JUN↓, NFkB↑
PRL	ERα↑	IL3	JUN↑
BRCA1	JUN↓	VEGFA	Hif1↑, ERβ↑
CSN2	STAT5↑	IL4	JUN↑
BCL2A1	JUN↑	NPPA	JUN↓
EPO	Hif1↑	CXCL9	NFkB↑
CASP1	p53↑	CCNG1	p53↑
JUNB	SMAD4↑	GADD45A	p53↑
FOS	ERβ↓, p53↓, TCF↑	FASN	STAT5↓
LDHA	Hif1↑	IL2	JUN↑, NFkB↑
PDX1	FOXO1↓	IFNB1	JUN↑, NFkB↑
CKM	p53↑	CDK4	Myc↑
PTTG1	p53↓	PMAIP1	p53↑

estimates with their simulated true counterparts Φ_R that are kept hidden during learning.

We can also conduct the above procedure at different noise levels. Note that, with simulated phenotypes Φ_R, our model assumes that the log of expected transcription rates $\log \xi_g(\Phi_R) = \{\log \xi_g(\phi_R^{(n)}) : n = 1, ..., N\}$ is the noise-free signal that determines the subject-dependent variation for each gene $g \in G$. Letting $\overline{\log \xi_g} = \sum_n \log \xi_g(\phi_R^{(n)})/N$ denote the corresponding sample average, and given the variance σ^2 of the measurement noise, the signal-to-noise ratio (SNR), measured in dB, is found by

$$\text{SNR} = 10 \log_{10} \frac{\sum_g \sum_n (\log \xi_g(\phi_R^{(n)}) - \overline{\log \xi_g})^2}{|G| N \sigma^2}.$$

Table 2.2 provides a summary of how accurately the activation rates are estimated at different SNR levels. For each $r \in R$, the correlation coefficients between the simulated true vector $\left[\phi_r^{(n)}\right]_{n=1}^N$ and its learned estimate $\left[\widehat{\phi}_r^{(n)}\right]_{n=1}^N$ are given as an average score over 10 independent experiments per choice of SNR, where the experiments differ in the random selections of the true parameters used to simulate data of sample size $N = 100$. Clearly the model is accurately identified with moderately sized learning samples and even with SNR $= 0$, where the standard deviation in $\log \xi_g(\phi_R)$ averaged across all genes $g \in G$, i.e. the root mean squared amplitude of the subject-dependent signal is the same as that of noise. In particular, the estimates of the receptors ERα and EGFR inferred from simulated data are more precise since they affect the majority of the genes observed.

Table 2.2: Model Identification from simulated data. Pearson correlation coefficients between true (simulated) activation rates $\left[\phi_R^{(n)}\right]_{n=1}^N$ and their learned estimates $\left[\widehat{\phi}_R^{(n)}\right]_{n=1}^N$ at different SNR levels. Scores averaged over 10 experiments with randomly selected parameters for simulation. Sample size $N = 100$.

SNR (dB)	ERα	ERβ	EGFR	TGFβR	DNA d.	Hypoxia
-10	0.72	0.26	0.76	0.53	0.51	0.33
-5	0.87	0.45	0.89	0.77	0.67	0.54
0	0.94	0.69	0.95	0.89	0.86	0.74
5	0.97	0.88	0.98	0.93	0.92	0.88
10	0.98	0.94	0.99	0.97	0.97	0.94

2.4.2 Estimating Receptor Activity from Real Data

One important way to measure the utility of the model is to estimate the states of the receptor proteins from the gene expression data. In our model, these states are binary variables, each sampled independently from a patient-dependent activation rate. Consequently, it is these rates which are the more natural targets of estimation. For each of the 118 patients, we are provided with a binary label for the measured phenotypes, either "ERα-positive" or "ERα-negative". Our activation rate estimates $\{\widehat{\phi}_{\mathrm{ER}\alpha}^{(n)}\}_{n=1}^N$ are scalars. The rank-sum test, also known as Mann-Whitney-Wilcoxon test, offers a natural and robust way to compare predictions, especially by averaging over different parameter initializations. It is a nonparametric procedure for testing the hypothesis that two independent samples are identically distributed.

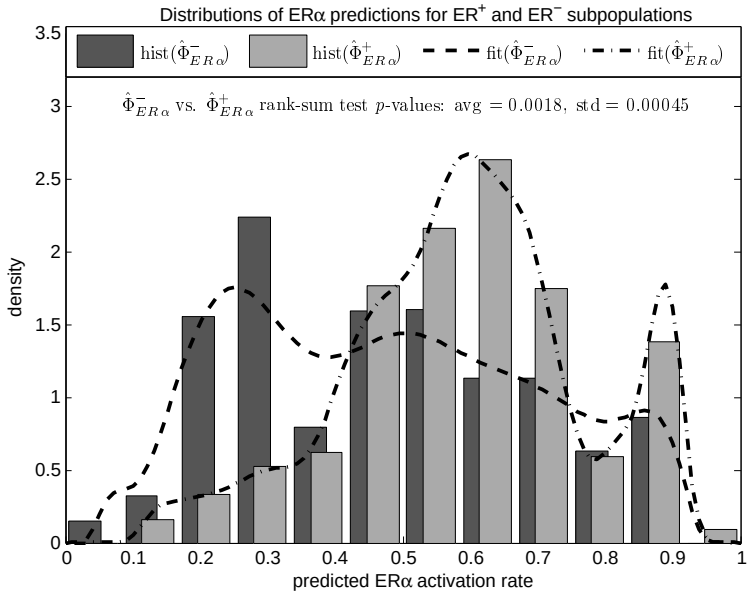

Figure 2.5: Normalized histograms and nonparametric density fits of patient dependent predictions for ERα activation rates corresponding to ER^+ and ER^- subpopulations. Histograms are generated and rank sum test p-values are averaged over 20 independent runs.

Let $ER^+, ER^- \subset \{1, 2, ..., N\}$ be the sub-populations of patients who are ERα-positive and ERα-negative, respectively. In our case, the null hypothesis H_0 is that the activation rates from these two sub-populations are identically distributed. Our data are the estimated rates $\widehat{\Phi}^+_{\mathrm{ER}\alpha} = \{\widehat{\phi}^{(n)}_{\mathrm{ER}\alpha} : n \in ER^+\}$ and $\widehat{\Phi}^-_{\mathrm{ER}\alpha} = \{\widehat{\phi}^{(n)}_{\mathrm{ER}\alpha} : n \in ER^-\}$, where $|ER^+| = N^+ = 75$ and $|ER^-| = N^- = 43$.

Figure 2.5 compares the histograms of estimates $\widehat{\Phi}^+_{\mathrm{ER}\alpha}$ and $\widehat{\Phi}^-_{\mathrm{ER}\alpha}$ (superposed with their non-parametric density fits for better visualization) obtained with 20 repeated experiments where each run of the algorithm differs in random parameter initializations. The rank-sum test p-value averaged over these 20 experiments is found 0.0018 with standard deviation 0.00045. As can be seen in the separation of histogram

modes, the estimates are reproducible and consistent with phenotypes.

The data set also reproduces the EGFR status for 79 of the 118 patients, again recorded as EGFR-positive or EGFR-negative, but with only 8 positives. The same rank-sum test approach to correlate this information and the EGFR rate predicted by the model, failed to provide a significant p-value, but this would have been very hard to achieve due to the limited power of the rank-sum test with such a small number of available EGFR-positive patients.

2.4.3 Estimating Other Protein Activities

Given predictions for receptor activation rates, we can also deduce the subject dependent states of the non-receptor components, i.e. signaling proteins and transcription factors that are not explicitly involved in (2.8). Having estimated $\widehat{\phi}_R^{(n)}$ for each patient n, the subject-specific expected status $\widehat{\bar{x}}_v^{(n)} = E\big[X_v|\widehat{\phi}_R^{(n)}\big]$ of each network component $v \in V \setminus G$ can be directly evaluated similarly to the way in which we computed the ξ_g's in Equation (2.7). Letting R_v denote the root ancestors of v, we get

$$\widehat{\bar{x}}_v^{(n)} = \sum_{x_{R_v} \in \{0,1\}^{|R_v|}} E[X_v|X_{R_v} = x_{R_v}] \prod_{r \in R_v} \big(\widehat{\phi}_r^{(n)}\big)^{x_r} \big(1 - \widehat{\phi}_r^{(n)}\big)^{1-x_r}, \tag{2.23}$$

where, again, the expectations involved in the sum are parameter-free and can be precomputed using Proposition 2.2.1. With that notation, subject n's expected status $\widehat{\bar{x}}_r^{(n)}$ at a root $r \in R$ is the same as the prediction of the corresponding activation rate

Table 2.3: Repeated random sub-sampling validation of the method: Estimated receptor activation rates are compared after training on disjoint sub-populations. Scatter plots are generated and correlations are averaged with 20 random selections of sub-sample $A \subset \{1, ..., N\}$ ($|A| = \frac{N}{2}$).

Root $r \in R$	ERα	ERβ	EGFR	TGFβR1-2	DNA damage	Hypoxia
$\widehat{\phi}_r^{(n\|A)}$ vs. $\widehat{\phi}_r^{(n\|A^c)}$ $(n = 1, ..., N)$						
$\text{corr}\left\{\left[\widehat{\phi}_r^{(n\|A)}\right]_{n=1}^N, \left[\widehat{\phi}_r^{(n\|A^c)}\right]_{n=1}^N\right\}$	0.98	0.99	0.98	0.98	0.96	0.91

$\widehat{\phi}_r^{(n)}$.

For a node v with only one parent, say u, the above computation simplifies to $\widehat{\bar{x}}_v = \widehat{\bar{x}}_u$ (resp. $1-\widehat{\bar{x}}_u$), since in evaluating the expectation $E[X_v|X_{R_v} = x_{R_v}]$, Equation (2.2) will give $\phi_v(x_u) = P(X_v = 1|X_u = x_u) = E[X_v|X_u = x_u] = x_u$ (resp. $1 - x_u$) if u activates (resp. inhibits) v. In other words, along linear sequences, signaling is assumed to propagate deterministically, where each node either copies or reverses the status of its single parent. Thus, our model is invariant to adding/removing components at such pathways. That is, topologies that reduce to the same collapsed structure yield the same data likelihood as well as the same predictions for common nodes.

Figure 2.6 shows a gray scale heat map (black: low, white: high) of estimates for hidden components appended to the observed gene expressions, where, to avoid redundancy, hidden nodes with only one parent are excluded. As mentioned above, these can be directly deduced from the ones already shown. Spot (v, n) gives the estimated or observed status of signaling component v, for patient n. Each row is scaled to a common dynamic range by subtracting the row mean and normalizing with row standard deviation. Columns (i.e. patients) are arranged according to the rank of the projection of the corresponding gene profile onto the direction of largest variation in gene space, namely the first eigenvector of the covariance matrix of observations. Besides demonstrating our ability to estimate the subject-dependent status of cell signaling, further analysis of this picture is limited by the absence of ground truth

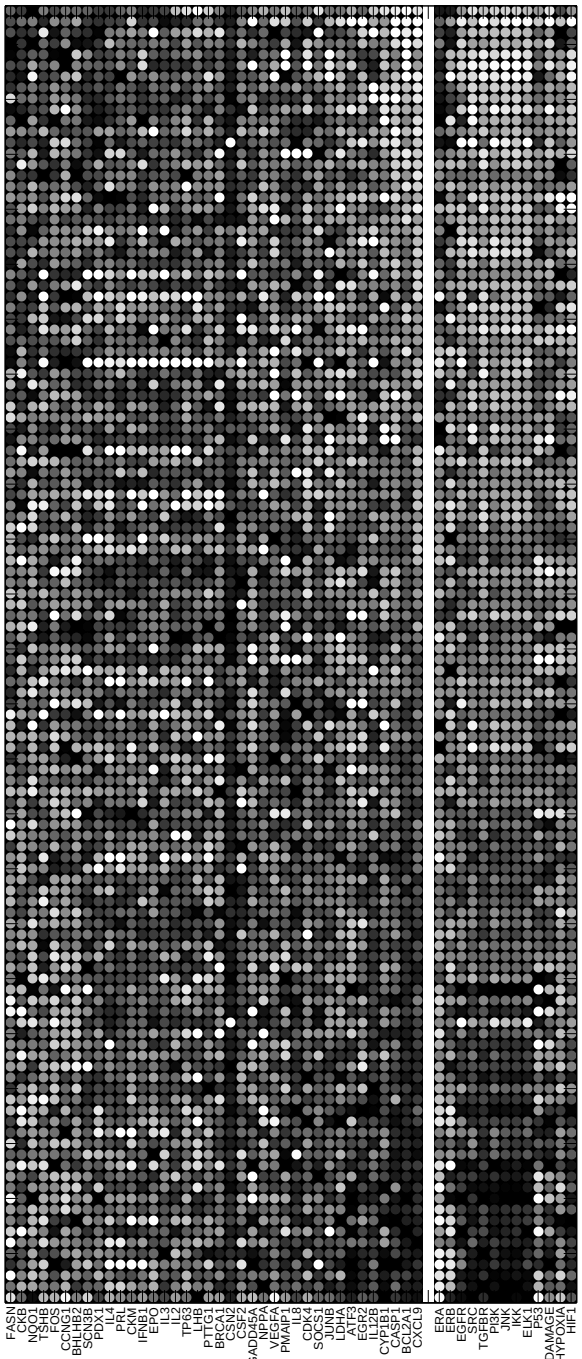

Figure 2.6: Grayscale heat map of patient-specific networks. Each row corresponds to a signaling component and each column to a patient. The rows are scaled to a common dynamic range. The columns are arranged according to the rank of the projection of the corresponding gene profile in the direction of largest variation in gene space. The white stripe separates the observed log gene expression levels on top from the estimates of the hidden components. Hidden nodes that have only one parent (the ones that are intermediate proteins along linear chains) are excluded to avoid repetition, since their predictions are either the same or the inverse of their parent, directly deducible from the ones already shown.

for hidden nodes. However it is noteworthy that our inference of hidden nodes aligns with the first order variation amongst genes.

2.4.4 Reproducibility and Sensitivity to Sample Size

In order to assess the method's generalization power and sensitivity to sample size we used a "repeated random sub-sampling validation" procedure, where we repeatedly partitioned the available gene expression data into two random halves, and checked the fit between models learned from these two disjoint subsets.

In order to describe this validation study, let $B \subset \{1, ..., N\}$ be a sub-population of patients and let $\widehat{\theta}^{(B)}$ denote the model parameters learned from the corresponding expression data $\mathbf{y}_G^{(B)} = \{y_g^{(n)} : g \in G, n \in B\}$. Then, based on the model with estimated parameters $\widehat{\theta}^{(B)}$, let $\widehat{\phi}_R^{(k|B)} = E\big[\phi_R | y_G^{(k)}; \widehat{\theta}^{(B)}\big]$ denote the predicted receptor activation rates for patient k, who may or may not be in B.

The expectation involved in $\widehat{\phi}_R^{(k|B)}$ can be evaluated by Monte Carlo integration as discussed in the *simulation* step of SAEM, i.e. by Gibbs sampling the model, with parameters $\widehat{\theta}^{(B)}$ and conditional to corresponding observations $y_G^{(k)}$. Note that, if $k \in B$, in other words if the queried patient is in the training set, then, as we reported so far, $\widehat{\phi}_R^{(k|B)}$ is already an output of our learning algorithm, and it is found in the same way by equation (2.22).

Now, let A and A^c be two disjoint halves of the experimented population $\{1, ..., N\}$. To validate our method, we want to compare, for each n, the estimations $\widehat{\phi}_R^{(n|A)}$ against

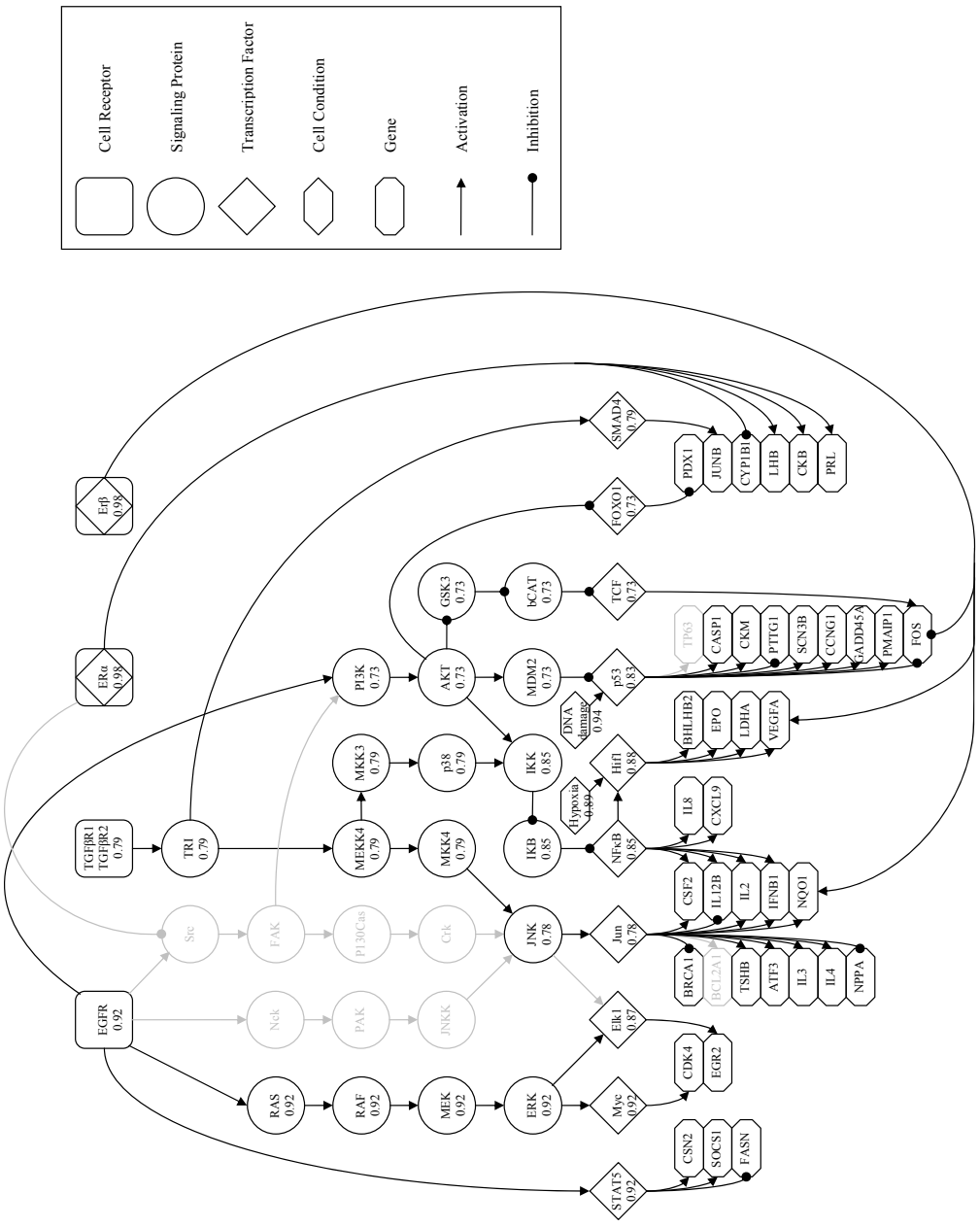

Figure 2.7: A simpler but plausible interpretation of the original core topology of Figure 2.1. Discarded components and edges are shown in light gray for comparison. Scores at each node indicate the Pearson correlation coefficient between the corresponding status estimates under the original and modified wirings.

$\widehat{\phi}_R^{(n|A^c)}$, that are predicted for the same person, but with respective model parameters $\widehat{\theta}^{(A)}$ and $\widehat{\theta}^{(A^c)}$, learned from two disjoint sets of subjects.

Table 2.3 shows the reproducibility results, where for each $r \in R$, we give the corresponding scatter plot of $\widehat{\phi}_r^{(n|A)}$ vs. $\widehat{\phi}_r^{(n|A^c)}$, for $n = 1, ..., N$, and accumulated over 20 random selections of A. Averaged over these repeated random sub-sampling experiments, the resulting correlation coefficient between predicted vectors $\left[\widehat{\phi}_r^{(n|A)}\right]_{n=1}^N$ and $\left[\widehat{\phi}_r^{(n|A^c)}\right]_{n=1}^N$ is used as a measure of fit between models learned on disjoint patient populations, showing how well the method generalizes, with even smaller learning samples.

2.4.5 Robustness under Modifications of Topology

We already discussed the invariance of statistical inference under structural perturbations such as collapsing or elongating linear pathways. Denoting the original core topology of Figure 2.1 by $\mathbb{G}$, we now examine the robustness of our model with respect to biologically realistic revisions $\widetilde{\mathbb{G}}$ which are similar to $\mathbb{G}$ but not equivalent in the previous sense of collapsed chains.

As another plausible representation of the signaling network, we consider the modified wiring diagram $\widetilde{\mathbb{G}}$ of Figure 2.7. Compared with $\mathbb{G}$, $\widetilde{\mathbb{G}}$ lacks a few genes that were originally observed, as well as several proteins and connections. Absent components and their discarded pathways are shown in light gray for visualizing the difference.

On the same gene expression data, we ran our algorithm using the revised topology $\widetilde{\mathbb{G}}$ and compared the new estimations to their counterparts found with $\mathbb{G}$. Figure 2.7 also quantifies the resulting agreement of inference for nodes that are common under both models. Attached to each v and averaged over 20 independent experiments, we give the correlation coefficient between the subject-dependent status estimations $\left[\widehat{\widehat{x}}_v^{(n|\mathbb{G})}\right]_{n=1}^N$ and $\left[\widehat{\widehat{x}}_v^{(n|\widetilde{\mathbb{G}})}\right]_{n=1}^N$ based on respective structures $\mathbb{G}$ and $\widetilde{\mathbb{G}}$, and evaluated according to (2.23). The magnitude of the correlations demonstrates the robustness of the model with respect to different wiring assumptions that are biologically reasonable.

2.4.6 Alternative Choices for Signal Transitions

Recall that the function ϕ_v returns the expected activation rate of each non-root component $v \in V \setminus R$, given the states of its parents. In the case of internal nodes, for which we assume binary variables, $\phi_v(x_{pa(v)})$ is simply the conditional success probability $P(X_v = 1 | X_{pa(v)} = x_{pa(v)})$, whereas for terminal genes, it is the multiplicative factor determining the amount of transcribed RNA in an individual cell.

So far, we presented our results with a plain and generic linear formulation for ϕ_v, as laid out in Equation (2.2). The motivation behind this over-simplified choice was primarily computational, where, thanks to Proposition 2.2.1, we exploited the parameter-free and linear form to swiftly evaluate quantities required for learning. We

now wish to explore more flexible, possibly nonlinear alternatives for ϕ_v to further validate our model.

Let us first consider another linear choice

$$\phi_v(x_{pa(v)}) = \frac{\sum_{u \in pa(v)} x_u \mathbf{1}\{u \in A_v\} + \tau(1 - x_u)\mathbf{1}\{u \in I_v\}}{|A_v| + \tau|I_v|} \tag{2.24}$$

which is slightly generalized from (2.2). The additional parameter $\tau > 0$ is again shared among all non-root nodes and it is introduced to discriminate the net effect of an inhibitor parent as compared to an activating one when both types are present. Note that, (2.24) is again a mapping from $\{0,1\}^{|pa(v)|}$ to $(0,1)$, and it boils down to the original version in (2.2) if $\tau = 1$, or if v's parents are all of the same type (e.g., activators only). On the other hand, if v has mixed types of parents, which is the case in the analyzed RAS-RAF network at the protein Src, transcription factor p53 and genes IL12B and FOS, then $\tau > 1$ corresponds to increased sensitivity to inhibition over activation, whereas $\tau \in (0,1)$ corresponds to the opposite situation.

Note that, when applied to a network with maximal pathway length l, the overall model in (2.8) now reformulated with (2.24), would involve logarithms of up to l^{th} order polynomials in τ, which is due to cell averaging assumption and thereby due to expected transcription rates ξ_g from equation (2.3). Thus, maximum likelihood estimation of τ is generally intractable. Instead we can do an exhaustive search, while optimizing receptor predictions.

Given the data set used in our previous experiments, playing with the value of τ had very little or no effect on our results. In particular, our biological validation,

namely the predictions for the ERα activation rates yielded similar rank-sum test p-values. Thus, to better assess the validity of the alternative formulation in (2.24), we switch to another data set of 278 subjects and over the same collection of genes, but now with complete ground truth information for both EGFR and ERα receptors containing sufficient amount of samples from each specific phenotype. To be precise, this second data set contains 164 ERα-positive patients versus 114 ERα-negatives, and 59 EGFR-positives as opposed to 219 EGFR-negatives. We applied our learning algorithm on these data, using the same overall model, which now involves the additional inhibition parameter τ. Since virtually no change was observed with ERα predictions on the first data set, in this experiment, we are particularly interested in inferring the EGFR node, which we can now compare to their complete ground truths.

Figure 2.8 gives rank sum test p-values for both ERα and EGFR, averaged over 20 independent experiments per choice of τ, which we increase from 0.1 to 1000. As the results show, the p-value for ERα remains low everywhere on the analyzed spectrum, which also reproduces our stable findings with the previous data set. But our EGFR predictions become significant only for τ greater than 10, their accuracy is poor with $\tau = 1$ corresponding to our previous definition (2.2). In other words, our model recovers the EGFR status well, once we represent inhibition to be substantially more powerful than activation. In fact, this argues for the validity of our model, since for conflicting signals dominance of inhibition is indeed a widely observed general rule in

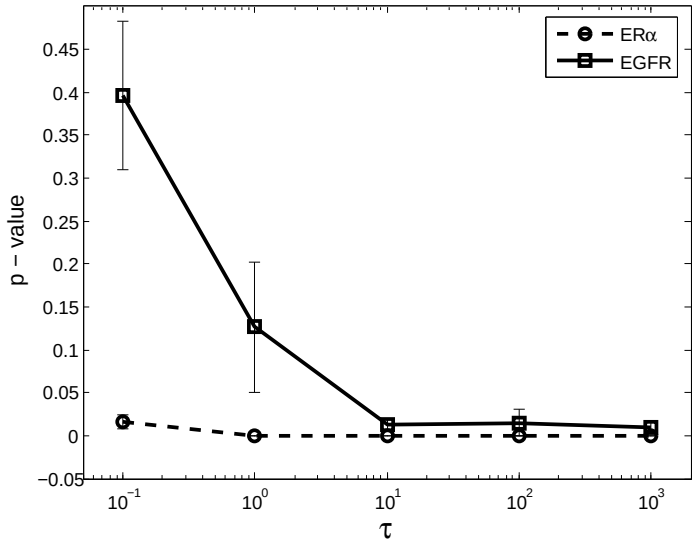

Figure 2.8: Rank-sum test p-values for predicted ERα and EGFR activation rates, obtained using (2.24) and averaged over 20 independent experiments for each choice of inhibition parameter τ. Note the improvement beyond the point $\tau = 1$, which corresponds to the original linear definition in (2.2).

biological networks [89].

These improved results obtained simply by the inclusion of the weight parameter τ, provide further insight for other possible formulations of ϕ_v. For example, let $X_v^{act} = \max_{u \in A_v} X_u$ and $X_v^{inh} = \max_{u \in I_v} X_u$ be the binary indicators of competing signals arriving at a non-root v, corresponding to its activation and inhibition, respectively. Then, let another alternative for ϕ_v be given as in the lookup Table 2.4, again for all non-root $v \in V \setminus R$ and this time using another shared parameter $\epsilon \in (0, \frac{1}{2})$. Now, ϕ_v attains the symmetric values ϵ and $1 - \epsilon$, corresponding to two extremes, while again assuming the dominance of inhibition over activation. When no signal is arriving from either types of parents, i.e., when both X_v^{act} and X_v^{inh} are zero, the state of v is

Table 2.4: Alternative ϕ_v given for different configurations of (X_v^{act}, X_v^{inh}) and parametrized with a shared parameter ϵ. N/A indicates absence of parents for the corresponding type.

X_v^{act}	X_v^{inh}	ϕ_v
0	0	$\frac{1}{2}$
0	1	ϵ
1	0	$1-\epsilon$
1	1	ϵ
0	N/A	ϵ
1	N/A	$1-\epsilon$
N/A	0	$1-\epsilon$
N/A	1	ϵ

determined in a coin flip.

Learning with this alternative formulation for ϕ_v is no longer straightforward as before, since it is not linear in the parent states as the original version (2.2) or its extension (2.24) are. In particular, the computation in (2.7) needs to be done node by node and at each iteration of SAEM.

Similar as above, we experimented with different values of ϵ ranging from 0.00001 to 0.1. Figure 2.9 shows rank-sum test p-values for the corresponding ERα and EGFR predictions, again averaged over 20 independent experiments per choice of ϵ. For $\epsilon \leq 0.01$ results are more significant than before, with corresponding p-values as

small as 2×10^{-4} for ERα and 1.5×10^{-4} for EGFR. This suggests that with reasonable hardwired parameters, EGFR status can even be better recovered compared to a supervised learner, such as naïve Bayes classifier.

The model loses the ability to recover receptor statuses for values of ϵ larger than 0.01. This is due to the length of the signaling pathways in the wiring diagram of Figure 2.1. To be precise, imagine a simple example on a linear pathway $X_0 \to X_1 \to ... \to X_L$ of length L, where each component is binary and activates its child when in "on" state. Then, as a function of the shared parameter ϵ as given in Table 2.4, we can write for the conditional expectation of the leaf X_L given the root X_0 as

$$E[X_L|X_0] = (1-2\epsilon)^L X_0 + \epsilon \sum_{l=0}^{L-1} (1-2\epsilon)^l.$$

Thus, the dependence on the root drops with increasing ϵ and the rate is exponential with the pathway length. For instance, when $\epsilon = 0.2$ and $L = 7$ as most of the pathways in the actual signaling diagram of Figure 2.1 are, we have $E[X_7|X_0 = 1] = 0.51 \approx E[X_7|X_0 = 0]$, i.e., the leaf is modeled to be very weakly dependent on the root, and its status is determined virtually in a coin flip.

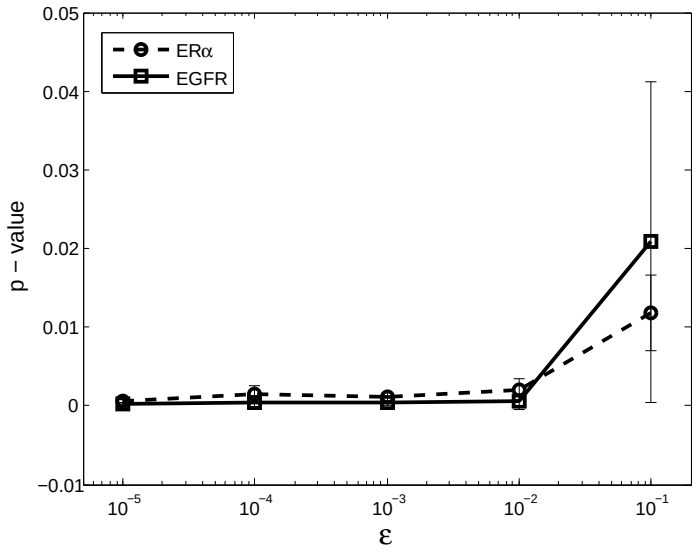

Figure 2.9: Rank-sum test p-values for predicted ERα and EGFR activation rates, obtained using lookup Table 2.4, and averaged over 20 independent experiments for each choice of ϵ.

Chapter 3

Nested Latent Variable Forest Models

3.1 Introduction

In the previous chapter, we presented a comprehensive statistical model applied to protein signaling, where, thanks to available domain knowledge, interactions were fixed in a pre-defined core topology. We now consider the more general challenge of model selection, in which the amount of such prior information is insufficient to circumscribe the combinatorial scale of the task. In such cases of interest, underlying dependency structures are unknown and thus, their discovery becomes the primary objective of learning.

When very little is known about the nature of interactions between variables, one is

usually confronted by an enormous space of possible models to search through. Even with only acyclic structures, namely Bayesian networks, the number of candidate explanations is super-exponential in the number of variables. Furthermore, the more serious problem is susceptibility to high variance and over-fitting the data, especially with high dimensional features and relatively small amount of samples available for learning. One usual way of alleviating such difficulties is to introduce priors and/or penalties to possible topologies, playing with their weights to favor candidates with fewer parameters [36]. However, simply penalizing model complexity tends to bias the estimation towards the absence of interactions, i.e., towards independence, blocking the discovery of higher order relationships that may exist. In those cases, where the dependency structure is known to be rich, it can be more effective to ease the penalties, while introducing proper structural biases and thereby severely restricting candidate explanations to a relatively small class of models. We argue that in this way, statistical learning becomes feasible due to variance reduction.

One such class with strong assumptions is the family of latent tree models, i.e., tree (or forest) structured distributions where observable variables are represented at the leaves and their joint interactions are encoded via internal latent variables. The potential of latent tree models as useful Bayesian networks is first identified by Pearl [59], and now, their use is prevalent in many areas, including computational biology and computer vision.

In addition to the vast body of work on latent tree models (see Chapter 1 for

details), our methodology and the intended scope of applications are particularly inspired by the "decomposable events" concept of Fleuret *et al.* [18], where the objective is to learn hierarchically arranged discriminative features, for recognizing objects in gray scale images. At this point, we find it more convenient to begin our construction with analogies drawn from that work. Basically, the authors' idea was to start with elementary binary random variables, or "events" as they call them (e.g., indicators of edge fragments indexed by orientation and polarity), and recursively search for their meaningful and more complex conjunctions (e.g., boundary segments, parts of object silhouettes) that are far more likely on the object class of interest. To achieve this, variables are grouped in a greedy agglomerative fashion, if they tend to occur (or not occur) together in the target class with a significant probability. In particular, groups are represented via the product of individual members, which yields another binary meta-variable. This makes a recursive stepwise approach possible, where at each step the pair with largest absolute correlation is selected and joined together, provided the magnitude of this correlation is larger than some threshold ρ. Eventually, "ρ-nested" clusters of highly dependent variables are obtained and used for discrimination.

Using similar ideas, we design and entertain here a confined class of probability models, for which the same dyadic aggregation procedure is casted as a model selection strategy. In other words, instead of feature conjunctions, we propose to learn *nested models* over dependent subsets of variables, in which case variable grouping corresponds to estimating a joint parametric model over the combined set. To en-

sure an exponential growth in the number of jointly modeled variables, we impose a *balance criterion*, where we only allow merges between comparably sized disjoint modules, with cardinalities differing by at most one. Once justified by its BIC score (similar to ρ threshold above) based on the net gain in data likelihood, each fusion will replace two independent laws by a joint distribution, which is more complex, but only involves a limited number of additional parameters. Eventually, an overall joint model will be progressively learned, which is, as in [18], "decomposable" in the sense that the distribution of every module has a representation in terms of successive fusions of smaller and smaller subsets, all the way down to individual variables.

3.1.1 Latent Variables

One important objective of this sequential learning scheme is to maintain a constant rate in parametric growth to have a control over complexity. That is, when fusing two submodels with respective numbers of parameters K_1 and K_2, we want the substituted joint model to have a total number of parameters less than $K_1 + K_2$ plus some small fixed number. An efficient way to achieve this is to accompany each fusion by a latent variable, which is invented to mediate the coupling between the merged subsets. Introducing such hidden variables can be very useful in capturing unobserved "regulatory" cues (e.g. transcription factors for observed genes, or semantic shape tokens for object images), and allows one to learn simpler models [43, 90].

In our case, the statistical utility of hidden variables comes in two ways. First,

they provide an efficient parametrization of joint models; and second, they can be used as group representatives for future fusions. When joining two subsets of observable variables, we replace their assumed *independence* by *conditional independence* given the invented hidden variable. Then, we can implement higher order fusions between larger modules by simply conditioning their hidden regulators on another hidden variable additional to the existing structure. Imposing the Markov property, we can accommodate complex dependencies at the observation level with incremental changes in the latent structure and thus can achieve a minimal amount of additional parameters at each step. As pointed out by [59], this will also allow linear time inference on the resulting tree structure, where dyadic recursions enable us to integrate out hidden variables very efficiently, yielding the sought after model over observations.

3.1.2 A Restricted Family of Models

Each step of the dyadic aggregation scheme consists of deciding "whether to do a merge", and if yes, "which available pair of clusters to merge". In other words, the task requires *model selection* from local candidates, since each possible fusion implants an alternative joint parametric model over the corresponding union formed. To be concrete, suppose, for a given step, that A, B and C are three pending distinct partitions of observable variables. Assume they have similar cardinalities, such that all three pair assignments, (A, B), (A, C) and (B, C) satisfy the balance criterion. Consequently, we have four model alternatives: the current base model

$M_0 : P(A)P(B)P(C)$ (no merge); (ii) $M_1 : P(A,B)P(C)$; (iii) $M_2 : P(A,C)P(B)$; and (iv) $M_3 : P(A)P(B,C)$. Joint probabilities $P(\cdot,\cdot)$ are evaluated via hidden variables, e.g., $P(A,B) = \sum_x P(A|x)P(B|x)P(X=x)$ for some latent X invented when joining A with B.

Clearly, the base model M_0 is nested in each of M_1, M_2 and M_3, which are in this construction the three possible "refinements" of M_0, i.e., its immediate more complex alternatives. Thus, choosing the appropriate model among these four candidates corresponds to a *local search* step within a restricted class of models, which we call "nested latent variable models" (NLVM). The focus of this chapter will be on design and analysis of the NLVM family introduced with *carefully chosen biases* like nested dyadic structures, hidden variables and the balance criterion.

Briefly, NLVM contains tree or forest structured distributions induced over the observable leaf variables. The formal characterization involves

- *Forest Representation*: The dependency structure is a forest of binary and balanced trees.
- *Markov Property*: Given its parent, each variable is conditionally independent of its non-descendants.
- *Model Variables*: Observable variables, for which training data is available. They are represented at terminal nodes (leaves) of the forest.
- *Latent Variables*: Hidden variables, which regulate the observable ones. They

are represented at non-terminal nodes of the forest.

3.1.3 Structure Discovery

The confined set of structures in NLVM renders feasible a *stepwise* discovery of dependencies among the observable variables. Starting from the product law (i.e., each individual variable is independent of others), one iteration involves a local search within the refinements of the current model. In particular, for each of the available candidates, the likelihood of training data is maximized and associated parameters are estimated using the EM algorithm. Then the "best" model is picked according to a commonly used selection criterion, BIC [42]. If no refinement scores better than the current model, which is by construction the best among all simpler versions explored so far, then the search terminates. Otherwise, the best refinement is implemented with the corresponding fusion and the search continues in the same way, now for the next more complex alternatives in NLVM. Consequently, hierarchical clusters of interacting variables are obtained with their discovered dependencies getting progressively more and more complex. When no merge is structurally possible or improving the BIC score, the final structure and its parameters are returned as the learned model from NLVM.

3.2 Nested Latent Variable Models

Let $X_O = \{X_i : i \in O\}$ denote a collection of random variables, indexed from a finite set $O = \{1, ..., D\}$. Each variable assumes values from a set $\mathcal{X}$, so that the complete configuration belongs to the product space $\Omega = \mathcal{X}^{|O|}$. While formulating the general setting, we consider that $\mathcal{X}$ is finite, i.e. each X_i is discrete.

Our overall objective is to estimate the true distribution of X_O, an unknown probability ψ on Ω. We are particularly interested in the case when the recorded samples available for training are scarce, whereas the underlying dependency structure is very rich, that is, when ψ is "far" from a product measure. For such a challenging context, we argue for restricting the search to some model class, which, in general, does not contain ψ, but hopefully contain reasonable approximations, at least at the level of important substructures of interest.

The proposed NLVM family is such a confined class, which we design in anticipation of efficient learning. The underlying graphical representations for NLVM are central to our analysis. Therefore, we start by giving some graph theoretic terminology.

3.2.1 Notation

Let $G = (V, E)$ be a directed acyclic graph (DAG), with set of vertices V and set of directed edges $E = \{(s, t) : s, t \in V\}$. For a node $s \in V$, we will use the following

notation

- $pa(s) = \{t \in V : (t, s) \in E\}$: parents of s
- $ch(s) = \{t \in V : (s, t) \in E\}$: children of s
- $sb(s) = \{t \in V : t \neq s, pa(s) \cap pa(t) \neq \emptyset\}$: siblings of s
- $td(s) = \{t \in V : ch(t) = \emptyset, \exists$ directed path from s to $t\}$: terminal descendants of s
- $tc(s) = \{t \in V \setminus td(s) : ch(t) = \emptyset, \exists$ undirected path from s to $t\}$: terminal cousins of s
- $dp(s) = \begin{cases} 0, & \text{if } pa(s) = \emptyset; \\ 1 + \min\{dp(t) : t \in pa(s)\}, & \text{otherwise.} \end{cases}$: depth of s
- $rk(s) = \begin{cases} 0, & \text{if } ch(s) = \emptyset; \\ 1 + \max\{rk(t) : t \in ch(s)\}, & \text{otherwise.} \end{cases}$: rank of s

This notation is also visualized in Figure 3.1, on an example DAG from the class of structures we employ and present in the next section.

In our graphical representations, we will label individual random variables and their sets with node correspondences given in subscripts. For instance, X_s will denote the variable at $s \in V$, whereas X_S will stand for the collection of variables carried by the set $S \subset V$ of vertices. We will use graphical notions like "parent", "child", "descendants" interchangeably for variables, as well, that is, $X_{pa(s)}$ and $X_{ch(s)}$ will

respectively denote parent and children variables of X_s. Keeping those correspondences intact, we will use small case letters for realizations and bold letters for i.i.d. populations. For example, we will write $\mathbf{x}_S = \{x_S^{(n)} : n = 1, ..., N\}$ for the observed data of collection X_S, where parenthesized superscripts will enumerate individual data points.

Unless a special notation is introduced, we will give generic probabilities with just the realizations in the argument, again node subscripts will remove any disambiguation. For instance, $P(x_s|x_t)$ will be shorthand for $P(X_s = x_s|X_t = x_t)$. The same notation will also apply to continuous random variables, in which case P will stand for a probability density.

3.2.2 Structures of Interest

The biases we introduce to the proposed model family are structural rather than parametric. Thus, it is convenient to anchor our discussion to the topologies we employ. Let $\mathcal{F}$ denote the special class of DAGs that we consider for representing dependencies amongst X_O. Formally, each forest $G = (V, E)$ from $\mathcal{F}$ is characterized as follows:

- The set $\{s \in V : ch(s) = \emptyset\}$ of G's terminal nodes is O.
- Each non-root node s has exactly one parent: $|pa(s)| = 1$.
- Each non-terminal node s has exactly two children: $|ch(s)| = 2$.

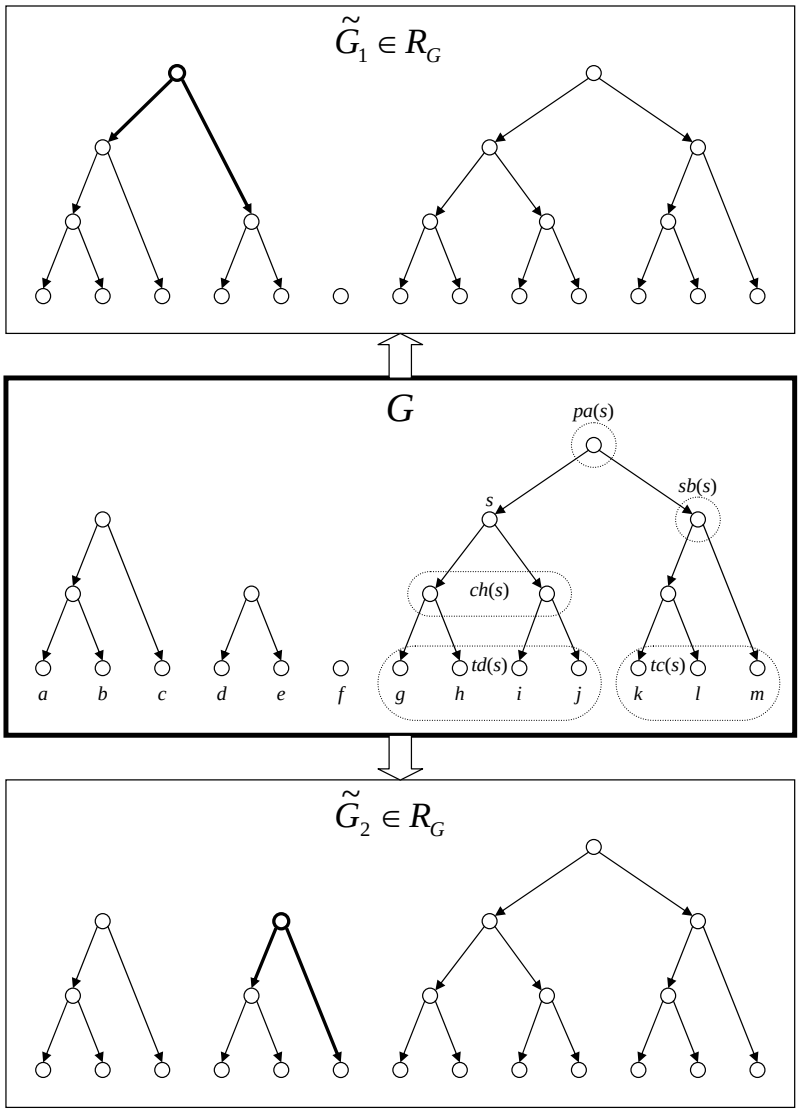

Figure 3.1: Middle: An example forest $G \in \mathcal{F}$ with $O = \{a, b, c, d, e, f, g, h, i, j, k, l, m\}$, composed of four rooted, binary and balanced trees. Parent $pa(s)$, sibling $sb(s)$, children $ch(s)$, terminal descendants $td(s)$ and terminal cousins $tc(s)$ are shown for the example node s. Top & Bottom: Possible "refinements" of G, with additional root and edges printed in bold.

- Each sibling pair $\{s, t : pa(s) = pa(t)\}$ satisfies $\big||td(s)| - |td(t)|\big| \leq 1$.

Briefly, graphs in $\mathcal{F}$ are forests composed of rooted, binary and balanced[1] trees, whose respective leaf sets partition O. Henceforth, we assume that the index set O is fixed. Like letters V and E, which we use for vertices and edges, O and $H := V \setminus O$ will by default stand for respective sets of terminal and non-terminal nodes of forests from $\mathcal{F}$.

To provide a structural guideline for our model selection framework, for each $G \in \mathcal{F}$, we also define $\mathcal{R}_G$ as the set of "refinements" $\widetilde{G} = (\widetilde{V}, \widetilde{E}) \in \mathcal{F}$, such that,

- $\widetilde{V} = V \cup \{u\}$ for some $u \notin V$;
- $\widetilde{E} = E \cup \{(u, s), (u, t)\}$, for some s, t, that are two distinct roots of G, which satisfy the balance condition $\big||td(s)| - |td(t)|\big| \leq 1$.

If G has only one connected component, i.e., if it is a single tree, then $\mathcal{R}_G$ is empty. Otherwise, $\mathcal{R}_G$ contains forests, which are derived from G by merging any two of its existing, disjoint trees, that differ in their respective leaf set cardinalities by at most one. The merge operation assigns a new parent node to their roots, yielding an aggregated tree, which is still binary, balanced and rooted at the new node introduced. Thus, $\mathcal{R}_G$ is a subset of $\mathcal{F}$. Figure 3.1 also illustrates how an example forest G is modified to its possible refinements $\mathcal{R}_G = \{\widetilde{G}_1, \widetilde{G}_2\}$, with appended root and edges printed in bold.

[1] each non-terminal node branches into two subtrees of similar size, whose respective leaf sets differ in cardinality by at most one

Note that, any $G \in \mathcal{F}$ can be constructed by iteratively refining the totally disconnected graph $G^{(0)} = (O, \emptyset)$, which is also in $\mathcal{F}$. For example, letting the k^{th} refinement $G^{(k)} \in \mathcal{R}_{G^{(k-1)}}$ be the restriction of G to its first $|O| + k$ rank-ordered nodes, we can obtain a sequence of forests growing to G.

3.2.3 Proposed Model Class

Let $G = (V, E)$ be a forest from $\mathcal{F}$, whose terminal nodes O are associated with observable variables X_O. Similarly, let X_H be an auxiliary set of hidden variables represented at non-terminal nodes H. Then, let directed edges E encode conditional independencies among $X_V = (X_O, X_H)$, with the Markov assumption that given its parent, each variable is conditionally independent of its non-descendants. Consequently, the complete set X_V forms a Bayesian network on the forest G with joint distribution factored as

$$P(x_V) = \prod_{s \in V} P(x_s | x_{pa(s)}). \tag{3.1}$$

As discussed earlier, the particular dyadic structure of G, casts each hidden variable X_s, $s \in H$, as a latent regulator, which couples two similarly sized disjoint subsets of X_O. The complexity of dependencies encoded by X_s grows with its rank $rk(s)$ in the graph. For consistency and ease of formulation, we will assume that hidden variables are also discrete.

Now, let $p_s(x_s|x_u)$ denote a parametric conditional probability at a non-root $s \in$

V, with an incident edge $(u, s) \in E$; and let $p_r(x_r)$ be a parametric unconditional probability at a root r, $pa(r) = \emptyset$. Then, indexed by its underlying topology G, let $\mathcal{M}_G$ denote the collection of all parametric distributions

$$\pi(x_O) = \sum_{x_H} \prod_{\substack{r \in V \\ pa(r)=\emptyset}} p_r(x_r) \prod_{(u,s)\in E} p_s(x_s|x_u). \tag{3.2}$$

defined over observable variables X_O and reduced from the corresponding G-structured complete Bayesian network representation, given in Equation (3.1). Model parameters θ will specify local transition probabilities $p_s(x_s|x_u)$ and their space will be denoted by Θ_G. As we will discuss next, exploiting the dyadic hierarchies in $\mathcal{F}$, we can integrate out the hidden part X_H very efficiently to evaluate π.

Our proposed model class, denoted by $\mathcal{M} = \{\mathcal{M}_G : G \in \mathcal{F}\}$, is the collection of all such models over X_O. $\mathcal{M}$ is a nested family, where for each $G \in \mathcal{F}$, the corresponding model $\mathcal{M}_G$ is a subset of its refinements $\mathcal{M}_{\widetilde{G}}$ with larger structures $\widetilde{G} \in \mathcal{R}_G$. This is evident, since the product of tree submodels for the former is a special case of the joint model obtained after fusing them in the latter. The nesting property will be crucial when we discuss our model learning procedure.

3.2.4 Dynamic Programming

In order to evaluate distributions π from $\mathcal{M}$, as well as other important quantities (e.g., posteriors of hidden variables) required for parameter estimation, we first introduce the following two probabilities, which we specify for each node $s \in V$ of the

underlying forest $G = (V, E) \in \mathcal{F}$:

- $\kappa_s(x_{td(s)}|x_s)$: Conditional probability that, given x_s, its terminal descendants are $x_{td(s)}$.
- $\lambda_s(x_{tc(s)}, x_s)$: Joint probability of x_s with its terminal cousins $x_{tc(s)}$.

Recall that nodes in $tc(s)$ are leaves from the same tree that contains s, but are not descendants of s. Thus, $tc(s)$ and $td(s)$ partition the terminal nodes of the tree component, to which s belongs. Clearly, if s is a root, then $tc(s)$ is empty. Keeping these relations in mind, κ_s and λ_s are computed recursively with the following lemma, which is closely related to the Belief Propagation algorithm [59].

Lemma 3.2.1. *For a given node $s \in V$, let $ch(s) = \{q, r\}$, $sb(s) = \{t\}$ and $pa(s) = \{u\}$, unless they are empty (see Figure 3.2). Then the following recursive relations hold*

$$(i) \quad \kappa_s(x_{td(s)}|x_s) = \begin{cases} \mathbf{1}\{x_{td(s)} = x_s\}, & \text{if } ch(s) = \emptyset; \\ \sum_{x_q} \kappa_q(x_{td(q)}|x_q) p_q(x_q|x_s) & \\ \quad \times \sum_{x_r} \kappa_r(x_{td(r)}|x_r) p_r(x_r|x_s), & \text{otherwise.} \end{cases}$$

$$(ii) \quad \lambda_s(x_{tc(s)}, x_s) = \begin{cases} p_s(x_s), & \text{if } pa(s) = \emptyset; \\ \sum_{x_u} \big[\lambda_u(x_{tc(u)}, x_u) p_s(x_s|x_u) & \\ \quad \times \sum_{x_t} \kappa_t(x_{td(t)}|x_t) p_t(x_t|x_u)\big], & \text{otherwise.} \end{cases}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function.

Proof. For part (i), if $s \in S$, then $td(s) = \{s\}$ and the result is straightforward. Otherwise, noting that $td(s) = td(q) \cup td(r)$ and using the Markov property of the network, we can write

$$\begin{aligned}
\kappa_s(x_{td(s)}|x_s) &= \sum_{x_q}\sum_{x_r} P(x_{td(q)}, x_{td(r)}, x_q, x_r|x_s) \\
&= \sum_{x_q}\sum_{x_r} P(x_{td(q)}, x_{td(r)}|x_q, x_r, x_s)P(x_q, x_r|x_s) \\
&= \sum_{x_q}\sum_{x_r} P(x_{td(q)}|x_q, x_r, x_s)P(x_{td(r)}|x_q, x_r, x_s)p_q(x_q|x_s)p_r(x_r|x_s) \\
&= \sum_{x_q} \kappa_q(x_{td(q)}|x_q)p_q(x_q|x_s) \sum_{x_r} \kappa_r(x_{td(r)}|x_r)p_r(x_r|x_s).
\end{aligned}$$

For part (ii), if $pa(s) = \emptyset$, then $tc(s) = \emptyset$ and the result is straightforward. Otherwise, noting that $tc(s) = td(t) \cup tc(u)$ and again with network's Markov assumptions, we can write

$$\begin{aligned}
\lambda_s(x_{tc(s)}, x_s) &= \sum_{x_t}\sum_{x_u} P(x_{td(t)}, x_{tc(u)}, x_s, x_t, x_u) \\
&= \sum_{x_t}\sum_{x_u} P(x_{td(t)}|x_{tc(u)}, x_s, x_t, x_u)P(x_{tc(u)}, x_s, x_t, x_u) \\
&= \sum_{x_t}\sum_{x_u} \kappa_t(x_{td(t)}|x_t)P(x_s|x_{tc(u)}, x_t, x_u)P(x_{tc(u)}, x_t, x_u) \\
&= \sum_{x_t}\sum_{x_u} \kappa_t(x_{td(t)}|x_t)p_s(x_s|x_u)P(x_t|x_{tc(u)}, x_u)\lambda_u(x_{tc(u)}, x_u) \\
&= \sum_{x_u} \lambda_u(x_{tc(u)}, x_u)p_s(x_s|x_u) \sum_{x_t} \kappa_t(x_{td(t)}|x_t)p_t(x_t|x_u).
\end{aligned}$$

□

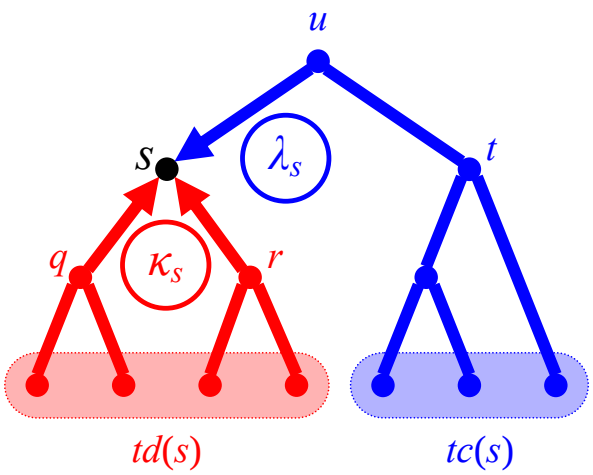

Figure 3.2: Recursion paths of κ_s and λ_s, which relate variables at terminal nodes $td(s)$ and $tc(s)$ to the one at s via their counterparts found for children $\{q, r\}$, sibling t and parent u.

Figure 3.2 illustrates the computations of κ_s and λ_s with corresponding recursion paths arriving at node s like "messages" from terminal nodes $td(s)$ and $tc(s)$. As a result, the Bayesian network $X_V = (X_O, X_H)$ on $G = (V, E) \in \mathcal{F}$, has not only a compactly written joint distribution by its very definition, but it also allows one to integrate out hidden variables X_H via dynamic programming. In particular, let U be a representative subset of V, which contains one node selected per each independent tree of G (e.g., $U = \{s : pa(s) = \emptyset\} \subset V$ can be set of roots of G). Then, the joint distribution π over the observable variables X_O, can be written as

$$\pi(x_O) = \prod_{u \in U} \sum_{x_s} \kappa_s(x_{td(s)}|x_s)\lambda_s(x_{tc(s)}, x_s). \tag{3.3}$$

3.3 Learning

For inference, let $\mathbf{x}_O = \{x_O^{(n)} : n = 1, ..., N\}$ be an available i.i.d. sample of size N. Given training data $\mathbf{x}_O$, our objective is to learn from the proposed model family $\mathcal{M}$, a parametric distribution π that is as "close" as possible to the unknown true distribution ψ of collection X_O. In general, our confined class of candidate explanations does not contain ψ, but it is rich enough to provide reasonable approximations, at least at the level of important substructures of interest. Exploiting the nesting property of $\mathcal{M}$, the dyadic aggregation procedure discussed previously, becomes an efficient model selection framework based on stepwise discovery of dependencies.

Our learning algorithm involves both parameter estimation and structure discovery. We explore candidate topologies, while we simultaneously estimate corresponding parameters by maximizing the likelihood of observations $\mathbf{x}_O$. At each step, the selected new model fuses two pending sub-models and replaces their product with a joint distribution learned over the combined set. In this way, we achieve a hierarchical refinement of hypotheses on variable dependencies.

In particular, we start with the product measure over X_O, assuming each variable is independent of others. Accordingly, we initialize our learning algorithm with $\langle G^{(0)}, \theta^{(0)} \rangle$, where $G^{(0)} = (V^{(0)}, E^{(0)}) := (O, \emptyset)$ is a totally disconnected graph in $\mathcal{F}$, and $\theta^{(0)}$ is the vector of corresponding initial parameters directly estimable from data. Then, at iteration $k \geq 0$, having arrived at $\langle G^{(k)}, \theta^{(k)} \rangle$ with the current dependency structure $G^{(k)}$ and maximum likelihood (ML) parameters $\theta^{(k)}$ found for the

corresponding model $M_{G^{(k)}}$, we examine possible refinements $\mathcal{M}_{\widetilde{G}}$ with aggregated topologies $\widetilde{G} \in \mathcal{R}_{G^{(k)}}$.

First, we estimate their ML parameters from data, which constitutes the *model identification* routine of the algorithm. Then, using maximized likelihoods, we evaluate scores for all available candidates $\{\mathcal{M}_{G^{(k)}}, \mathcal{M}_{\widetilde{G}} : \widetilde{G} \in \mathcal{R}_{G^{(k)}}\}$ and perform a local *model selection* based on the Bayesian information criterion (BIC).

The algorithm terminates and returns $\langle G^{(k)}, \theta^{(k)} \rangle$ whenever the model $\mathcal{M}_{G^{(k)}}$ is justified to be better than its refinements, or $\mathcal{R}_{G^{(k)}}$ is empty. Next, we give details for model identification and model selection procedures.

3.3.1 Model Identification

Given model $\mathcal{M}_G$ with fixed topology $G \in \mathcal{F}$, this stage involves maximum likelihood parameter estimation using data $\mathbf{x}_O$. With hidden variables, the standard approach for model identification is the Expectation Maximization (EM) algorithm [38]. Briefly, EM starts with some initial parameters and iteratively improves them by maximizing the conditional expectation of complete data log-likelihood, given i.i.d. observations $\mathbf{x}_O$. In order to avoid confusion with the step count k used for our sequential model search, we will omit to write loop indices for EM when we discuss it below and later in the context of particular parametric examples. Instead, we will formulate a single iteration of EM like a function, that maps current parameters θ to updated ones $\widehat{\theta}$.

In particular, given the structure G, and known parameters θ for $\mathcal{M}_G$, a single iteration of EM returns updated parameters $\widehat{\theta}$ via the following two steps

- **E-step** Write the objective $Q(\widetilde{\theta}|\theta) = E[\log P(\mathbf{X}_O, \mathbf{X}_H|\widetilde{\theta})|\mathbf{x}_O; \theta]$, which is the conditional expectation of complete data log-likelihood written as a function of $\widetilde{\theta}$, and evaluated with known current parameters θ and i.i.d. observations $\mathbf{x}_O$.
- **M-step** Solve for the maximizer $\widehat{\theta} = \text{argmax}_{\widetilde{\theta}} Q(\widetilde{\theta}|\theta)$.

Then, setting current parameters θ to new ones $\widehat{\theta}$, EM alternates between these operations until convergence is evident.

EM is repeatedly executed for each new candidate model, which is, by our search strategy, a one-step refinement of an already processed simpler version. Recall that each refinement implements a joint distribution for two previously independent modules, now joining their trees by a new hidden variable. Independent of this fusion, the rest of the overall structure remains the same, and so do the corresponding ML parameters that are estimated previously. Thus, model identification is required only for the new tree formed, nodes of which are enough to compose the "complete data" of E-step. We will make use of this modularity during model selection as well.

3.3.2 Model Selection

Using data likelihoods as the sole model selection criterion would always favor the most complex candidate, which results in over-fitting the data. This classical

problem of model selection is heavily studied in various contexts. A well principled way to penalize complexity is to introduce priors for candidate models as well as their parameters. But when there is partial observability, as in our case, such Bayesian methods become intractable, and some kind of approximation is usually put in place [36, 40].

Current literature embodies a wide range of selection criteria (e.g., AIC [41], AICc [91], BIC [42], SRM [92]), which are proposed for situations like ours and others. For a good review the reader is referred to [93, 94]. In our case, we prefer the Schwarz criterion [42], also known as the Bayesian information criterion (BIC), since its formulation decomposes into substructures and it incorporates training sample size, so that the model discovery can adapt itself when more data are available.

In particular, for a model with K parameters and likelihood L maximized over N data points, BIC is defined as

$$\mathrm{BIC} = -2 \log L + K \log N, \tag{3.4}$$

where, equivalent to the minimum description length approach, smaller values of BIC mean "better" fit. The $K \log N$ term is the penalty, which involves both the model complexity and available sample size.

In our case, $\{\mathcal{M}_{G^{(k)}}, \mathcal{M}_{\widetilde{G}} : \widetilde{G} \in \mathcal{R}_{G^{(k)}}\}$ is the set of candidate models at a given step $k \geq 0$, where $G^{(k)}$ is the current structure, for which we already have the maximized

likelihood $L_{G^{(k)}} = \sup_{\mathcal{M}_{G^{(k)}}} \pi(\mathbf{x}_O)$. Then, let

$$\rho_{\widetilde{G}} = \log \Lambda_{\widetilde{G}} - \Delta K_{\widetilde{G}} \log \sqrt{N} \tag{3.5}$$

be the relative score we compute for refinement $\mathcal{M}_{\widetilde{G}}$ after finding its ML parameters, where $\Lambda_{\widetilde{G}} = \frac{L_{\widetilde{G}}}{L_{G^{(k)}}}$ is the ratio of maximum likelihoods under $\mathcal{M}_{\widetilde{G}}$ and the current model $\mathcal{M}_{G^{(k)}}$, and $\Delta K_{\widetilde{G}} = K_{\widetilde{G}} - K_{G^{(k)}}$ is the difference of their respective numbers of parameters. Then, choosing the model with lowest BIC (3.4) corresponds to doing

- Find $\widehat{G} = \arg\max_{\widetilde{G} \in \mathcal{R}_{G^{(k)}}} \rho_{\widetilde{G}}$
- Set $G^{(k+1)} = \begin{cases} \widehat{G}, & \text{if } \rho_{\widehat{G}} > 0; \\ G^{(k)}, & \text{otherwise.} \end{cases}$

In summary, we first determine the best refinement. Then, if it is justified to be better than the current model, it is accepted as the next one, otherwise it is rejected, in which case the algorithm terminates and returns model $\mathcal{M}_{G^{(k)}}$.

Note that, by modularity, the likelihood ratio $\Lambda_{\widetilde{G}}$ and increase in complexity $\Delta K_{\widetilde{G}}$ of Equation (3.5) are deducible from the particular pair of trees in $G^{(k)}$, which are joined together in the new forest $\widetilde{G}$. Consequently, $\rho_{\widetilde{G}}$ depends only on the fusion, and not on the rest of the structure, which is the same for both $G^{(k)}$ and $\widetilde{G}$. Therefore, fusions with negative $\rho_{\widetilde{G}}$ at step k can be discarded from future candidates $\mathcal{R}_{G^{(l)}}, l > k$ in advance. Note that this would not be possible with other sample size dependent selection criteria like AICc [91], for which the formulation is not decomposable into independent tree components and therefore scores need to be recomputed from scratch

at every step of the search and for each particular fusion, even it has been examined before.

3.4 NLVM with Bernoulli Regulation

For binary valued variables X_O, we introduce a specific case from NLVM, which we denote "NLVM-Bern" and symbolize with $\mathcal{M}^{\text{Bern}}$. Dependency structures in this particular subfamily are from $\mathcal{F}$ as usual, but we consider binary hidden variables and encode parent-child interactions accordingly with conditional Bernoulli distributions. Before going into details, we will first elaborate on *identifiability* of $\mathcal{M}^{\text{Bern}}$, which is the desired property that different parameter values yield different distributions over X_O, so that inference can be possible.

Let us consider the simple example with two binary valued observable variables X_a and X_b that are conditionally independent given a third one X_c, which is hidden and also binary. The corresponding dependency structure is the tree $a \leftarrow c \rightarrow b$. The parameters for the resulting Bayesian network (X_a, X_b, X_c) are conditional success probabilities $\theta = \big(p_a(1|0), p_a(1|1), p_b(1|0), p_b(1|1), p_c(1)\big)$ specifying transitions between variables. Since the observable pair (X_a, X_b) can assume four different configurations, their bivariate joint distribution $p_{ab}(x_a, x_b)$ has three degrees of freedom, i.e., smaller than θ's dimensionality. Thus, reducing p_{ab} from the complete Bayesian network, i.e., writing it as $\sum_{x_c} p_a(x_a|x_c)p_b(x_b|x_c)p_c(x_c)$ is an over-parametrized representation. Consequently, θ fails to be identifiable in this case.

The redundancy demonstrated in this simple example can be generalized to arbitrary structures in $\mathcal{M}^{\text{Bern}}$. In the case of binary variables, it is superfluous to represent the joint distribution of depth one siblings via Bernoulli conditionals given

their hidden root parent and an unconditional root probability. The same statistical coupling can be implemented using fewer parameters by a bivariate probability mass function (pmf), which is written unconditionally and specified by any three of the four point probabilities involved. In the case of binary variables, replacing the usual Bayesian network parameters by bivariate pmfs at depth one, not only reduces complexity, but it also removes parametric non-identifiability, which we will discuss next. However, to be consistent with our usual recipe on tree aggregation and forest growth, we will not change our definition for underlying topologies. We will keep our graph related notation with respect to our initial DAG formulation, as if the non-terminal root nodes are still intact, although their hidden variables no longer appear in the modified decomposition.

Incorporating this adjustment, we specify a model $\mathcal{M}_G^{\text{Bern}}$ with topology $G = (V, E) \in \mathcal{F}$, via the reduced set of parameters

$$\theta = \begin{pmatrix} \big(p_s(1)\big)_{s\in V: pa(s)=ch(s)=\emptyset} \\ \big(p_s(1|0), p_s(1|1)\big)_{s\in V: dp(s)>1} \\ \big(p_{st}(0,0), p_{st}(0,1), p_{st}(1,0)\big)_{s,t\in V: pa(s)=pa(t), dp(s)=dp(t)=1} \end{pmatrix} \tag{3.6}$$

Figure 3.3 shows a graphical illustration for $\mathcal{M}^{\text{Bern}}$, where undirected edges printed in green represent the bivariate joint pmfs used to couple depth one siblings, replacing their univariate conditionals at light gray edges under the usual Bayesian network formulation.

With bivariate parameters p_{st} of (3.6), Equation (3.3) for π still holds, where λ-

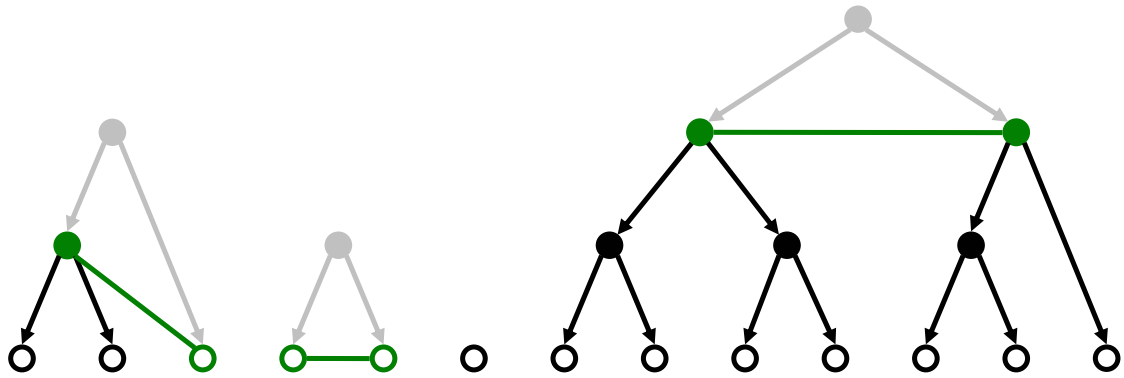

Figure 3.3: Revised graphical interpretation for NLVM-Bern: undirected edges printed in green encode the joint pmfs of depth 1 siblings for the identifiable parametrization, substituting the usual Bayesian network parameters at the light gray edges and non-terminal roots.

recursions stated in Lemma 3.2.1, are now initialized from depth one siblings, say s and t, for which we write $\lambda_s(x_{tc(s)}, x_s) = \sum_{x_t} p_{st}(x_s, x_t)\kappa_t(x_{td(t)}|x_t)$ and $\lambda_t(x_{tc(t)}, x_t) = \sum_{x_s} p_{st}(x_s, x_t)\kappa_s(x_{td(s)}|x_s)$.

3.4.1 Nesting in NLVM-Bern

Model nesting in NLVM-Bern still holds after discarding hidden root variables from the latent network and reducing the usual Bayesian netwotrk parametrization. But for clarity and completeness, it may need to be made explicit here. Let θ be the fixed parameters under the model $\mathcal{M}_G^{\text{Bern}}$ with dependency structure $G \in \mathcal{F}$. Now, derived from θ we provide detailed expressions for new parameters $\widetilde{\theta}$ under the refined model $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$ structured with $\widetilde{G} \in \mathcal{R}_G$, such that distributions $\pi(\cdot|\theta) \in \mathcal{M}_G^{\text{Bern}}$ and $\pi(\cdot|\widetilde{\theta}) \in \mathcal{M}_{\widetilde{G}}^{\text{Bern}}$ over X_O become identical. This will verify the nesting property in

NLVM-Bern, but also provide a recipe on how to initialize the EM parameter learning algorithm for the refined model $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$, when we switch from the coarse one $\mathcal{M}_{G}^{\text{Bern}}$.

Suppose e and f are two distinct roots in G that are joined in $\widetilde{G}$ as children of a new node g (see Figure 3.5). If node e (resp. f) is terminal, then θ contains the parameter $p_e(1)$ (resp. $p_f(1)$). In that case, let $\widetilde{p}_e(1) = p_e(1)$ (resp. $\widetilde{p}_f(1) = p_f(1)$). Otherwise, if node e is non-terminal with children $ch(e) = \{a, b\}$, then θ of simpler model $\mathcal{M}_{G}^{\text{Bern}}$ contains bivariate parameters p_{ab} for siblings X_a and X_b. In that case, let $\delta = p_{ab}(0,0)p_{ab}(1,1) - p_{ab}(0,1)p_{ab}(1,0)$, and for $\delta \neq 0$ let

$$\phi = \begin{cases} \frac{p_{ab}(0,0)+p_{ab}(1,1)+\sqrt{(p_{ab}(0,0)-p_{ab}(1,1))^2+4\delta}}{2}, & \text{if } \delta > 0; \\ \frac{p_{ab}(0,1)+p_{ab}(1,0)+\sqrt{(p_{ab}(0,1)-p_{ab}(1,0))^2-4\delta}}{2}, & \text{if } \delta < 0. \end{cases}$$

Then, we can introduce

$$\widetilde{p}_a(1|0) = \begin{cases} \frac{p_{ab}(1,0)}{p_{ab}(1,0)+\phi}, & \text{if } \delta > 0; \\ \frac{p_{ab}(1,1)}{p_{ab}(1,1)+\phi}, & \text{if } \delta < 0; \\ p_{ab}(1,0) + p_{ab}(1,1), & \text{otherwise.} \end{cases}$$

$$\widetilde{p}_a(1|1) = \begin{cases} \frac{\phi}{p_{ab}(0,1)+\phi}, & \text{if } \delta > 0; \\ \frac{\phi}{p_{ab}(0,0)+\phi}, & \text{if } \delta < 0; \\ p_{ab}(1,0) + p_{ab}(1,1), & \text{otherwise.} \end{cases}$$

$$\widetilde{p}_b(1|0) = \begin{cases} \frac{p_{ab}(0,1)}{p_{ab}(0,1)+\phi}, & \text{if } \delta > 0; \\ \frac{\phi}{p_{ab}(0,0)+\phi}, & \text{if } \delta < 0; \\ p_{ab}(0,1) + p_{ab}(1,1), & \text{otherwise.} \end{cases}$$

$$\widetilde{p}_b(1|1) = \begin{cases} \frac{\phi}{p_{ab}(1,0)+\phi}, & \text{if } \delta > 0; \\ \frac{p_{ab}(1,1)}{p_{ab}(1,1)+\phi}, & \text{if } \delta < 0; \\ p_{ab}(0,1) + p_{ab}(1,1), & \text{otherwise.} \end{cases}$$

$$\widetilde{p}_e(1) = \begin{cases} \frac{\phi(\phi - p_{ab}(0,0))}{\phi^2 - p_{ab}(0,1)p_{ab}(1,0)}, & \text{if } \delta > 0; \\ \frac{\phi(\phi - p_{ab}(0,1))}{\phi^2 - p_{ab}(0,0)p_{ab}(1,1)}, & \text{if } \delta < 0; \\ \frac{1}{2}, & \text{otherwise.} \end{cases}$$

such that $p_{ab}(x_a, x_b) = \sum_{x_e} \widetilde{p}_a(x_a|x_e)\widetilde{p}_b(x_b|x_e)\widetilde{p}_e(x_e)$ holds for all $x_a, x_b \in \{0, 1\}$.

Since ϕ is strictly positive by definition, it is clear that $\widetilde{p}_a(1|0)$, $\widetilde{p}_a(1|1)$, $\widetilde{p}_b(1|0)$ and $\widetilde{p}_b(1|1)$ are all probabilities on the unit interval. To confirm the same for $\widetilde{p}_e(1)$, first, note that for $\delta \neq 0$ we have the inequality

$$\phi > \begin{cases} \frac{p_{ab}(0,0)+p_{ab}(1,1)+|p_{ab}(0,0)-p_{ab}(1,1)|}{2} = \max(p_{ab}(0,0), p_{ab}(1,1)), & \text{if } \delta > 0; \\ \frac{p_{ab}(0,1)+p_{ab}(1,0)+|p_{ab}(0,1)-p_{ab}(1,0)|}{2} = \max(p_{ab}(0,1), p_{ab}(1,0)), & \text{if } \delta < 0. \end{cases}$$

Then, for $\delta > 0$, we can write

$$
\begin{aligned}
&\phi^2 - p_{ab}(0,1)p_{ab}(1,0) \quad \text{(denominator of } \widetilde{p}_e(1)\text{)} \\
&= \phi(\phi - p_{ab}(0,0)) + (\phi p_{ab}(0,0) - p_{ab}(0,1)p_{ab}(1,0)) \\
&> \phi(\phi - p_{ab}(0,0)) + (\max(p_{ab}(0,0), p_{ab}(1,1))p_{ab}(0,0) - p_{ab}(0,1)p_{ab}(1,0)) \\
&\geq \phi(\phi - p_{ab}(0,0)) + (p_{ab}(0,0)p_{ab}(1,1) - p_{ab}(0,1)p_{ab}(1,0)) \\
&= \phi(\phi - p_{ab}(0,0)) + \delta \\
&> \phi(\phi - p_{ab}(0,0)) \quad \text{(numerator of } \widetilde{p}_e(1)\text{)} \\
&> \phi(\max(p_{ab}(0,0), p_{ab}(1,1)) - p_{ab}(0,0)) \\
&\geq 0
\end{aligned}
$$

and similar inequalities for $\delta < 0$, which delivers the result.

Analogously, if f is non-terminal with children $ch(f) = \{c, d\}$, then let $\widetilde{p}_c(1|0)$, $\widetilde{p}_c(1|1)$, $\widetilde{p}_d(1|0)$, $\widetilde{p}_d(1|1)$ and $\widetilde{p}_f(1)$ be written in terms of available bivariate parameters p_{cd} of θ, the same way as above with subscripts a, b, e now replaced by c, d, f. Again $p_{cd}(x_c, x_d) = \sum_{x_f} \widetilde{p}_c(x_c|x_f)\widetilde{p}_d(x_d|x_f)\widetilde{p}_f(x_f)$ will be satisfied for all $x_c, x_d \in \{0, 1\}$. Finally, let $\widetilde{p}_{ef}(x_e, x_f) = \widetilde{p}_e(x_e)\widetilde{p}_f(x_f)$ for $x_e, x_f \in \{0, 1\}$.

Replacing θ's components p_{ab}, p_{cd}, p_e and p_f by the new ones $\widetilde{p}_a$, $\widetilde{p}_b$, $\widetilde{p}_c$, $\widetilde{p}_d$ and $\widetilde{p}_{ef}$, we can obtain a $\widetilde{\theta}$, which satisfies $\mathcal{M}_G^{\text{Bern}} \ni \pi(\cdot|\theta) = \pi(\cdot|\widetilde{\theta}) \in \mathcal{M}_{\widetilde{G}}^{\text{Bern}}$.

3.4.2 Identifiability in NLVM-Bern

In order to address parametric identifiability in NLVM-Bern, let us first give some definitions that will be useful in our analysis. Given two (sets of) random variables X_A and X_B, let

$$\Delta_{A;B}(x_A, x_B) = P(x_A, x_B) - P(x_A)P(x_B) \tag{3.7}$$

denote the difference between their joint distribution and the product of their marginals. Suppose X_A and X_B are conditionally independent given a binary master variable X_m that follows a Bernoulli distribution p_m. Accordingly, let $p_A(\cdot|x_m)$ and $p_B(\cdot|x_m)$ stand for the respective conditional distributions of X_A and X_B given $X_m = x_m \in \{0, 1\}$. Then, defining

$$\eta_A^m(x_A) = p_A(x_A|1) - p_A(x_A|0) \tag{3.8}$$

and η_B^m analogously, as the differences of conditional distributions given $X_m = 1$ and $X_m = 0$, we can write

$$\begin{aligned}\Delta_{A;B}(x_A; x_B) &= \sum_{x_m} p_A(x_A|x_m)p_B(x_B|x_m)p_m(x_m) \\ &\qquad - \sum_{y_m} p_A(x_A|y_m)p_m(y_m) \sum_{z_m} p_B(x_B|z_m)p_m(z_m) \\ &= p_m(0)p_m(1)\eta_A(x_A)\eta_B(x_B).\end{aligned} \tag{3.9}$$

Furthermore, unless X_A and X_B are independent, there exists a reference configuration (x_A^*, x_B^*), for which $\Delta_{A;B}$ is nonzero, such that we can define the ratio

$$\nu_A(x_A) = \frac{\Delta_{A;B}(x_A; x_B^*)}{\Delta_{A;B}(x_A^*; x_B^*)} = \frac{\eta_A^m(x_A)}{\eta_A^m(x_A^*)} \tag{3.10}$$

and ν_B analogously, which are respective functions of x_A and x_B, only.

Using these quantities, we can now examine identifiability of models in NLVM-Bern inductively. We first state the following lemma, which establishes the result for a simple case and serves as the basis for larger and arbitrary structures.

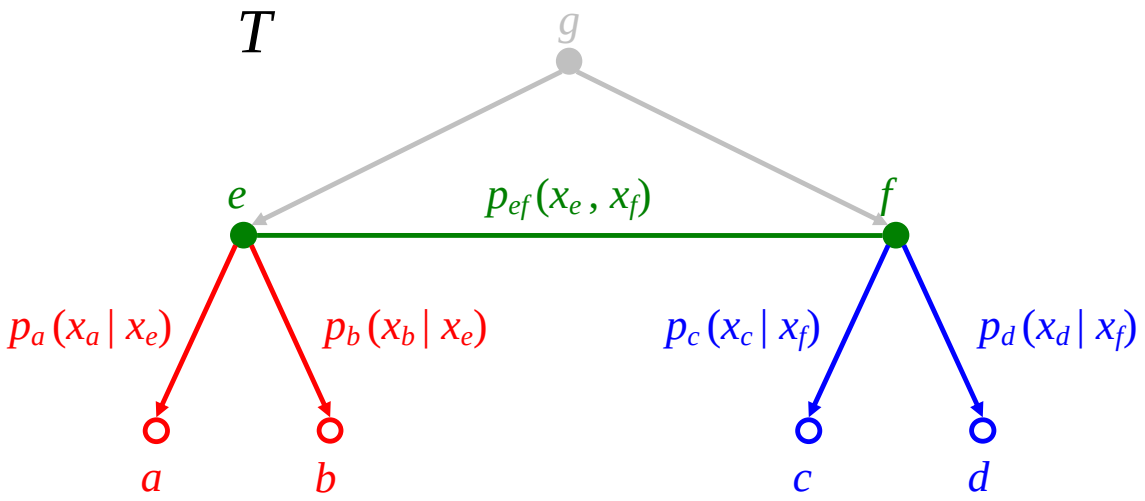

Figure 3.4: Tree structure $T \in \mathcal{F}$ analyzed in Lemma 3.4.1 for NLVM-Bern with identifiable probabilities $p_a(x_a|x_e)$, $p_b(x_b|x_e)$, $p_c(x_c|x_f)$, $p_d(x_d|x_f)$ and $p_{ef}(x_e, x_f)$ (Non-terminal root g and its outgoing edges that are bypassed for reduced parametrization are shown in light gray)

Lemma 3.4.1. *Let $\{X_a, X_b, X_c, X_d\}$ be a collection of binary random variables that are regulated by another pair $\{X_e, X_f\}$ of binary latent variables. Let their dependency structure be given by the tree $T = (V, E) \in \mathcal{F}$, as shown in Figure 3.4, with vertices*

$V = \{a, b, c, d, e, f, g\}$ and directed edges $E = \{(g, e), (g, f), (e, a), (e, b), (f, c), (f, d)\}$. Suppose that each variable is dependent on its sibling. Then, from the joint distribution

$$\pi(x_a, x_b, x_c, x_d|\theta) = \sum_{x_e, x_f} p_a(x_a|x_e)p_b(x_b|x_e)p_c(x_c|x_f)p_d(x_d|x_f)p_{ef}(x_e, x_f) \tag{3.11}$$

over $\{X_a, X_b, X_c, X_d\}$, the parameters

$$\theta = \big(p_a(1|0), p_a(1|1), p_b(1|0), p_b(1|1), p_c(1|0), p_c(1|1), p_d(1|0), p_d(1|1),$$
$$p_{ef}(0, 0), p_{ef}(0, 1), p_{ef}(1, 0)\big)$$

given as in (3.6)*, are identifiable, up to binary inversions of the latent states of X_e and/or X_f.*

Proof. Given a distribution π from the model in (3.11), we can proceed sequentially, to identify

- $\theta_{ab} = \big(p_a(1|0), p_a(1|1), p_b(1|0), p_b(1|1)\big)$ on the left subtree $a \leftarrow e \rightarrow b$
- $\theta_{cd} = \big(p_c(1|0), p_c(1|1), p_d(1|0), p_d(1|1)\big)$ on the right subtree $c \leftarrow f \rightarrow d$
- $\theta_{ef} = \big(p_{ef}(0, 0), p_{ef}(0, 1), p_{ef}(1, 0)\big)$ on the link between the subtrees.

as shown in Figure 3.5. This can be done separately in three steps, as follows:

(i) From the joint distribution π, we can directly compute $\Delta_{ab;cd} = \pi - \pi_{ab}\pi_{cd}$ and $\Delta_{a;b} = \pi_{ab} - \pi_a\pi_b$, where π_{ab} and π_{cd} are obtained from π as the bivariate marginals of pairs $\{X_a, X_b\}$ and $\{X_c, X_d\}$; and π_a and π_b are univariate marginals of X_a and X_b, respectively.

Given the binary variable X_e, the model in (3.11) asserts conditional independence between pairs $\{X_a, X_b\}$ and $\{X_c, X_d\}$; as well as between individual variables X_a and X_b. Thus, to find θ_{ab}, we can consider the reduced decomposition

$$\pi(x_a, x_b, x_c, x_d|\theta) = \sum_{x_e} p_a(x_a|x_e)p_b(x_b|x_e)p_{cd}(x_c, x_d|x_e)p_e(x_e), \tag{3.12}$$

which involves a single latent variable X_e with the univariate Bernoulli law $p_e(x_e) = \sum_{x_f} p_{ef}(x_e, x_f)$, written as a function of θ_{ef} (Note that, unlike X_a and X_b, X_c and X_d may be still dependent given X_e, thus, their joint conditional distribution is written as the generic bivariate $p_{cd}(x_c, x_d|x_e)$, which however, is immaterial to finding θ_{ab}).

Thus, as in (3.9), we can express the differences $\Delta_{ab;cd}$ and $\Delta_{a;b}$ with

$$\Delta_{ab;cd}(x_a, x_b; x_c, x_d) = p_e(0)p_e(1)\eta^e_{ab}(x_a, x_b)\eta^e_{cd}(x_c, x_d) \tag{3.13}$$

$$\Delta_{a;b}(x_a; x_b) = p_e(0)p_e(1)\eta^e_a(x_a)\eta^e_b(x_b), \tag{3.14}$$

where η^e_{ab}, η^e_{cd}, η^e_a and η^e_b are defined analogously to (3.8) (e.g., $\eta^e_a(x_a) = p_a(x_a|1) - p_a(x_a|0)$ and $\eta^e_{cd}(x_c, x_d) = p_{cd}(x_c, x_d|1) - p_{cd}(x_c, x_d|0)$), such that right hand sides above are entirely composed of individual terms of the reduced representation in (3.12).

It is easy to verify from (3.7) that

$$\Delta_{a;b}(x_a; x_b) = -\Delta_{a;b}(1 - x_a; x_b) = -\Delta_{a;b}(x_a; 1 - x_b) = \Delta_{a;b}(1 - x_a; 1 - x_b)$$

for all $x_a, x_b \in \{0, 1\}$. Thus, by the dependence assumption between X_a and X_b, $\Delta_{a;b}$ never vanishes and by (3.14), this implies nonzero quantities p_e, η^e_a and η^e_b.

Similarly, by the dependence between the pairs $\{X_a, X_b\}$ and $\{X_c, X_d\}$, which follows from the dependence of siblings X_e and X_f, we can assume without loss of generality, that $\Delta_{ab;cd}(0,0;0,0)$ is nonzero. Then, as in (3.10), we can define $\nu_{ab}(x_a, x_b) = \Delta_{ab;cd}(x_a, x_b; 0, 0)/\Delta_{ab;cd}(0,0;0,0)$, which is directly obtained from π. Accordingly, we have

$$\eta^e_{ab}(x_a, x_b) - \nu_{ab}(x_a, x_b)\eta^e_{ab}(0,0) = 0. \tag{3.15}$$

Then, using the relations

$$p_a(x_a|0) = p_a(x_a|0)\big(p_e(0) + p_e(1)\big) \pm p_a(x_a|1)p_e(1) = p_a(x_a) - \eta^e_a(x_a)p_e(1), \tag{3.16}$$

$$p_b(x_b|0) = p_b(x_b|0)\big(p_e(0) + p_e(1)\big) \pm p_b(x_b|1)p_e(1) = p_b(x_a) - \eta^e_b(x_b)p_e(1) \tag{3.17}$$

and the conditional independence of X_a and X_b given X_e, we can express η^e_{ab} further, in terms of η^e_a and η^e_b:

$$\begin{aligned}
\eta^e_{ab}(x_a, x_b) &= p_a(x_a|1)p_b(x_b|1) - p_a(x_a|0)p_b(x_b|0) \\
&= \big(\eta^e_a(x_a) + p_a(x_a|0)\big)\big(\eta^e_b(x_b) + p_b(x_b|0)\big) - p_a(x_a|0)p_b(x_b|0) \\
&= \eta^e_a(x_a)\eta^e_b(x_b) + \eta^e_a(x_a)p_b(x_b|0) + \eta^e_b(x_b)p_a(x_a|0) \\
&= \eta^e_a(x_a)\eta^e_b(x_b) + \eta^e_a(x_a)\big(p_b(x_b) - \eta^e_b(x_b)p_e(1)\big) + \eta^e_b(x_b)\big(p_a(x_a) - \eta^e_a(x_a)p_e(1)\big) \\
&= \eta^e_a(x_a)\eta^e_b(x_b)\big(p_e(0) - p_e(1)\big) + \eta^e_a(x_a)p_b(x_b) + \eta^e_b(x_b)p_a(x_a) \\
&= \Delta_{a;b}(x_a; x_b)\frac{p_e(0) - p_e(1)}{p_e(0)p_e(1)} + \eta^e_a(x_a)p_b(x_b) + \eta^e_b(x_b)p_a(x_a),
\end{aligned} \tag{3.18}$$

where the last line replaces the product $\eta^e_a\eta^e_b$ using (3.14).

Evaluating Equation (3.15) at the particular configurations $(x_a, x_b) = (0, 1)$ and $(x_a, x_b) = (1, 0)$, with η^e_{ab} terms written as in (3.18), and noting that $\eta^e_a(1) = -\eta^e_a(0)$ and $\eta^e_b(1) = -\eta^e_b(0)$, we can finally obtain a 2×2 linear system

$$\mathbf{A}(\eta^e_a(0), \eta^e_b(0))^\top = \frac{p_e(0) - p_e(1)}{p_e(0)p_e(1)}\mathbf{c} \tag{3.19}$$

in the unknowns $\eta^e_a(0)$ and $\eta^e_b(0)$, where

$$\mathbf{A} = \begin{pmatrix} 1 - p_b(0)\big(1 + \nu_{ab}(0,1)\big) & -p_a(0)\big(1 + \nu_{ab}(0,1)\big) \\ -p_b(0)\big(1 + \nu_{ab}(1,0) & 1 - p_a(0)\big(1 + \nu_{ab}(1,0)\big) \end{pmatrix}$$

and

$$\mathbf{c} = \begin{pmatrix} \nu_{ab}(0,1)\Delta_{a;b}(0,0) - \Delta_{a;b}(0,1) \\ \nu_{ab}(1,0)\Delta_{a;b}(0,0) - \Delta_{a;b}(1,0) \end{pmatrix}.$$

are directly obtainable from π. Since $\eta^e_a(x_a) = \sum_{x_b} \eta^e_{ab}(x_a, x_b)$ and $\eta^e_b(x_b) = \sum_{x_a} \eta^e_{ab}(x_a, x_b)$, we have

$$\begin{aligned} \det \mathbf{A} &= 1 - p_a(0)\big(1 + \nu_{ab}(1,0)\big) - p_b(0)\big(1 + \nu_{ab}(0,1)\big) \\ &= 1 - p_a(0)\frac{\eta^e_{ab}(0,0) + \eta^e_{ab}(1,0)}{\eta^e_{ab}(0,0)} - p_b(0)\frac{\eta^e_{ab}(0,0) + \eta^e_{ab}(0,1)}{\eta^e_{ab}(0,0)} \\ &= 1 - p_a(0)\frac{\eta^e_b(0)}{\eta^e_{ab}(0,0)} - p_b(0)\frac{\eta^e_a(0)}{\eta^e_{ab}(0,0)}. \end{aligned}$$

As a result, we end up with the following two cases:

- If $\det \mathbf{A} = 0$, then $\eta^e_{ab}(0,0) = p_a(0)\eta^e_b(0) + p_b(0)\eta^e_a(0)$. By Equation (3.18), that would imply $p_e(0) = p_e(1) = 1/2$. In that case, from the first linear equation of (3.19), we can write

$$\eta^e_a(0) = \frac{p_a(0)\big(1 + \nu_{ab}(0,1)\big)}{1 - p_b(0)\big(1 + \nu_{ab}(0,1)\big)}\eta^e_b(0). \tag{3.20}$$

Putting this into (3.14), we obtain a quadratic equation in the only remaining unknown $\eta_b^e(0)$, with the solution

$$\eta_b^e(0) = \pm 2\sqrt{\frac{1 - p_b(0)\big(1 + \nu_{ab}(0,1)\big)}{p_a(0)\big(1 + \nu_{ab}(0,1)\big)}\Delta_{a;b}(0,0)}, \tag{3.21}$$

which also gives $\eta_a^e(0)$ explicitly from (3.20). Since a solution must exist by construction (i.e. π already belongs to the model in (3.11)), the expression inside the square root of (3.21) is positive, which can be verified easily by non-vanishing $\Delta_{a;b}$ and vanishing $\det \mathbf{A}$.

- If $\det \mathbf{A} \neq 0$, the system in (3.19) gives

$$(\eta_a^e(0), \eta_b^e(0))^\top = \frac{p_e(0) - p_e(1)}{p_e(0)p_e(1)}\mathbf{A}^{-1}\mathbf{c}. \tag{3.22}$$

Plugging this into Equation (3.14), writing $p_e(1) = 1 - p_e(0)$, and letting K denote the product of components of the vector $\mathbf{A}^{-1}\mathbf{c}$, we obtain a quadratic equation in the only unknown $p_e(0)$, with the solution

$$p_e(0) = \frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{\Delta_{a;b}(0,0)}{4K + \Delta_{a;b}(0,0)}}, \tag{3.23}$$

which also gives $\eta_a^e(0)$ and $\eta_b^e(0)$ explicitly form (3.22). Again, the positivity of the expression inside the square root follows by construction, and can be confirmed in a way similar to the previous case, with nonzero $\Delta_{a;b}$ and $\det \mathbf{A}$.

With quantities $p_e(0) = 1 - p_e(1)$, $\eta_a^e(0) = -\eta_a^e(1)$ and $\eta_b^e(0) = -\eta_b^e(1)$, we can obtain explicit values for $p_a(1|0)$ and $p_b(1|0)$ from equations (3.16) and (3.17), respectively. Then, $p_a(1|1) = \eta_a^e(1) + p_a(1|0)$ and $p_b(1|1) = \eta_b^e(1) + p_b(1|0)$ follow directly.

Both cases based on $\det \mathbf{A}$ yield exactly two solutions $\theta_{ab}^{(1)}$ and $\theta_{ab}^{(2)}$. By (3.21) or (3.23), these solutions are symmetric, in the sense that they correspond to complementary interpretations of the binary latent variable X_e. In particular, $\theta_{ab}^{(1)}$ and $\theta_{ab}^{(2)}$ are componentwise permutations of each other: $p_a^{(1)}(1|x_e) = p_a^{(2)}(1|1-x_e)$ and $p_b^{(1)}(1|x_e) = p_b^{(2)}(1|1-x_e)$ for $x_e \in \{0,1\}$.

(ii) Due to symmetry of subtrees $a \leftarrow e \rightarrow b$ and $c \leftarrow f \rightarrow d$, identification of θ_{cd} is entirely analogous to part (i). Similarly, there are two solutions $\theta_{cd}^{(1)}$ and $\theta_{cd}^{(2)}$ corresponding to complementary interpretations of the binary latent variable X_f, where $p_c^{(1)}(1|x_f) = p_c^{(2)}(1|1-x_f)$ and $p_d^{(1)}(1|x_f) = p_d^{(2)}(1|1-x_f)$ for $x_f \in \{0,1\}$.

(iii) By the dependence between X_a and X_b, as well as between X_c and X_d, there exists configurations $(x_a^*, x_b^*) \in \{0,1\}^2$ and $(x_c^*, x_d^*) \in \{0,1\}^2$, such that $\eta_{ab}^e(x_a^*, x_b^*) = p_a(x_a^*|1)p_b(x_b^*|1) - p_a(x_a^*|0)p_b(x_b^*|0)$ and $\eta_{cd}^f(x_c^*, x_d^*) = p_c(x_c^*|1)p_d(x_d^*|1) - p_c(x_c^*|0)p_d(x_d^*|0)$ are both nonzero. Then, with explicit values of θ_{ab} and θ_{cd}, as well as $p_e(0)$ and $p_f(0)$ found in the previous steps, we can write the linear system

$$
\begin{aligned}
p_e(0) &= p_{ef}(0,0) + p_{ef}(0,1) \\
p_f(0) &= p_{ef}(0,0) + p_{ef}(1,0) \\
p_{ef}(1,1) &= 1 - p_{ef}(0,0) - p_{ef}(0,1) - p_{ef}(1,0) \\
\pi(x_a^*, x_b^*, x_c^*, x_d^*|\theta) &= \sum_{x_e, x_f} p_a(x_a^*|x_e)p_b(x_b^*|x_e)p_c(x_c^*|x_f)p_d(x_d^*|x_f)p_{ef}(x_e, x_f)
\end{aligned}
$$

which finally gives θ_{ef} as

$$p_{ef}(0,0) = \frac{\pi(x_a^*, x_d^*, x_c^*, x_d^*|\theta) - p_a(x_a^*|1)p_b(x_b^*|1)p_c(x_c^*|1)p_d(x_d^*|1)}{\eta_{ab}^e(x_a^*, x_b^*)\eta_{cd}^f(x_c^*, x_d^*)} \tag{3.24}$$

$$+ \frac{p_f(0)p_a(x_a^*|1)p_b(x_b^*|1)}{\eta_{ab}^e(x_a^*, x_b^*)} + \frac{p_e(0)p_c(x_c^*|1)p_d(x_d^*|1)}{\eta_{cd}^f(x_c^*, x_d^*)} \tag{3.25}$$

$$p_{ef}(0,1) = p_e(0) - p_{ef}(0,0) \tag{3.26}$$

$$p_{ef}(1,0) = p_f(0) - p_{ef}(0,0). \tag{3.27}$$

As a result, we can arrive at exactly four possible $\theta = (\theta_{ab}, \theta_{cd}, \theta_{ef})$ (two due to each subtree) that, as claimed, correspond to binary inversions of latent variables X_e and X_f. □

The dependency structure T discussed in Lemma 3.4.1 is of particular importance, in that it is the smallest strictly balanced tree (i.e., its immediate subtrees are of the same exact size), for which it is nontrivial to analyze identifiability. By a similar analysis, one can prove the same result for the weakly balanced versions, where either e or f is terminal.

Using the result from 3.4.1, we can now state the following proposition on identifiability of parameters for arbitrary models in NLVM-Bern.

Proposition 3.4.1. *For models in NLVM-Bern, where, observable or not, each variable is dependent on its sibling, the parameters given in* (3.6) *are identifiable, up to binary inversions of hidden variables.*

Proof. For a given model $\mathcal{M}_G^{\text{Bern}}$ with dependency structure $G \in \mathcal{F}$, the result is either

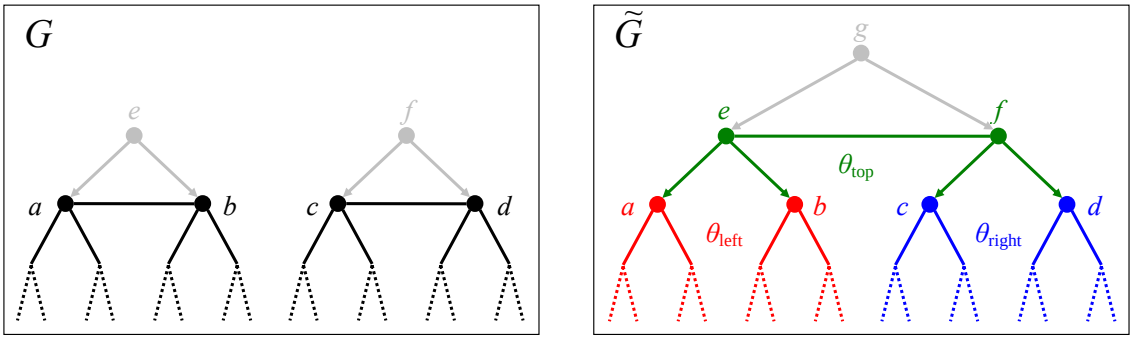

Figure 3.5: Model refinement in NLVM-Bern: The refined topology $\widetilde{G}$ on the right is derived from G on the left. Red, green and blue colors on $\widetilde{G}$ are to indicate parameters that can be identified separately under $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$. θ_{top} are the additional parameters. Non-terminal roots and their outgoing edges, which are bypassed in the reduced parametrization, are shown in light gray

straightforward or analogous to Lemma 3.4.1, if independent substructures of G are no larger than the tree T assumed for the simple case in (3.11). Otherwise, suppose, the claim holds for $\mathcal{M}_G^{\text{Bern}}$. Then, let $\widetilde{G} \in \mathcal{R}_G$ be given as in Figure 3.5 bottom, with the additional root node g that joins distinct roots e and f of G. We consider the general case when both e and f are non-terminal with respective children $ch(e) = \{a, b\}$ and $ch(f) = \{c, d\}$. Cases with terminal e and/or f, are already contained in the first statement.

Using Lemma 3.4.1, we will show that the hypothesis on the smaller model $\mathcal{M}_G^{\text{Bern}}$ implies the claim for the refined one $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$. The only topological difference between these two models is the new tree formed in $\widetilde{G}$. Remaining structures are the same with a common decomposition, and their parametric identifiability under $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$ follows by hypothesis. Thus, we can confine the proof to the new joint distribution $\pi_{td(g)}$ under

$\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$, which is implanted for the variables $X_{td(g)} = (X_{td(e)}, X_{td(f)})$ terminal to the new tree rooted at g.

Let $\theta = (\theta_{\text{left}}, \theta_{\text{right}}, \theta_{\text{top}})$ be the parameters of $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$ specifying $\pi_{td(g)}$, where θ_{left} and θ_{right} correspond to two subnetworks below the respective pairs $\{X_a, X_b\}$ and $\{X_c, X_d\}$; whereas θ_{top} are associated with the transitions above $\{X_a, X_b, X_c, X_d\}$ (see Figure 3.5). We will show the claim sequentially for θ_{left}, θ_{right} and θ_{top}.

The result follows for θ_{left} and θ_{right} directly by induction hypothesis: Given $\pi_{td(g)}$, the marginal distribution $\pi_{td(e)} = \sum_{x_{td(f)}} \pi_{td(g)}$ is readily obtainable over variables $X_{td(e)}$, which are terminal to the left subtree below g. The decomposition of $\pi_{td(e)}$ under the refined model $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$ is given by $\sum_{x_a, x_b} P(x_{td(e)}|x_a, x_b)P(x_a, x_b)$, which is also a representation under the smaller model $\mathcal{M}_{G}^{\text{Bern}}$. Thus, the identifiability claim readily holds for parameters θ_{left}, which specify $P(x_{td(e)}|x_a, x_b)$. In the same way, we can obtain the result for θ_{right}, this time considering the marginal distribution $\pi_{td(f)}$ over terminal variables $X_{td(f)}$ of the right subtree below g.

To show identifiability of the remaining parameters θ_{top}, we use Lemma 3.4.1. We can generalize that lemma, in particular, part (iii) of its proof to the current case. To be precise, suppose $\{a_1, a_2\} \subset td(a)$, $\{b_1, b_2\} \subset td(b)$, $\{c_1, c_2\} \subset td(c)$ and $\{d_1, d_2\} \subset td(d)$ comprise eight distinct nodes, selected from terminal descendants of a, b, c and d, such that observable variables X_{a_1} and X_{a_2} are conditionally independent given the latent one X_a; and similarly observable variables X_{b_1} and X_{b_2} are conditionally independent given X_b, and so on (see Figure 3.6). Then, using the Markov property,

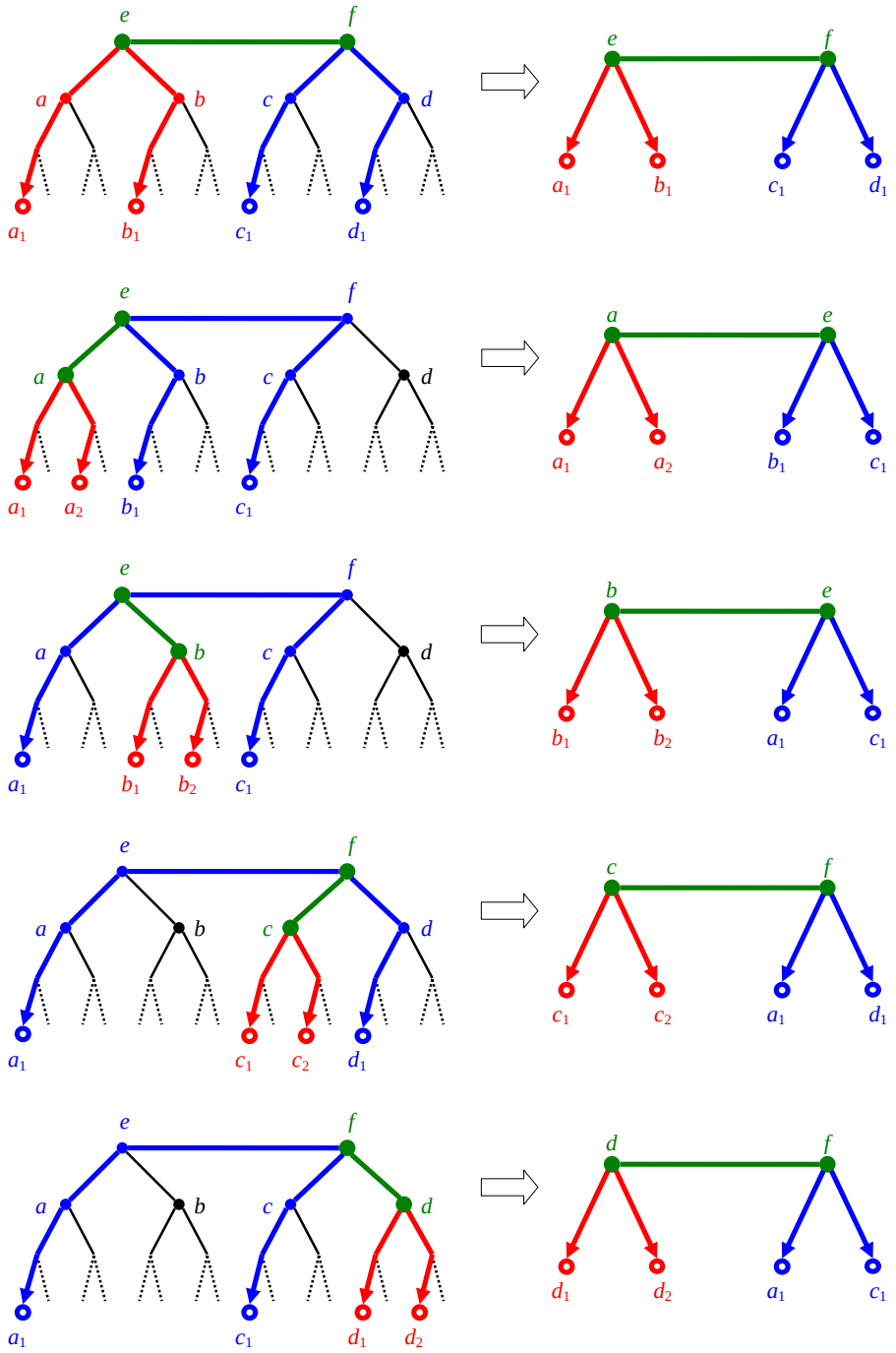

Figure 3.6: Reductions of the Markov structure among designated nodes in $\widetilde{G}$ to the tree of Lemma 3.4.1 for identifying probabilities along edges indicated with green.

we can write under $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$

$$\pi_{a_1b_1c_1d_1}(x_{a_1}, x_{b_1}, x_{c_1}, x_{d_1}|\theta) = \sum_{x_e,x_f} P(x_{a_1}|x_e)P(x_{b_1}|x_e)P(x_{c_1}|x_f)P(x_{d_1}|x_f)P(x_e, x_f)$$

$$\pi_{a_1a_2b_1c_1}(x_{a_1}, x_{a_2}, x_{b_1}, x_{c_1}|\theta) = \sum_{x_a,x_e} P(x_{a_1}|x_a)P(x_{a_2}|x_a)P(x_{b_1}|x_e)P(x_{c_1}|x_e)P(x_a, x_e)$$

$$\pi_{b_1b_2a_1c_1}(x_{b_1}, x_{b_2}, x_{a_1}, x_{c_1}|\theta) = \sum_{x_b,x_e} P(x_{b_1}|x_b)P(x_{b_2}|x_b)P(x_{a_1}|x_e)P(x_{c_1}|x_e)P(x_b, x_e)$$

$$\pi_{c_1c_2a_1d_1}(x_{c_1}, x_{c_2}, x_{a_1}, x_{d_1}|\theta) = \sum_{x_c,x_f} P(x_{c_1}|x_c)P(x_{c_2}|x_c)P(x_{a_1}|x_f)P(x_{d_1}|x_f)P(x_c, x_f)$$

$$\pi_{d_1d_2a_1c_1}(x_{d_1}, x_{d_2}, x_{a_1}, x_{c_1}|\theta) = \sum_{x_d,x_f} P(x_{d_1}|x_d)P(x_{d_2}|x_d)P(x_{a_1}|x_f)P(x_{c_1}|x_f)P(x_d, x_f),$$

where, in each case, the marginals at the left hand sides are directly obtainable from $\pi_{td(g)}$ by summing out everything but the designated four arguments. Note that, each of the five decompositions are from the tree representation laid out in Equation (3.11) of Lemma 3.4.1. By the dependence assumption between siblings in $\widetilde{G}$, the conditions are similarly satisfied for each of the quadruplets used. Thus, by part (iii) of Lemma 3.4.1, the bivariate distributions $P(x_e, x_f)$, $P(x_a, x_e)$, $P(x_b, x_e)$, $P(x_c, x_f)$ and $P(x_d, x_f)$ are identifiable, up to binary inversions of their hidden variables. The first of these pairwise joints directly gives p_{ef}, whereas the remaining four conditional parameters of θ_{top} are found by $p_a(x_a|x_e) = \frac{P(x_a,x_e)}{\sum_{x_a} P(x_a,x_e)}$, $p_b(x_b|x_e) = \frac{P(x_b,x_e)}{\sum_{x_b} P(x_b,x_e)}$, $p_c(x_c|x_f) = \frac{P(x_c,x_f)}{\sum_{x_c} P(x_c,x_f)}$ and $p_d(x_d|x_f) = \frac{P(x_d,x_f)}{\sum_{x_d} P(x_d,x_f)}$, which completes the proof.

□

Note that, the identifiability of parameters for NLVM-models can be easily improved to a one-to-one relation $\theta \leftrightarrows \pi(\cdot|\theta)$ (i.e., parameters θ that give rise to

$\pi(\cdot|\theta) \in \mathcal{M}_G^{\text{Bern}}$ can be made unique) with simple additional constraints on the parameter space. For example, imposing $p_s(1|0) \leq p_s(1|1)$ for the first member s of each sibling pair $\{s,t\}$ of depth greater than one, would force hidden variables to have a non-negative correlation with their left child, thus removing the ambiguity on their interpretation.

3.4.3 EM for NLVM-Bern

Given i.i.d. training data $\mathbf{x}_O = \{x_s^{(n)}; s \in O, n = 1, ..., N\}$, a fixed forest $G = (V, E) \in \mathcal{F}$ and known parameters θ as in equation (3.6), a single iteration of EM returns updated parameters $\widehat{\theta}$ for model $\mathcal{M}_G^{\text{Bern}}$ as follows (these will be derived in the next section):

(i) For each $s \in V$, such that $pa(s) = ch(s) = \emptyset$,

$$\widehat{p}_s(1) = \frac{1}{N} \sum_n x_s^{(n)}; \tag{3.28}$$

(ii) For each $(u, s) \in E$, such that $dp(s) > 1$, and each $x_u \in \{0, 1\}$,

$$\begin{aligned} \widehat{p}_s(1|x_u) = &\left(\sum_n \frac{\kappa_s(x_{td(s)}^{(n)}|1) p_s(1|x_u)}{\sum_{y_s} \kappa_s(x_{td(s)}^{(n)}|y_s) p_s(y_s|x_u)} \frac{\kappa_u(x_{td(u)}^{(n)}|x_u) \lambda_u(x_{tc(u)}^{(n)}, x_u)}{\sum_{y_u} \kappa_u(x_{td(u)}^{(n)}|y_u) \lambda_u(x_{tc(u)}^{(n)}, y_u)} \right) \\ &\times \left(\sum_n \frac{\kappa_u(x_{td(u)}^{(n)}|x_u) \lambda_u(x_{tc(u)}^{(n)}, x_u)}{\sum_{y_u} \kappa_u(x_{td(u)}^{(n)}|y_u) \lambda_u(x_{tc(u)}^{(n)}, y_u)} \right)^{-1} \end{aligned} \tag{3.29}$$

(iii) For each sibling pair $\{s, t\} \in V$, such that $dp(s) = dp(t) = 1$, and each $x_s, x_t \in$

$\{0,1\}$,

$$\widehat{p}_{st}(x_s,x_t)=\frac{1}{N}\sum_n \frac{\kappa_s(x_{td(s)}|x_s)\kappa_t(x_{td(t)}|x_t)p_{st}(x_s,x_t)}{\sum_{y_s,y_t}\kappa_s(x_{td(s)}|y_s)\kappa_t(x_{td(t)}|y_t)p_{st}(y_s,y_t)} \tag{3.30}$$

3.4.3.1 Derivation of EM for NLVM-Bern

Given an i.i.d. sample $\mathbf{x}_O$ and current parameters θ, the objective function of EM is given by

$$Q(\widetilde{\theta}|\theta)=\sum_n\sum_{y_O,y_H} P(y_O,y_H|x_O^{(n)};\theta)\log P(x_O^{(n)},y_H|\widetilde{\theta})$$

with complete log-likelihood

$$\log P(x_O,x_H|\widetilde{\theta})=\sum_{\substack{s\in V:\\ pa(s)=ch(s)=\emptyset}}\log\widetilde{p}_s(x_s)+\sum_{\substack{(u,s)\in E:\\ pa(u)\neq\emptyset}}\log\widetilde{p}_s(x_s|x_u)+\sum_{\substack{s,t\in V:\\ pa(s)=pa(t)\\ dp(s)=dp(t)=1}}\log\widetilde{p}_{st}(x_s,x_t).$$

which is written as a function of $\widetilde{\theta}$ to be optimized. When taking derivatives of Q, we consider the missing data posterior $P(y_H|x_O^{(n)};\theta)$ to be further expanded to $\sum_{y_O}P(y_O,y_H|x_O^{(n)};\theta)$ in order to arrive at common expressions for both terminal and non-terminal variables. Then, for the update rules, we have:

(i) (3.28) is straightforward, since for observed singletons, ML estimates of success probabilities are empirical averages.

(ii) To arrive at (3.29), first let $x_s,x_u\in\{0,1\}$ be fixed and let $\bar{x}_s=1-x_s$. Differentiating Q with respect to probability $\widetilde{p}_s(x_s|x_u)$, while noting that $\widetilde{p}_s(\bar{x}_s|x_u)=$

$1 - \widetilde{p}_s(x_s|x_u)$, we get

$$\begin{aligned}\frac{\partial Q}{\partial \widetilde{p}_s(x_s|x_u)} &= \sum_n \sum_{y_s, y_u} P(y_s, y_u|x_O^{(n)};\theta)\left(\frac{\mathbf{1}\{y_s = x_s, y_u = x_u\}}{\widetilde{p}_s(x_s|x_u)} - \frac{\mathbf{1}\{y_s = \bar{x}_s, y_u = x_u\}}{1 - \widetilde{p}_s(x_s|x_u)}\right) \\ &= \sum_n \frac{P(x_s, x_u|x_O^{(n)};\theta)}{\widetilde{p}_s(x_s|x_u)} - \sum_n \frac{P(\bar{x}_s, y_u|x_O^{(n)};\theta)}{1 - \widetilde{p}_s(x_s|x_u)}\end{aligned}$$

which vanishes at

$$\begin{aligned}\widehat{p}_s(x_s|x_u) &= \frac{\sum_n P(x_s, x_u|x_O^{(n)};\theta)}{\sum_n P(x_s, x_u|x_O^{(n)};\theta) + P(\bar{x}_s, x_u|x_O^{(n)};\theta)} \\ &= \frac{\sum_n P(x_s|x_O^{(n)}, x_u;\theta) P(x_u|x_O^{(n)};\theta)}{\sum_n P(x_u|x_O^{(n)};\theta)} \\ &= \frac{\sum_n P(x_s|x_{td(s)}^{(n)}, x_u;\theta) P(x_u|x_{td(u)}^{(n)}, x_{tc(u)}^{(n)};\theta)}{\sum_n P(x_u|x_{td(u)}^{(n)}, x_{tc(u)}^{(n)};\theta)}.\end{aligned}$$

Then, by network's Markov property and Bayes' rule, we can further write

$$\begin{aligned}P(x_s|x_{td(s)}, x_u;\theta) &= \frac{P(x_{td(s)}, x_s|x_u;\theta)}{\sum_{y_s} P(x_{td(s)}, y_s|x_u;\theta)} = \frac{\kappa_s(x_{td(s)}|x_s) p_s(x_s|x_u)}{\sum_{y_s} \kappa_s(x_{td(s)}|y_s) p_s(y_s|x_u)} \\ P(x_u|x_{td(u)}, x_{tc(u)};\theta) &= \frac{P(x_{td(u)}, x_{tc(u)}, x_u|\theta)}{\sum_{y_u} P(x_{td(u)}, x_{tc(u)}, y_u|\theta)} = \frac{\kappa_u(x_{td(u)}|x_u)\lambda_u(x_{tc(u)}, x_u)}{\sum_{y_u} \kappa_u(x_{td(u)}|y_u)\lambda_u(x_{tc(u)}, y_u)}.\end{aligned}$$

providing the result.

(iii) To arrive at (3.30), again, first let $x_s, x_t \in \{0, 1\}$ be fixed and let $\bar{x}_s = 1 - x_s$ and $\bar{x}_t = 1 - x_t$. Differentiating Q with respect to point probabilities $\widetilde{p}_{st}(x_s, x_t)$, $\widetilde{p}_{st}(\bar{x}_s, x_t)$ and $\widetilde{p}_{st}(x_s, \bar{x}_t)$, while noting that $\widetilde{p}_{st}(\bar{x}_s, \bar{x}_t) = 1 - \widetilde{p}_{st}(x_s, x_t) - \widetilde{p}_{st}(\bar{x}_s, x_t) -$

$\widetilde{p}_{st}(x_s, \bar{x}_t)$, we get

$$\frac{\partial Q}{\partial \widetilde{p}_{st}(x_s, x_t)} = \sum_n \sum_{y_s, y_t} P(y_s, y_t | x_O^{(n)}; \theta) \Bigg(\frac{\mathbf{1}\{y_s = x_s, y_t = x_t\}}{\widetilde{p}_{st}(x_s, x_t)} - \frac{\mathbf{1}\{y_s = \bar{x}_s, y_t = \bar{x}_t\}}{1 - \widetilde{p}_{st}(x_s, x_t) - \widetilde{p}_{st}(\bar{x}_s, x_t) - \widetilde{p}_{st}(x_s, \bar{x}_t)} \Bigg)$$
$$= \sum_n \frac{P(x_s, x_t | x_O^{(n)}; \theta)}{\widetilde{p}_{st}(x_s, x_t)} - \sum_n \frac{P(\bar{x}_s, \bar{x}_t | x_O^{(n)}; \theta)}{1 - \widetilde{p}_{st}(x_s, x_t) - \widetilde{p}_{st}(\bar{x}_s, x_t) - \widetilde{p}_{st}(x_s, \bar{x}_t)}$$

along with $\partial Q / \partial \widetilde{p}_{st}(\bar{x}_s, x_t)$ and $\partial Q / \partial \widetilde{p}_{st}(x_s, \bar{x}_t)$ written similarly with arguments x_s and x_t in the first sum replaced by $\bar{x}_s$ and $\bar{x}_t$, respectively. Then, the resulting gradient vanishes at

$$\widehat{p}_{st}(x_s, x_t) = \frac{1}{N} \sum_n P(x_s, x_t | x_O^{(n)}; \theta) = \frac{1}{N} \sum_n P(x_s, x_t | x_{td(s)}^{(n)}, x_{td(t)}; \theta)$$

along with $\widehat{p}_{st}(\bar{x}_s, x_t)$ and $\widehat{p}_{st}(x_s, \bar{x}_t)$ written similarly again with arguments replaced. Then, again by network's Markov assumptions and Bayes' rule, we can write

$$\begin{aligned} P(x_s, x_t | x_{td(s)}, x_{td(t)}; \theta) &= \frac{P(x_{td(s)}, x_{td(t)}, x_s, x_t | \theta)}{\sum_{y_s, y_t} P(x_{td(s)}, x_{td(t)}, y_s, y_t | \theta)} \\ &= \frac{P(x_{td(s)} | x_{td(t)}, x_s, x_t; \theta) P(x_{td(t)}, x_s, x_t | \theta)}{\sum_{y_s, y_t} P(x_{td(s)} | x_{td(t)}, y_s, y_t; \theta) P(x_{td(t)}, y_s, y_t | \theta)} \\ &= \frac{\kappa_s(x_{td(s)} | x_s) P(x_{td(t)} | x_s, x_t; \theta) p_{st}(x_s, x_t)}{\sum_{y_s, y_t} \kappa_s(x_{td(s)} | y_s) P(x_{td(t)} | y_s, y_t; \theta) p_{st}(y_s, y_t)} \\ &= \frac{\kappa_s(x_{td(s)} | x_s) \kappa_t(x_{td(t)} | x_t) p_{st}(x_s, x_t)}{\sum_{y_s, y_t} \kappa_s(x_{td(s)} | y_s) \kappa_t(x_{td(t)} | y_t) p_{st}(y_s, y_t)} \end{aligned}$$

providing the result.

3.5 NLVM with Linear Gaussian Regulation

For real valued variables X_O, we now consider a Gaussian family from NLVM, which we call "NLVM-Gauss" and symbolize with $\mathcal{M}^{\text{Gauss}}$. Each model $\mathcal{M}_G^{\text{Gauss}}$ has again a forest structure $G = (V, E) \in \mathcal{F}$, now with real valued hidden variables X_H, and represents parent-child relations with linear Gaussian conditional densities:

- For each root $s \in V$

$$p_s(x_s) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left\{ -\frac{x_s^2}{2\sigma_s^2} \right\} \tag{3.31}$$

- For each directed edge $(u, s) \in E$

$$p_s(x_s|x_u) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left\{ -\frac{(x_s - w_s x_u)^2}{2\sigma_s^2} \right\} \tag{3.32}$$

The model parameters are

$$\theta = \begin{pmatrix} (w_s)_{s\in V: pa(s)\neq\emptyset} \\ (\sigma_s^2)_{s\in V} \end{pmatrix} \tag{3.33}$$

subject to constraints $w_s^2 + w_t^2 = 1$ and $\sigma_s^2 = \sigma_t^2$ for siblings $\{s, t\}$.

The constraints for sibling variables, imposed on the weights w and variances σ^2, are required for parametric identifiability, which we analyze later in detail. Note that, for generality one could additionally consider parent independent mean parameters for each observable variable. Here, we don't include such additional means, simply

for ease of notation and consistency with hidden variables, which are modeled to have zero marginal means. Thus, we will consider centered data when learning from NLVM-Gauss.

Briefly, models in $\mathcal{M}^{\text{Gauss}}$ cast each individual variable (observed or latent) as a linear function of its parent, with some additive Gaussian noise, whose variance is kept equal for siblings. In fact, on the simple tree $a \leftarrow c \rightarrow b$, the corresponding model boils down to factor analysis or probabilistic principal component analysis (PPCA) [95, 96] applied to the pair (X_a, X_b), where the hidden parent X_c plays the role of the "factor" or "principal component". Thus, given an arbitrary structure $G \in \mathcal{F}$, the corresponding model $\mathcal{M}_G^{\text{Gauss}}$ can be considered as a hierarchically arranged collection of PPC analyzers, where each non-terminal variable is a one dimensional descriptor of its two children.

Clearly, the entire forest on X_V and thereby its observable leaves X_O will have a multivariate Gaussian joint density. Though variables are continuous, inference is relatively straightforward with such Gaussian graphical models and can be achieved by an extension of the belief propagation algorithm, the so-called Gaussian Belief Propagation (GaBP), which is introduced and analyzed by Weiss and Freeman [97].

Gaussian graphical models have been heavily studied for discovering interactions among continuous variables, and especially in the "small N, large D" context. The usual approach involves no latent variables, in which case the objective is to learn a sparse dependency structure. Basically this corresponds to covariance estimation:

The large off-diagonal entries from the inverse of the estimated covariance matrix are indicators of edges between the corresponding nodes, whereas values small in magnitude will reveal the conditional independence relationships. In this line of work [98–101], several estimators are proposed within a general model selection formalism. In our case however, the model family is restricted to forests of binary trees, and the involvement of latent variables drastically changes the learning approach.

Again related to our setting, latent variable models on Gaussian observations have been extensively employed for mixture representations and their hierarchical generalizations [102, 103]. However, these involve discrete hidden variables for encoding the mixture components. Other probabilistic approaches that combine Gaussianity with latent variables include examples like dynamical Gaussian process-latent variable models (GP-LVM) [104–106], which are remotely connected to our construction. Basically, these are aimed at finding low dimensional representations for data in a way similar to kernel PCA methods, and use approximate inference due to assumed nonlinearities.

We will combine the relevant notions in those related approaches in an organized way and specific to our latent variable forest structures of binary trees. Next, we explicitly demonstrate nesting, identifiability, inference and EM parameter estimation for NLVM-Gauss.

3.5.1 Nesting in NLVM-Gauss

For consistency, we demonstrate here the nesting property in NLVM-Gauss, as well. Again, given the parameters θ under the model $\mathcal{M}_G^{\text{Guass}}$ with dependency structure $G \in \mathcal{F}$, we formulate a $\widetilde{\theta}$ under the refined model $\mathcal{M}_{\widetilde{G}}^{\text{Gauss}}$ structured with $\widetilde{G} \in \mathcal{R}_G$, such that distributions $\pi(\cdot|\theta) \in \mathcal{M}_G^{\text{Gauss}}$ and $\pi(\cdot|\widetilde{\theta}) \in \mathcal{M}_{\widetilde{G}}^{\text{Gauss}}$ over X_O become identical. Again, this $\widetilde{\theta}$ will serve as the initial parameters for EM applied to the refined model $\mathcal{M}_{\widetilde{G}}^{\text{Bern}}$, when we switch from the coarse one $\mathcal{M}_G^{\text{Bern}}$.

Suppose a and b are two distinct roots in G that are joined in $\widetilde{G}$ as children of a new node c (see Figure 3.8). Then, $\widetilde{\theta}$ can be obtained after replacing variances σ_a^2 and σ_b^2 of θ, which we can assume to satisfy $\sigma_a^2 \geq \sigma_b^2$ without loss of generality, by $\widetilde{w}_a = 1$, $\widetilde{w}_b = 0$, $\widetilde{\sigma}_c^2 = \sigma_a^2 - \sigma_b^2$ and $\widetilde{\sigma}_a^2 = \widetilde{\sigma}_b^2 = \sigma_b^2$.

3.5.2 Identifiability in NLVM-Gauss

Again, we will use induction to establish identifiability for models in NLVM-Gauss. But first we state the following two lemmas.

Lemma 3.5.1. *Let $\{X_a, X_b, X_c, X_d, X_e, X_f\}$ be a collection of zero mean Gaussian random variables, which are Markov with respect to the dependency structure S as shown in Figure 3.7 left. Suppose X_a and X_c are correlated with X_b and X_d, respectively. Then, the correlation between X_e and X_f is given by*

$$\text{corr}(X_e, X_f) = \sqrt{\frac{E[X_a X_d]E[X_b X_c]}{E[X_a X_b]E[X_c X_d]}} \tag{3.34}$$

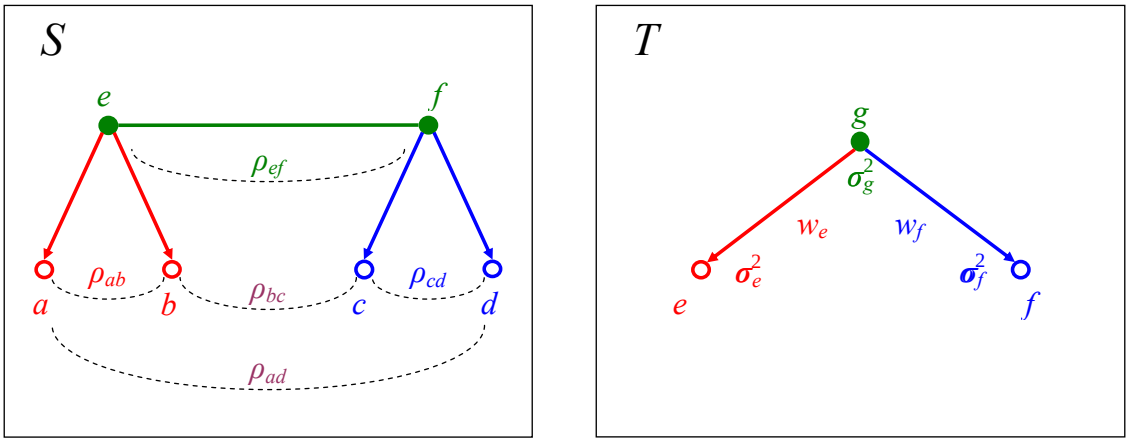

Figure 3.7: Left: Markov structure S analyzed in Lemma 3.5.1 satisfying the relation $\rho_{ef} = \sqrt{\frac{\rho_{ad}\rho_{bc}}{\rho_{ab}\rho_{cd}}}$ among correlation coefficients $\rho_{ef} = \text{corr}(X_e, X_f)$, $\rho_{ad} = \text{corr}(X_a, X_d)$, $\rho_{bc} = \text{corr}(X_b, X_c)$, $\rho_{ab} = \text{corr}(X_a, X_b)$ and $\rho_{cd} = \text{corr}(X_c, X_d)$. Right: Tree structure $T \in \mathcal{F}$ analyzed in Lemma 3.5.2 for NLVM-Gauss with identifiable parameters $\theta = \big(\sigma_e^2, \sigma_f^2, \sigma_g^2, w_e, w_f\big)$.

Proof. The result follows from the product rule for trees. Since the collection of variables is a zero-mean multivariate Gaussian, we can write $E[X_a|X_e] = w_a X_e$, $E[X_b|X_e] = w_b X_e$, $E[X_c|X_f] = w_c X_f$ and $E[X_d|X_f] = w_d X_f$ for some linearity coefficients w_a, w_b, w_c and w_d. Then, by law of total expectation and the Markov property, we can write

$$E[X_a X_d] = E\big[E[X_a X_d|X_e, X_f]\big] = E\big[E[X_a|X_e]E[X_d|X_f]\big] = w_a w_d E[X_e X_f]$$

and

$$E[X_a X_b] = E\big[E[X_a X_b|X_e]\big] = E\big[E[X_a|X_e]E[X_b|X_e]\big] = w_a w_b E[X_e^2].$$

Similarly, wa have $E[X_b X_c] = w_b w_c E[X_e X_f]$ and $E[X_c X_d] = w_c w_d E[X_f^2]$. Since X_a is correlated with X_b, and so is X_c with X_d, the covariances $E[X_a X_b]$ and $E[X_c X_d]$,

and thereby coefficients w_a, w_b, w_c and w_d are all nonzero. Then,

$$\begin{aligned}\operatorname{corr}(X_e, X_f) &= \sqrt{\frac{E[X_e X_f]^2}{E[X_e^2]E[X_f^2]}} \\ &= \sqrt{\frac{w_a w_d E[X_e X_f] w_b w_c E[X_e X_f]}{w_a w_b E[X_e^2] w_c w_d E[X_f^2]}} \\ &= \sqrt{\frac{E[X_a X_d]E[X_b X_c]}{E[X_a X_b]E[X_c X_d]}}\end{aligned}$$

completing the proof. Note that, by a similar analysis we can also write

$$\operatorname{corr}(X_e, X_f) = \sqrt{\frac{E[X_a X_f]E[X_b X_f]}{E[X_a X_b]E[X_f^2]}} = \sqrt{\frac{E[X_c X_e]E[X_d X_e]}{E[X_c X_d]E[X_e^2]}} \tag{3.35}$$

which will be useful for showing identifiability in NLVM-Gauss in weakly balanced trees. □

Lemma 3.5.2. *Let X_e and X_f be two Gaussian and observable random variables that are correlated but conditionally independent given a latent variable X_g, with the corresponding tree structure $T = (e \leftarrow g \rightarrow f) \in \mathcal{F}$ as shown in Figure 3.7 right. Let X_g have the density $p_g = \mathcal{N}(0, \sigma_g^2)$ as in (3.31), and given $X_g = x_g$, let X_e and X_f have the respective conditional densities $p_e(\cdot|x_g) = \mathcal{N}(w_e x_g, \sigma_e^2)$ and $p_f(\cdot|x_g) = \mathcal{N}(w_f x_g, \sigma_f^2)$ as in (3.32). Then, from the joint density*

$$\pi(x_e, x_f|\theta) = \int_{\mathbb{R}} p_e(x_e|x_g) p_f(x_f|x_g) p_g(x_g) dx_g \tag{3.36}$$

over $\{X_e, X_f\}$, the parameters

$$\theta = \left(\sigma_e^2, \sigma_f^2, \sigma_g^2, w_e, w_f\right)$$

given as in (3.33) *with the constraints* $\sigma_e^2 = \sigma_f^2$ *and* $w_e^2 + w_f^2 = 1$*, are identifiable up to simultaneous sign change of* w_e *and* w_f*.*

Proof. Working out the integral in (3.36), we see that $\pi(x_e, x_f|\theta)$ is a centered bivariate Gaussian density in x_e and x_f with the covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_e^2 + \sigma_g^2 w_e^2 & w_e w_f \sigma_g^2 \\ w_e w_f \sigma_g^2 & \sigma_f^2 + \sigma_g^2 w_f^2 \end{pmatrix}.$$

where $\sigma_e^2 = \sigma_f^2$, as constrained. Since means and covariances are identifiable parameters for a multivariate Gaussian density, it suffices to check whether θ is identifiable from Σ.

It is easy to show that, $\alpha = \sigma_e^2 = \sigma_f^2$ and $\beta = \alpha + \sigma_g^2$ are the two eigenvalues of Σ, whereas $(w_a, w_b)^\top$ is the normalized eigenvector corresponding to the large eigenvalue β. Due to nonzero correlation between X_e and X_f, Σ is not spherical (i.e., it is not a multiple of identity matrix), thus, its eigenvectors are unique up to orientation. Therefore, θ can be determined from Σ, up to simultaneous sign change of w_e and w_f, which completes the proof. □

Using the results established in Lemmas 3.5.1 and 3.5.2, we can now state the following proposition on identifiability of parameters for arbitrary models in NLVM-Gauss.

Proposition 3.5.1. *For models in NLVM-Gauss, where, observable or not, each variable is correlated with its sibling, parameters given in* (3.33) *are identifiable, up to sign changes of vectors* (w_s, w_t) *for siblings* $\{s, t\}$*.*

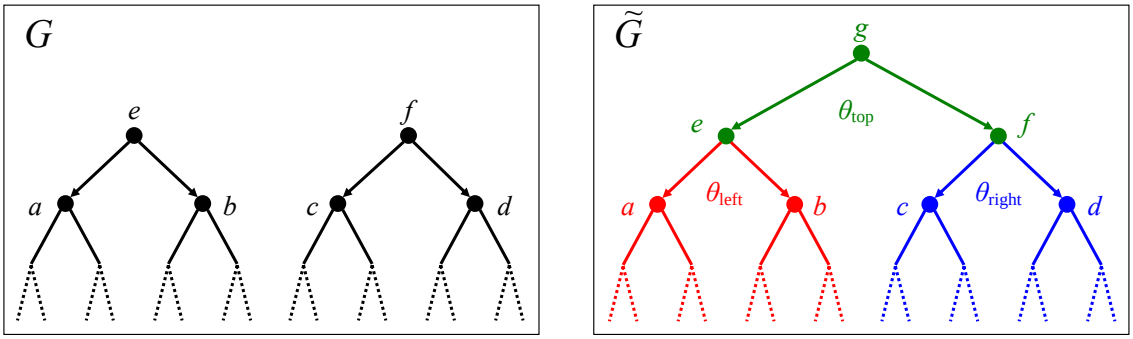

Figure 3.8: Model refinement in NLVM-Gauss: The refined topology $\widetilde{G}$ on the right is derived from G on the left. Red, green and blue colors on $\widetilde{G}$ are to indicate parameters that can be identified separately under $\mathcal{M}_{\widetilde{G}}^{\text{Gauss}}$. θ_{top} are the additional parameters.

Proof. For a given model $\mathcal{M}_G^{\text{Gauss}}$ with forest structure $G \in \mathcal{F}$, the result follows directly by Lemma 3.5.2, if the independent substructures of G are no larger than the topology T considered for the simple tree model in (3.36). Otherwise, suppose the claim holds for $\mathcal{M}_G^{\text{Gauss}}$. Then, similar to our proof for the Bernoulli case, let $\widetilde{G} \in \mathcal{R}_G$ be given with the additional root node g that joins distinct roots e and f of G (see Figure 3.8). Again, as in the Bernoulli case, we can confine our analysis to this new aggregated tree of $\widetilde{G}$, rooted at g with a refined joint distribution $\pi_{td(g)}$ over its terminal variables $X_{td(g)} = (X_{td(e)}, X_{td(f)})$.

Let $\theta = (\theta_{\text{left}}, \theta_{\text{right}}, \theta_{\text{top}})$ be the parameters of $\mathcal{M}_{\widetilde{G}}^{\text{Gauss}}$ specifying $\pi_{td(g)}$, where θ_{left} and θ_{right} correspond to two subtrees below X_e and X_f; whereas θ_{top} are associated with the transitions between X_e, X_f and the new root X_g. We will show the claim sequentially for θ_{left}, θ_{right} and θ_{top}.

The result follows for θ_{left} and θ_{right} directly by induction hypothesis: Given $\pi_{td(g)}$,

the marginal density $\pi_{td(e)} = \int \pi_{td(g)} dx_{td(f)}$ is readily obtainable over variables $X_{td(e)}$, which are terminal to the left subtree below g. The decomposition of $\pi_{td(e)}$ under the refined model $\mathcal{M}_{\widetilde{G}}^{\text{Gauss}}$ is given by $\int \kappa_e(x_{td(e)}|x_e)P(x_e)dx_e$, which is also a representation under the smaller model $\mathcal{M}_G^{\text{Gauss}}$. Thus, the identifiability claim readily holds for parameters θ_{left}, which specify $\kappa_e(x_{td(e)}|x_e)$, as well as for the marginal density of X_e. In the same way, we can obtain the result for θ_{right} and the marginal density of X_f, this time considering $\pi_{td(f)} = \int \pi_{td(g)} dx_{td(e)}$ over terminal variables $X_{td(f)}$ of the right subtree below g.

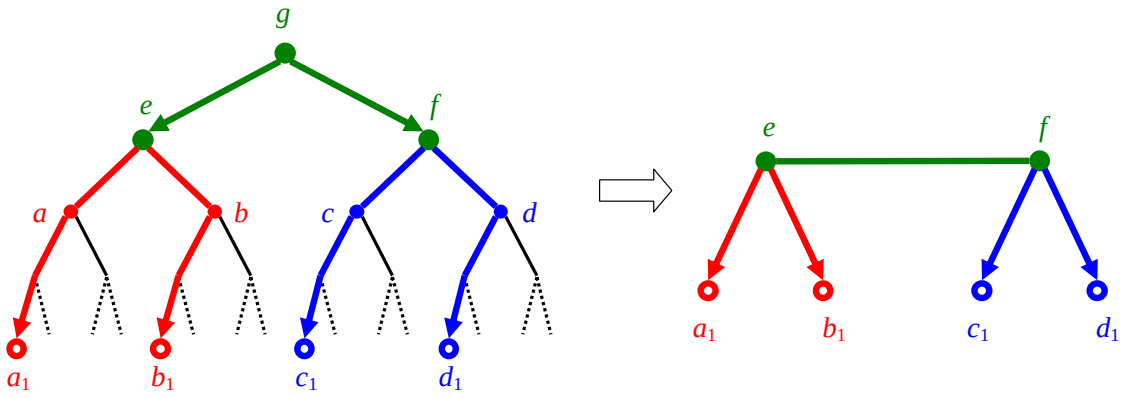

Figure 3.9: Reduction of the Markov structure among designated nodes in $\widetilde{G}$ to the tree of Lemma 3.5.1 to compute $\text{corr}(X_e, X_f)$.

To establish the result for the remaining parameters θ_{top}, we use Lemmas 3.5.1 and 3.5.2. If both e and f are terminal, there is nothing to prove, since that would correspond to the first statement of the induction. If one of them is terminal, say, f, then e must have terminal children $\{a, b\}$ for the balance criterion to hold, and thus, from $\pi_{td(g)}$ we can compute the variance $E[X_f^2]$ and covariances $E[X_a X_f]$, $E[X_b X_f]$

and $E[X_a X_b]$ for the corresponding observable variables, where $E[X_a X_b]$ is nonzero by the assumption on the correlation between siblings X_a and X_b. Then, Equation (3.35) in the proof of Lemma 3.5.1 delivers $\text{corr}(X_e, X_f)$. Otherwise, suppose both e and f are non-terminal, possibly with non-terminal sets of children $\{a, b\}$ and $\{c, d\}$, respectively. Letting, $a_1 \in td(a)$, $b_1 \in td(b)$, $c_1 \in td(c)$ and $d_1 \in td(d)$, we see that the Markov structure of Lemma 3.5.1 is valid for the dependencies between observable variables $\{X_{a_1}, X_{b_1}, X_{c_1}, X_{d_1}\}$ and latent variables $\{X_e, X_f\}$ (see Figure 3.9). Thus, we can again obtain $\text{corr}(X_e, X_f)$ from covariances $E[X_{a_1} X_{d_1}]$, $E[X_{b_1} X_{c_1}]$, $E[X_{a_1} X_{b_1}]$ and $E[X_{c_1} X_{d_1}]$ computed directly from $\pi_{td(g)}$, where the latter two are nonzero implied by the nonzero correlation assumption for siblings. Since we have already identified the marginal densities of X_e and X_f, knowing $\text{corr}(X_e, X_f)$ suffices to ascertain their joint density $P(x_e, x_f)$. Then, using Lemma 3.5.2, we can identify the remaining parameters θ_{top} up to sign change of the vector (w_e, w_f), which completes the proof.

□

Similar to the Bernoulli case, the parametric identifiability of models in NLVM-Gauss can be easily improved to a one-to-one relation $\theta \leftrightarrow \pi(\cdot|\theta)$ (i.e., parameters θ that give rise to $\pi(\cdot|\theta) \in \mathcal{M}_G^{\text{Gauss}}$ can be made unique), with simple additional constraints on the parameter space. For example, analogous to NLVM-Bern, imposing $w_s \geq 0$ for the first member s of each sibling pair $\{s, t\}$, one would force hidden variables to have a non-negative correlation with their left child, thus removing the ambiguity on their interpretation.

3.5.3 Representing Gaussian Densities

Given a forest $G = (V, E) \in \mathcal{F}$, the corresponding model from the NLVM-Gauss family contains joint distributions, which are induced from multivariate Gaussian densities over the complete network (X_O, X_H) on G. Thus, any probability $P(x_S|x_U)$ for node subsets $S, U \subset V$, can be written as a Gaussian function (i.e., a scaled Gaussian probability density) of each individual argument x_s, $s \in S \cup U$. This makes inference relatively easy with Gaussian graphical models; in particular, computation of marginals can be achieved with the Gaussian belief propagation (GaBP) algorithm introduced in [97]. This algorithm was originally formulated for arbitrary undirected Gaussian topologies and in a general matrix-vector notation, where derivations were not detailed for the most part. In our case, though equivalent to GaBP in spirit, we rather want to give the derivations thoroughly for inference as well as EM parameter estimation, in a way specific to our latent variable binary tree structures. This will also provide coherence with previous sections. Our particular choice below for representing Gaussian functions will conveniently lead to such a formulation.

More precisely, we prefer to express a Gaussian function $g(x) = \frac{C}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$ with mean μ, variance σ^2 and some scaling constant C, as the exponential of a quadratic polynomial of x, for instance, of the form

$$g(x) = \exp\{-[A + x\dot{A} + x^2\ddot{A}]\}, \tag{3.37}$$

where for ease of distinction, coefficients A, $\dot{A}$ and $\ddot{A} > 0$ are labeled with dots, which

we put as many as the corresponding power of x.

Since $\mu = -\frac{\dot{A}}{2\ddot{A}}$, $\sigma^2 = \frac{1}{2\ddot{A}}$ and $C = \sqrt{\frac{\pi}{\ddot{A}}} \exp\left\{\frac{\dot{A}^2}{4\ddot{A}} - A\right\}$, we can express the following integrals as functions of the coefficients A, $\dot{A}$ and $\ddot{A}$, too:

$$\int_{\mathbb{R}} g(x)dx = C = \sqrt{\frac{\pi}{\ddot{A}}} \exp\left\{\frac{\dot{A}^2}{4\ddot{A}} - A\right\}, \tag{3.38}$$

$$\int_{\mathbb{R}} xg(x)dx = C\mu = -\frac{\dot{A}}{2\ddot{A}}\sqrt{\frac{\pi}{\ddot{A}}} \exp\left\{\frac{\dot{A}^2}{4\ddot{A}} - A\right\}, \tag{3.39}$$

$$\int_{\mathbb{R}} x^2 g(x)dx = C(\sigma^2 + \mu^2) = \left(\frac{1}{2\ddot{A}} + \frac{\dot{A}^2}{4\ddot{A}^2}\right)\sqrt{\frac{\pi}{\ddot{A}}} \exp\left\{\frac{\dot{A}^2}{4\ddot{A}} - A\right\}. \tag{3.40}$$

Variants of this representation are commonly used for Gaussian functions, they are especially useful when multiple variables are involved. Suppose A, $\dot{A}$ and $\ddot{A}$ are themselves polynomials in another variable y with respective degrees 2, 1 and 0, so that $A - \frac{\dot{A}^2}{4\ddot{A}}$ becomes quadratic in y with positive leading coefficient. Then, similar to g itself, the first integral in (3.38) becomes $\exp\{-[B + y\dot{B} + y^2\ddot{B}]\}$ with coefficients B, $\dot{B}$ and $\ddot{B} > 0$, yielding again a Gaussian function of y. Thus, working out right hand sides above, one can express all three integrals as linear combinations of y's zeroth, first and second powers multiplied by Gaussian functions of y, so that further integrating them with respect to y boils down to repeating the same moment evaluation procedure [2]. We will make use of this recursive property when evaluating nested integrals of Gaussian functions over multiple variables.

[2]This corresponds to finding conditional moments from a Gaussian bivariate (X, Y): If covariance σ_{XY} is nonzero, $E[X^k|Y = y]$ is a degree k polynomial in y.

3.5.4 Dynamic Programming Revisited

Recall that for the generic discrete case we introduced the probabilities κ_s and λ_s in Section 3.2.4, which we could compute recursively for each node using dynamic programming. For continuous variables, we can regard them as probability densities, such that replacing sums by integrals Lemma 3.2.1 still holds. In the specific case of NLVM-Gauss, we now make this lemma more explicit with closed form expressions by exploiting properties of Gaussian variables.

It is easy to show that, for each non-terminal $s \in H$, probabilities κ_s and λ_s of Equation (3.3) are also Gaussian functions of x_s. Thus, we can write them as

$$\kappa_s(x_{td(s)}|x_s) = \exp\big\{ - \big[K_s + x_s\dot{K}_s + x_s^2\ddot{K}_s\big]\big\} \tag{3.41}$$

$$\lambda_s(x_{tc(s)}, x_s) = \exp\big\{ - \big[L_s + x_s\dot{L}_s + x_s^2\ddot{L}_s\big]\big\} \tag{3.42}$$

with coefficients $\{K_s, \dot{K}_s, \ddot{K}_s\}$, $\{L_s, \dot{L}_s, \ddot{L}_s\}$, again labeled with dots according to the power of x_s they multiply. In facts, coefficients $\{K_s, \dot{K}_s\}$ and $\{L_s, \dot{L}_s\}$ will be respective functions of terminal variables $x_{td(s)}$ and $x_{tc(s)}$, but for simplicity, we omit this relation in the notation here and below.

Now, the following proposition, which is equivalent to GaBP, states a continuous extension of Lemma 3.2.1 for computing each of the coefficients $\{K_s, \dot{K}_s, \ddot{K}_s\}$ and $\{L_s, \dot{L}_s, \ddot{L}_s\}$.

Lemma 3.5.3. *For each parent to child edge $(p, c) \in E$, let*

$$K_{pc} = \begin{cases} \log\sqrt{2\pi\sigma_c^2} + \frac{x_c^2}{2\sigma_c^2}, & \text{if } ch(c) = \emptyset; \\ K_c + \log\sqrt{1 + 2\sigma_c^2\ddot{K}_c} - \frac{\sigma_c^2\dot{K}_c^2}{2+4\sigma_c^2\ddot{K}_c}, & \text{otherwise.} \end{cases}$$

$$\dot{K}_{pc} = \begin{cases} -\frac{w_c x_c}{\sigma_c^2}, & \text{if } ch(c) = \emptyset; \\ \frac{w_c\dot{K}_c}{1+2\sigma_c^2\ddot{K}_c}, & \text{otherwise.} \end{cases}$$

$$\ddot{K}_{pc} = \begin{cases} \frac{w_c^2}{2\sigma_c^2}, & \text{if } ch(c) = \emptyset; \\ \frac{w_c^2\ddot{K}_c}{1+2\sigma_c^2\ddot{K}_c}, & \text{otherwise.} \end{cases}$$

Then, for each non-terminal $s \in H$ with $ch(s) = \{q, r\}$, $sb(s) = \{t\}$ and $pa(s) = \{u\}$ (unless they are empty), coefficients $\{K_s, \dot{K}_s, \ddot{K}_s\}$, $\{L_s, \dot{L}_s, \ddot{L}_s\}$ satisfy the following recursions

$$K_s = K_{sq} + K_{sr}$$

$$\dot{K}_s = \dot{K}_{sq} + \dot{K}_{sr}$$

$$\ddot{K}_s = \ddot{K}_{sq} + \ddot{K}_{sr}$$

$$L_s = \begin{cases} \log\sqrt{2\pi\sigma_s^2}, & \text{if } pa(s) = \emptyset; \\ L_u + K_{ut} + \log\sqrt{w_s^2 + 2\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} - \frac{\sigma_s^2(\dot{L}_u+\dot{K}_{ut})^2}{2w_s^2+4\sigma_s^2(\ddot{L}_u+\ddot{K}_{ut})}, & \text{otherwise.} \end{cases}$$

$$\dot{L}_s = \begin{cases} 0, & \text{if } pa(s) = \emptyset; \\ \frac{(\dot{L}_u+\dot{K}_{ut})w_s}{w_s^2+2\sigma_s^2(\ddot{L}_u+\ddot{K}_{ut})}, & \text{otherwise.} \end{cases}$$

$$\ddot{L}_s = \begin{cases} \frac{1}{2\sigma_s^2}, & \text{if } pa(s) = \emptyset; \\ \frac{\ddot{L}_u+\ddot{K}_{ut}}{w_s^2+2\sigma_s^2(\ddot{L}_u+\ddot{K}_{ut})}, & \text{otherwise.} \end{cases}$$

Proof. For each parent to child edge $(p, c) \in E$, let

$$I_{pc} = \begin{cases} p_c(x_c|x_p), & \text{if } ch(c) = \emptyset; \\ \int_{\mathbb{R}} \kappa_c(x_{td(c)}|x_c) p(x_c|x_p) dx_c, & \text{otherwise.} \end{cases}$$

In the first case, i.e. if child c is terminal, then I_{pc} becomes

$$\frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{ -\frac{(x_c - w_c x_p)^2}{2\sigma_c^2} \right\} = \exp\left\{ -\left[\log\sqrt{2\pi\sigma_c^2} + \frac{x_c^2}{2\sigma_c^2} - x_p \frac{w_c x_c}{\sigma_c^2} + x_p^2 \frac{w_c^2}{2\sigma_c^2} \right] \right\}$$

Otherwise, I_{pc} is given by

$$\begin{aligned} &\int_{\mathbb{R}} \exp\{-[K_c + x_c \dot{K}_c + x_c^2 \ddot{K}_c]\} \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{ -\frac{(x_c - w_c x_p)^2}{2\sigma_c^2} \right\} dx_c \\ &= \int_{\mathbb{R}} \exp\left\{ -\left[K_c + \frac{w_c^2 x_p^2}{2\sigma_c^2} + \log\sqrt{2\pi\sigma_c^2} + x_c\left(\dot{K}_c - \frac{w_c x_p}{\sigma_c^2} \right) + x_c^2 \left(\ddot{K}_c + \frac{1}{2\sigma_c^2} \right) \right] \right\} dx_c \\ &= \sqrt{\frac{\pi}{\ddot{K}_c + \frac{1}{2\sigma_c^2}}} \exp\left\{ \frac{\left(\dot{K}_c - \frac{w_c x_p}{\sigma_c^2} \right)^2}{4\left(\ddot{K}_c + \frac{1}{2\sigma_c^2} \right)} - K_c - \frac{w_c^2 x_p^2}{2\sigma_c^2} - \log\sqrt{2\pi\sigma_c^2} \right\} \\ &= \exp\left\{ -\left[K_c + \log\sqrt{1 + 2\sigma_c^2 \ddot{K}_c} - \frac{\sigma_c^2 \dot{K}_c^2}{2 + 4\sigma_c^2 \ddot{K}_c} + x_p \frac{w_c \dot{K}_c}{1 + 2\sigma_c^2 \ddot{K}_c} + x_p^2 \frac{w_c^2 \ddot{K}_c}{1 + 2\sigma_c^2 \ddot{K}_c} \right] \right\}, \end{aligned}$$

where the second equality follows by formula (3.38) and the third one is obtained after collecting x_p terms. Thus, both cases give $I_{pc} = \exp\{-[K_{pc} + x_p \dot{K}_{pc} + x_p^2 \ddot{K}_{pc}]\}$ with $\{K_{pc}, \dot{K}_{pc}, \ddot{K}_{pc}\}$ as defined in the statement.

Then, since $\kappa_s = I_{sq} I_{sr}$ by the κ-recursion of Lemma 3.2.1, we obtain $K_s = K_{sq} + K_{sr}$, $\dot{K}_s = \dot{K}_{sq} + \dot{K}_{sr}$ and $\ddot{K}_s = \ddot{K}_{sq} + \ddot{K}_{sr}$ as claimed.

If s is a root, λ_s becomes $p_s(x_s)$, which is given by

$$\frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left\{ -\frac{x_s^2}{2\sigma_s^2} \right\} = \exp\left\{ -\left[\log\sqrt{2\pi\sigma_s^2} + \frac{x_s^2}{2\sigma_s^2} \right] \right\}$$

which yields $L_s = \log\sqrt{2\pi\sigma_s^2}$, $\dot{L}_s = 0$ and $\ddot{L}_s = \frac{1}{2\sigma_s^2}$ as claimed.

Otherwise, by λ-recursion of Lemma 3.2.1, λ_s becomes $\int \lambda_u p_s I_{ut} dx_u$, ($I_{ut}$ is as given above, i.e., it is either p_t or $\int \kappa_t p_t dx_t$ depending on whether t is terminal), which is then given by

$$\begin{aligned}
&\int_{\mathbb{R}} \exp\{-[L_u + x_u \dot{L}_u + x_u^2 \ddot{L}_u]\} \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left\{ - \frac{(x_s - w_s x_u)^2}{2\sigma_s^2} \right\} \\
&\quad \times \exp\{-[K_{ut} + x_u \dot{K}_{ut} + x_u^2 \ddot{K}_{ut}]\} dx_u \\
&= \int_{\mathbb{R}} \exp\left\{ - \left[L_u + K_{ut} + \log\sqrt{2\pi\sigma_s^2} + \frac{x_s^2}{2\sigma_s^2} + x_u \left(\dot{L}_u + \dot{K}_{ut} - \frac{x_s w_s}{\sigma_s^2} \right) \right.\right. \\
&\qquad \left.\left. + x_u^2 \left(\ddot{L}_u + \ddot{K}_{ut} + \frac{w_s^2}{2\sigma_s^2} \right) \right] \right\} dx_u \\
&= \sqrt{\frac{\pi}{\ddot{L}_u + \ddot{K}_{ut} + \frac{w_s^2}{2\sigma_s^2}}} \exp\left\{ \frac{\left(\dot{L}_u + \dot{K}_{ut} - \frac{x_s w_s}{\sigma_s^2} \right)^2}{4\left(\ddot{L}_u + \ddot{K}_{ut} + \frac{w_s^2}{2\sigma_s^2} \right)} - L_u - K_{ut} - \log\sqrt{2\pi\sigma_s^2} - \frac{x_s^2}{2\sigma_s^2} \right\} \\
&= \exp\left\{ - \left[L_u + K_{ut} + \log\sqrt{w_s^2 + 2\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} - \frac{\sigma_s^2(\dot{L}_u + \dot{K}_{ut})^2}{2w_s^2 + 4\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} \right.\right. \\
&\qquad \left.\left. + x_s \frac{(\dot{L}_u + \dot{K}_{ut}) w_s}{w_s^2 + 2\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} + x_s^2 \frac{\ddot{L}_u + \ddot{K}_{ut}}{w_s^2 + 2\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} \right] \right\},
\end{aligned}$$

where again the second equality follows by formula (3.38) and the third one is obtained after collecting x_s terms. Then, the last expression yields

$$\begin{aligned}
L_s &= L_u + K_{ut} + \log\sqrt{w_s^2 + 2\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} - \frac{\sigma_s^2(\dot{L}_u + \dot{K}_{ut})^2}{2w_s^2 + 4\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} \\
\dot{L}_s &= \frac{(\dot{L}_u + \dot{K}_{ut}) w_s}{w_s^2 + 2\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})} \\
\ddot{L}_s &= \frac{\ddot{L}_u + \ddot{K}_{ut}}{w_s^2 + 2\sigma_s^2(\ddot{L}_u + \ddot{K}_{ut})}
\end{aligned}$$

as claimed. □

With probabilities κ_s and λ_s written explicitly, we can again designate some U as a representative subset of V that contains one vertex per independent tree of G

(e.g., $U = \{s \in V : pa(s) = \emptyset\}$ can be the set of roots), and write the joint density of observable variables X_O in closed form by Equations (3.3) and (3.38):

$$\begin{aligned}
\pi(x_O|\theta) &= \prod_{s\in R} \int_{\mathbb{R}} \kappa_s(x_{td(s)}|x_s)\lambda_s(x_{tc(s)}, x_s)dx_s \\
&= \prod_{s\in R} \int_{\mathbb{R}} \exp\{-[K_s + L_s + x_s(\dot{K}_s + \dot{L}_s) + x_s^2(\ddot{K}_s + \ddot{L}_s)]\}dx_s \\
&= \prod_{s\in R} \sqrt{\frac{\pi}{\ddot{K}_s + \ddot{L}_s}} \exp\left\{\frac{(\dot{K}_s + \dot{L}_s)^2}{4(\ddot{K}_s + \ddot{L}_s)} - K_s - L_s\right\}. \qquad (3.43)
\end{aligned}$$

3.5.5 Posteriors of Hidden variables

Similar to κ_s and λ_s written in closed form above, for each non-terminal $s \in H$, the posterior probability of the corresponding hidden value x_s given the terminal observations x_O, can be explicitly obtained with the network's Markov property and Bayes' rule, as

$$\begin{aligned}
P(x_s|x_O;\theta) &= P(x_s|x_{td(s)}, x_{tc(s)};\theta) \\
&= \frac{P(x_{td(s)}, x_{tc(s)}, x_s|\theta)}{\int_{\mathbb{R}} P(x_{td(s)}, x_{tc(s)}, y_s|\theta)dy_s} \\
&= \frac{\kappa_s(x_{td(s)}|x_s)\lambda_s(x_{tc(s)}, x_s)}{\int_{\mathbb{R}} \kappa_s(x_{td(s)}|y_s)\lambda_s(x_{tc(s)}, y_s)dy_s} \\
&= \frac{\exp\{-[K_s + L_s + x_s(\dot{K}_s + \dot{L}_s) + x_s^2(\ddot{K}_s + \ddot{L}_s)]\}}{\sqrt{\frac{\pi}{\ddot{K}_s+\ddot{L}_s}} \exp\left\{\frac{(\dot{K}_s+\dot{L}_s)^2}{4(\ddot{K}_s+\ddot{L}_s)} - K_s - L_s\right\}}
\end{aligned}$$

This is a Gaussian density in x_s, since it is the product of two Gaussian functions normalized by their integrals. Then using formulae (3.39) and (3.40), we can compute

the first and second posterior moments of the hidden variable X_s by

$$\dot{M}_s = \int_{\mathbb{R}} x_s P(x_s|x_O;\theta)dx_s = -\frac{\dot{K}_s + \dot{L}_s}{2\big(\ddot{K}_s + \ddot{L}_s\big)} \tag{3.44}$$

$$\ddot{M}_s = \int_{\mathbb{R}} x_s^2 P(x_s|x_O;\theta)dx_s = \frac{1}{2\big(\ddot{K}_s + \ddot{L}_s\big)} + \frac{\big(\dot{K}_s + \dot{L}_s\big)^2}{4\big(\ddot{K}_s + \ddot{L}_s\big)^2}, \tag{3.45}$$

which we introduce similarly, with as many dots as the order of the moment. Though we omit them in the notation, the dependence on terminal variables $(x_{td(s)}, x_{tc(s)})$ is obvious through quantities $\dot{K}_s$ and $\dot{L}_s$.

3.5.6 EM for NLVM-Gauss

Using algebraic preliminaries for hidden posteriors from the previous section, we can now give EM's exact update rules. Given i.i.d. training data $\mathbf{x}_O = \{x_s^{(n)}; s \in O, n = 1, ..., N\}$, a fixed forest $G = (V, E) \in \mathcal{F}$ and known parameters θ as in (3.33), a single iteration of EM returns updated parameters $\widehat{\theta}$ for model $\mathcal{M}_G^{\text{Gauss}}$ as follows:

(i) For each root s

$$\widehat{\sigma}_s^2 = \begin{cases} \frac{1}{N}\sum_n \big(x_s^{(n)}\big)^2, & \text{if } s \in O; \\ \frac{1}{N}\sum_n \ddot{M}_s^{(n)}, & \text{otherwise.} \end{cases} \tag{3.46}$$

(ii) For each non-root s with $pa(s) = \{u\}$ and $sb(s) = \{t\}$

$$\widehat{w}_s = \frac{W_{us}}{\sqrt{W_{us}^2 + W_{ut}^2}}$$

$$\widehat{\sigma}_s^2 = \frac{S_{us} + S_{ut}}{2N}$$

where for each parent to child edge $(p, c) \in E$

$$W_{pc} = \begin{cases} \sum_n x_c^{(n)} \dot{M}_p^{(n)}, & \text{if } c \in O; \\ \sum_n \frac{w_c \ddot{M}_p^{(n)} - \sigma_c^2 \dot{K}_c^{(n)} \dot{M}_p^{(n)}}{1 + 2\sigma_c^2 \ddot{K}_c}, & \text{otherwise.} \end{cases} \tag{3.47}$$

$$S_{pc} = \begin{cases} \left(\sum_n \left(x_c^{(n)} \right)^2 + \widehat{w}_c^2 \ddot{M}_p^{(n)} \right) - 2\widehat{w}_c W_{pc}, & \text{if } c \in O; \\ \left(\sum_n \ddot{M}_c^{(n)} + \widehat{w}_c^2 \ddot{M}_p^{(n)} \right) - 2\widehat{w}_c W_{pc}, & \text{otherwise.} \end{cases} \tag{3.48}$$

are evaluated with current parameters θ, using Lemma 3.5.3 and Equations (3.44) and (3.45) (superscript (n) indicates that the corresponding term is a function of the n^{th} observation $x_O^{(n)}$).

3.5.6.1 Derivation of EM for NLVM-Gauss

Given an i.i.d. sample $\mathbf{x}_O$ and current parameters θ, the constrained objective function of EM is given by

$$\begin{aligned} Q(\widetilde{\theta}, \alpha, \beta | \theta) = & \sum_{\substack{s,t \in V \\ pa(s) = pa(t) \neq \emptyset}} \alpha_{st} (\widetilde{w}_s^2 + \widetilde{w}_t^2 - 1) + \beta_{st} (\widetilde{\sigma}_s^2 - \widetilde{\sigma}_t^2) \\ & + \sum_n \int_{\mathbb{R}^{|H|}} P(y_H | x_O^{(n)}; \theta) \log P(x_O^{(n)}, y_H | \widetilde{\theta}) dy_H \end{aligned}$$

with Lagrange multipliers $(\alpha, \beta) = \left(\alpha_{st}, \beta_{st} \right)_{s,t \in V : pa(s) = pa(t) \neq \emptyset}$ for the parametric constraints; and with complete data log-likelihood

$$\log P(x_O, x_H | \widetilde{\theta}) = - \sum_{\substack{s \in V: \\ pa(s) = \emptyset}} \log \sqrt{2\pi \widetilde{\sigma}_s^2} + \frac{x_s^2}{2\widetilde{\sigma}_s^2} - \sum_{(u,s) \in E} \log \sqrt{2\pi \widetilde{\sigma}_s^2} + \frac{(x_s - \widetilde{w}_s x_u)^2}{2\widetilde{\sigma}_s^2}.$$

written as a function of $\widetilde{\theta}$ to be optimized. When taking Q's derivatives below, we will consider the missing data posterior $P(y_H|x_O^{(n)};\theta)$ to be further expanded to $\int_{\mathbb{R}^{|O|}} P(y_O, y_H|x_O^{(n)};\theta)dy_O$ in order to arrive at common expressions for both terminal and non-terminal variables.

(i) Differentiating Q with respect to $\widetilde{\sigma}_s^2$ we get

$$\frac{\partial Q}{\partial \widetilde{\sigma}_s^2} = -\frac{N}{2\widetilde{\sigma}_s^2} + \frac{1}{2\widetilde{\sigma}_s^4}\sum_n \int_{\mathbb{R}} y_s^2 P(y_s|x_O^{(n)};\theta)dy_s$$

which vanishes at

$$\widehat{\sigma}_s^2 = \frac{1}{N}\sum_n \int_{\mathbb{R}} y_s^2 P(y_s|x_O^{(n)})dy_s$$

If s is terminal then $x_s^{(n)}$ is observed and the integral inside the sum reduces to ${x_s^{(n)}}^2$. Otherwise, it is evaluated as $\ddot{M}_s^{(n)}$ by equation (3.45), providing the result.

(ii) Differentiating Q with respect to $\{\widetilde{w}_s, \widetilde{w}_t, \widetilde{\sigma}_s^2, \widetilde{\sigma}_t^2, \alpha_{st}, \beta_{st}\}$ we get

$$\begin{aligned}
\frac{\partial Q}{\partial \widetilde{w}_s} &= 2\alpha_{st}\widetilde{w}_s + \sum_n \int_{\mathbb{R}^2} \frac{y_u(y_s - \widetilde{w}_s y_u)}{\widetilde{\sigma}_s^2} P(y_s, y_u|x_O^{(n)};\theta)dy_s dy_u \\
\frac{\partial Q}{\partial \widetilde{w}_t} &= 2\alpha_{st}\widetilde{w}_t + \sum_n \int_{\mathbb{R}^2} \frac{y_u(y_t - \widetilde{w}_t y_u)}{\widetilde{\sigma}_s^2} P(y_t, y_u|x_O^{(n)};\theta)dy_t dy_u \\
\frac{\partial Q}{\partial \widetilde{\sigma}_s^2} &= \beta_{st} - \frac{N}{2\widetilde{\sigma}_s^2} + \frac{1}{2\widetilde{\sigma}_s^4}\sum_n \int_{\mathbb{R}^2} (y_s - \widetilde{w}_s y_u)^2 P(y_s, y_u|x_O^{(n)};\theta)dy_s dy_u \\
\frac{\partial Q}{\partial \widetilde{\sigma}_t^2} &= -\beta_{st} - \frac{N}{2\widetilde{\sigma}_t^2} + \frac{1}{2\widetilde{\sigma}_t^4}\sum_n \int_{\mathbb{R}^2} (y_t - \widetilde{w}_t y_u)^2 P(y_t, y_u|x_O^{(n)};\theta)dy_t dy_u \\
\frac{\partial Q}{\partial \alpha_{st}} &= \widetilde{w}_s^2 + \widetilde{w}_t^2 - 1 \\
\frac{\partial Q}{\partial \beta_{st}} &= \widetilde{\sigma}_s^2 - \widetilde{\sigma}_t^2
\end{aligned}$$

which together vanish for maximizer parameters

$$\widehat{w}_s = \frac{V_{us}}{\sqrt{V_{us}^2 + V_{ut}^2}}$$
$$\widehat{w}_t = \frac{V_{ut}}{\sqrt{V_{us}^2 + V_{ut}^2}}$$
$$\widehat{\sigma}_s^2 = \widehat{\sigma}_t^2 = \frac{R_{us} + R_{ut}}{2N}$$

where for each parent to child edge $(p, c) \in E$

$$V_{pc} = \sum_n \int_{\mathbb{R}^2} y_c y_p P(y_c, y_p | x_O^{(n)}; \theta) dy_c dy_p$$
$$R_{pc} = \sum_n \int_{\mathbb{R}^2} (y_c - \widehat{w}_c y_p)^2 P(y_c, y_p | x_O^{(n)}) dy_c dy_p$$

To complete the proof, we are left with showing that V_{pc} and R_{pc} respectively equal W_{pc} and S_{pc} as given in the statement.

If child c is terminal, then $x_c^{(n)}$ is observed and the integral inside V_{pc}'s sum reduces to $\int_{\mathbb{R}} x_c^{(n)} y_p P(y_p | x_O^{(n)}; \theta) dy_p$, such that, by Equation (3.44), we obtain

$$V_{pc} = \sum_n x_c^{(n)} \dot{M}_p^{(n)} = W_{pc}.$$

Similarly, for terminal c, the integral inside R_{pc}'s sum can be simplified to $\int_{\mathbb{R}} (x_c^{(n)} - \widehat{w}_c x_p)^2 P(x_p | x_O^{(n)}) dx_p$, such that by equations (3.44) and (3.45), we get

$$R_{pc} = \sum_n \left({x_c^{(n)}}^2 + \widehat{w}_c^2 \ddot{M}_p^{(n)} - 2\widehat{w}_c x_c^{(n)} \dot{M}_p^{(n)} \right) = \left(\sum_n {x_c^{(n)}}^2 + \widehat{w}_c^2 \ddot{M}_p \right) - 2\widehat{w}_c W_{pc} = S_{pc}.$$

On the other hand, if c is non-terminal, we first evaluate

$$\begin{aligned}
&\int_{\mathbb{R}} x_c P(x_c|x_p, x_{td(c)};\theta)dx_c = \frac{\int_{\mathbb{R}} x_c P(x_{td(c)}, x_c|x_p;\theta)dx_c}{\int_{\mathbb{R}} P(x_{td(c)}, y_c|x_p;\theta)dy_c} \\
&= \frac{\int_{\mathbb{R}} x_c \kappa_c(x_{td(c)}|x_c) p_c(x_c|x_p)dx_c}{\int_{\mathbb{R}} \kappa_c(x_{td(c)}|y_c) p_c(y_c|x_p)dy_c} \\
&= \frac{\int_{\mathbb{R}} x_c \exp\{-[K_c + x_c\dot{K}_c + x_c^2\ddot{K}_c]\}\frac{1}{\sqrt{2\pi\sigma_c^2}}\exp\big\{-\frac{(x_c - w_c x_p)^2}{2\sigma_c^2}\big\}dx_c}{\int_{\mathbb{R}} \exp\{-[K_c + y_c\dot{K}_c + y_c^2\ddot{K}_c]\}\frac{1}{\sqrt{2\pi\sigma_c^2}}\exp\left\{-\frac{(y_c - w_c x_p)^2}{2\sigma_c^2}\right\}dy_c} \\
&= \frac{\int_{\mathbb{R}} x_c \exp\big\{-\big[K_c + \frac{w_c^2 x_p^2}{2\sigma_c^2} + \log\sqrt{2\pi\sigma_c^2} + x_c\big(\dot{K}_c - \frac{w_c x_p}{\sigma_c^2}\big) + x_c^2\big(\ddot{K}_c + \frac{1}{2\sigma_c^2}\big)\big]\big\}dx_c}{\int_{\mathbb{R}} \exp\big\{-\big[K_c + \frac{w_c^2 x_p^2}{2\sigma_c^2} + \log\sqrt{2\pi\sigma_c^2} + y_c\big(\dot{K}_c - \frac{w_c x_p}{\sigma_c^2}\big) + y_c^2\big(\ddot{K}_c + \frac{1}{2\sigma_c^2}\big)\big]\big\}dx_c} \\
&= \frac{w_c x_p - \sigma_c^2\dot{K}_c}{1 + 2\sigma_c^2\ddot{K}_c}
\end{aligned}$$

where last equality is obtained by formulae (3.38) and (3.39), and after canceling common terms from numerator and denominator. Then, for non-terminal c, V_{pc} can be written as

$$\begin{aligned}
V_{pc} &= \sum_n \int_{\mathbb{R}} \left(\int_{\mathbb{R}} x_c P(x_c|x_p, x_{td(c)}^{(n)};\theta)dx_c \right) x_p P(x_p|x_O^{(n)};\theta)dx_p \\
&= \sum_n \int_{\mathbb{R}} \frac{w_c x_p - \sigma_c^2\dot{K}_c^{(n)}}{1 + 2\sigma_c^2\ddot{K}_c} x_p P(x_p|x_O^{(n)};\theta)dx_p \\
&= \sum_n \frac{w_c \int_{\mathbb{R}} x_p^2 P(x_p|x_O^{(n)})dx_p - \sigma_c^2\dot{K}_c^{(n)} \int_{\mathbb{R}} x_p P(x_p|x_O^{(n)})dx_p}{1 + 2\sigma_c^2\ddot{K}_c} \\
&= \sum_n \frac{w_c\ddot{M}_p^{(n)} - \sigma_c^2\dot{K}_c^{(n)}\dot{M}_p^{(n)}}{1 + 2\sigma_c^2\ddot{K}_c} \\
&= W_{pc}
\end{aligned}$$

where first equality uses the fact that $P(x_c, x_p|x_O;\theta) = P(x_c|x_p, x_{td(c)};\theta)P(x_p|x_O;\theta)$, second equality uses the previous result and fourth equality follows by Equations (3.44) and (3.45).

Similarly, for non-terminal c, we can write for R_{pc}

$$\begin{aligned}
R_{pc} &= \sum_n \int_{\mathbb{R}^2} (x_c^2 + \widehat{w}_c^2 x_p^2 - 2\widehat{w}_c x_c x_p) P(x_c, x_p | x_O^{(n)}; \theta) dx_c dx_p \\
&= \sum_n \int_{\mathbb{R}} x_c^2 P(x_c | x_O^{(n)}; \theta) dx_c + \widehat{w}_c^2 \int_{\mathbb{R}} x_p^2 P(x_p | x_O^{(n)}; \theta) dx_p \\
&\quad - 2\widehat{w}_c \sum_n \int_{\mathbb{R}} \left(\int_{\mathbb{R}} x_c P(x_c | x_p, x_{td(c)}^{(n)}; \theta) dx_c \right) x_p P(x_p | x_O^{(n)}) dx_p \\
&= \left(\sum_n \ddot{M}_c^{(n)} + \widehat{w}_c^2 \ddot{M}_p \right) - 2\widehat{w}_c W_{pc} \\
&= S_{pc}
\end{aligned}$$

where again third equality follows by Equations (3.44), (3.45) and the previous result for V_{pc}.

Chapter 4

Applications of NLVM

4.1 Experiments with Handwritten Digits

Handwritten digit recognition is an immensely studied topic in OCR applications and pattern classification research. As an OCR problem it poses formidable real world challenges demanding high accuracy and speed, for example, in automatic postal mail sorting based on handwritten zip codes and managing hand-drawn bank checks. Thanks to availability of well known data sets, including a competition sponsored by the National Institute of Standards and Technology (NIST), the task of recognizing handwritten digits has attracted great attention from pattern classification and machine learning communities. With attempts to benchmark the state-of-the-art [21, 107], it has also become a standard application for validating newer methods and comparing their generalization errors.

As far as human performances and humans' ease of learning are concerned, hand written digit recognition still remains an unsolved problem in the machine learning context, except in restricted situations, which require sophisticated preprocessing and massive training with augmented examples to adequately capture possible geometric transformations. The best reported rates seem to be achieved quite recently by San *et al.* with a test error rate as low as 0.35%, which is assessed on the well known MNIST data set using deep neural networks with elastic distortions, where the algorithm is ran on GPUs to speed up the heavy learning phase [108].

Using our proposed NLVM family, we also experimented on the MNIST data, which contains $60,000$ training images and $10,000$ test images, each with 28×28 resolution (see Figure 4.1 for sample images). Our experiments are devised to assess performances in basically two important aspects, namely *classification* and *synthesis*, the latter posing a distinction from the large body of previous work on learning digit-, and in general, object categories.

It is important to mention here that our approach is entirely *generative* unlike the vast majority of competing methods applied on MNIST data. For classification, we compare the likelihoods under the learned probability models, instead of formulating decision surfaces, as in other methods like decision trees, nearest neighbors, SVMs, neural networks etc. In that regard, the latter group of discriminative techniques can make a better use of a massive learning set like MNIST, but in the small sample regime they usually tend to perform poorer than a model-based approach like ours.

Figure 4.1: Some training examples from MNIST handwritten digit database.

Furthermore our methods can be used as a synthesis tool, as well. Thus, we can and did simulate artificial instances from the trained models to visually assess how well we can capture underlying probability distributions.

For our digit classification experiments, we use NLVM-Bern models trained on binary edge features. On the other hand, for synthesis, we consider models from NLVM-Gauss, in which case discretized boundaries of digit shapes are used as observations. The reason why we utilize these two parametric families exclusively for different tasks, and not for both classification and synthesis, is merely due to the nature of their input features.

Binary edge features, namely indicators of image discontinuities, which we extract and arrange according to their quantized orientations, comply with the state space assumptions of NLVM-Bern, and they are known to be good discriminative cues for a recognition task [107]. But those high dimensional feature vectors or their simulations cannot be displayed as a visually plausible image. They rather form a bag of features. On the other hand, once registered to a common framework, discretized shape boundaries, i.e., spatial coordinates of regularly sampled curve points, which can be suitably described with models from NLVM-Gauss, allow a natural visual assessment of learned densities through simulations. As we explain later, the registration process is done by a correspondence matching scheme that employs *class-specific* digit templates, such that in each sample of a given class, the represented points have the same spatial ordering. But then, registered inputs from different classes do not share a common feature space and thus, their comparison and classification based on likelihood ratios is no longer meaningful.

4.1.1 Classifying Handwritten digits

We apply our methods to maximum likelihood classification of MNIST handwritten digits, where for each digit class, we learn models from NLVM-Bern induced over binary image features of local intensity discontinuities.

We first apply a crude slant correction to raw images using spatial moments of the foreground object. Then, we extract oriented edge features sensitive to 8 directions on the image plane, which are quantized with angles $A = \{k \times 45^\circ : k = 0, ..., 7\}$. As in [18], we use photometrically invariant edge detectors. For example, at pixel location (u, v) of a given gray scale image I, we declare a horizontal edge with positive polarity if

$$\begin{aligned} I(u,v) - I(u+1,v) > \max\big\{ & |I(u,v) - I(u-1,v)|, \\ & |I(u,v) - I(u,v-1)|, \\ & |I(u,v) - I(u,v+1)|, \\ & |I(u+1,v) - I(u+2,v)|, \\ & |I(u+1,v) - I(u+1,v-1)|, \\ & |I(u+1,v) - I(u+1,v+1)|\big\} \end{aligned}$$

that is, once the intensity discrepancy in the right hand side is positive and larger than the maximum of six absolute differences found with respect to other nearest neighbors of (u, v) and $(u + 1, v)$. Figure 4.2 illustrates this simple comparison on an example image patch. Comparisons for other edge directions are analogous. For a

better interpretation, edges detected in this way are shown on a synthetic image in Figure 4.3

Again as in [18], we further introduce some geometric invariance, where we allow the edges to "float" with some spatial tolerance δ around their detected positions and perpendicular to their alignment (see Figure 4.2). In particular, we consider one dimensional neighborhoods of length $\delta \in \{1, 3, 5\}$, such that eventually, the feature variable $X_s = X_{s(u,v,\alpha,\delta)}$ becomes the binary indicator of observing an edge along the direction $\alpha \in A$ and inside some perpendicular strip of length δ centered at pixel location (u, v).

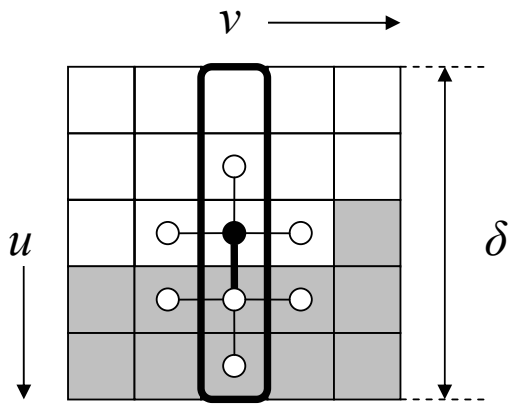

Figure 4.2: The detector for an horizontal edge at pixel location (u, v) (black dot). The detector responds positively, if the difference between pixels connected by the thick segment is greater than the differences between pixels connected by thin segments. The rectangular strip of length δ centered at (u, v) is the tolerance region in which the edge is allowed to "float".

With 8 directional edge types and 3 possible spatial tolerances, resulting feature

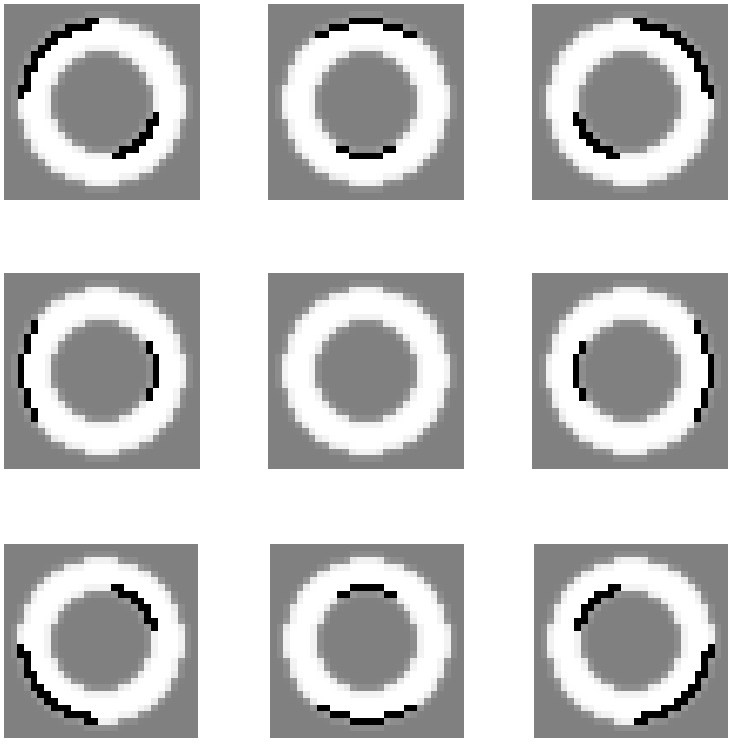

Figure 4.3: Results of directional edge detection. Input gray level image is in the center. Detected edge locations are indicated in black in each of the surrounding images, which are placed with respect to the center, in accordance with the direction of discontinuity of the corresponding edge type.

vectors are $28 \times 28 \times 8 \times 3 = 18,816$ dimensional extracted from each 28×28 image. We reduce this large dimensionality and select for learning the $D = 500$ most relevant features X_O found with the conditional mutual information maximization (CMIM) algorithm [109].

In our experiments with the complete training set from MNIST data, we prefer to partition the ample amount of examples into subcategories, each exhibiting smaller variations. In particular, for each digit class, we form K clusters of similar instances

of X_O using the standard K-means algorithm and estimate individual models from NLVM-Bern, trained separately on those resulting subclasses. We observe that such an attempt improves recognition accuracy thanks to a richer overall representation, which is essentially a mixture of models from the proposed family. We heuristically set $K = 10$, where thanks to biases we incorporate, the process of learning is still robust with the reduced number $N \approx 600$ of data points per cluster.

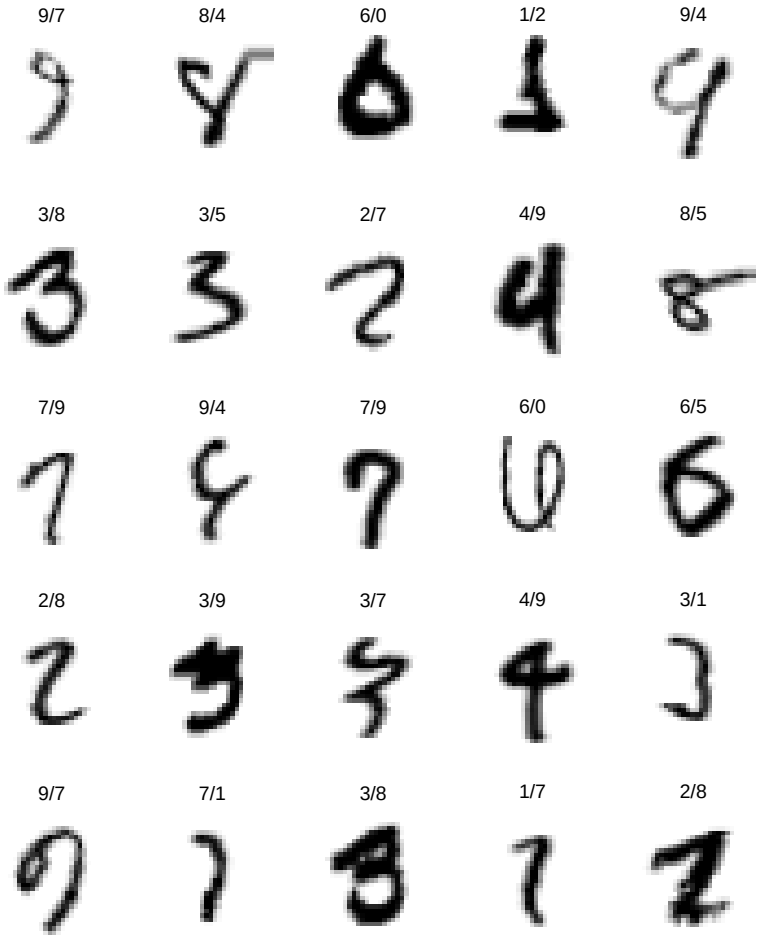

Figure 4.4: Some randomly selected misclassified examples (true class/recognized class).

Eventually, for a test observation x_O, our maximum likelihood classifier is given by

$$f(x_O) = \arg\max_k \{\sum_{l=1}^{K} \beta_l \pi(x_O|k,l)\},$$

where $\pi(\cdot|k,l) \in \mathcal{M}^{\text{Bern}}$ is the estimated joint distribution of X_O that is learned over training data for l^{th} cluster of class k, and β_l is the relative size of that cluster in class k.

Trained on a total of $60,000$ digit examples, our method has achieved 1.25% error rate on the test set, which contains $10,000$ samples. The confusion matrix is given in Table 4.1. Note that our method is not particularly optimized, nor it is equipped with specific prior knowledge concerning the digit world. Except for slant correction it does not involve any preprocessing, or data augmentation with perturbations. Thus, it is reasonable to believe that there is still room for improvement, which can be achieved, for instance, by boosting-like retraining or selecting additional discriminative features for ambiguous examples seen during training. Figure 4.4 shows some randomly selected misclassified examples on the test set, whereas Table 4.2 compares our result with various other approaches (methods are selected uniformly from the range of reported error rates).

To demonstrate the small sample behavior, we also repeated our classification experiments with reduced amounts of training examples. This time, we do not partition data into subclasses, that is learning and classification are done based on a single model from NLVM-Bern per each digit class. In Figure 4.5, we give the error rates

Table 4.1: Confusion matrix of handwritten digit classification on MNIST test set. Each row gives the distribution of samples from the true class among recognized classes indicated in columns.

	0	1	2	3	4	5	6	7	8	9
0	978	0	0	0	0	0	1	1	0	0
1	0	1128	4	0	0	0	0	1	1	1
2	1	0	1018	1	2	0	0	6	4	0
3	0	1	2	991	0	9	0	3	3	1
4	0	0	1	0	970	0	0	0	0	11
5	1	0	0	4	0	882	3	0	0	2
6	4	1	1	0	2	2	947	0	1	0
7	0	2	2	2	2	0	0	1009	1	10
8	1	0	2	2	3	1	0	2	961	2
9	0	0	0	2	10	0	0	6	0	991

Table 4.2: Error rates of various methods on MNIST test data. Numbers are taken from [110].

Method	Test Error Rate (%)	Reference
NLVM-Bern	1.25	*this thesis*
3 nearest neighbor	2.4	Lecun *et al.*, 1998
SVM	1.6	Schölkopf, 1997
Tangent distance	1.1	Simard *et. al*, 1993
Boosted Decision Trees	1.53	Kegl *et al.* 2009
Convolutional Net LeNet4	1.1	LeCun *et al.*, 1998
Convolutional Net LeNet5	0.8	LeCun *et al.*, 1998
6-layer neural network	0.35	Ciresan *et al.*, 2010

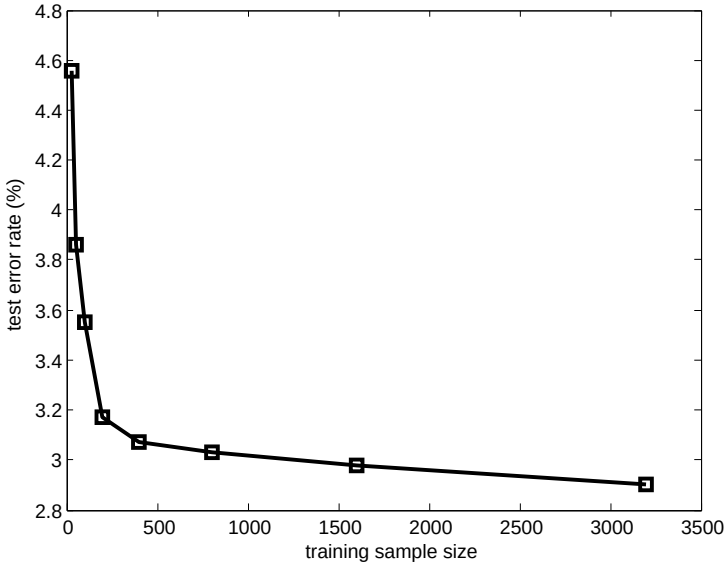

Figure 4.5: Error rates of NLVM-Bern classifier on MNIST test set as a function of training sample size. Learning and classification are done without subclass modeling.

on the test set as a function of training sample size. The smaller performances are attributed to lack of subclass modeling, but the small sample behavior is particularly noteworthy. Even with only 25 learning examples per class, the test error rate is about 4.5%, which is comparable to the k-nearest neighbor classifier, or 2-layer neural networks with 300 hidden units when these are trained on the entire set of circa 6,000 images from each digit class [21].

4.1.2 Synthesizing Handwritten Digits

We further demonstrate the utility of our models in synthesizing artificial instances from their learned density estimates. For this, we train models from NLVM-Gauss

over continuous features, which are extracted from MNIST digit images, as regularly sampled points on shape boundaries. Unlike the annotated edge indicators used for the previous classification experiments, these features can be conveniently visualized from simulations, but in order to be modeled accurately, they first need to be registered to a common framework. This means matching correspondences for points from different samples on a representative template.

4.1.2.1 Registering Digit Shapes

Our approach is based on deformable template matching; it is rather heuristic but very efficient and fast. Given a particular shape class (e.g., digit "5"), we first obtain a coarse template, which is designated as the median intensity iso-contour on the average image. Discretizing this template contour, we obtain uniformly sampled 2D points, which we later match to boundary points found similarly on each training sample. Then, the horizontal and vertical image coordinates of registered curve points will be our observed variables.

Since the number of points we use is on the order of hundreds, we prefer matching them automatically rather than by hand. This is done by finding a transformation between two 2D curves, represented as two point sets, say A and B. We consider A as the collection of points on the template boundary, which is to be mapped onto points B on the target sample. We assume that A and B are already coarsely aligned and that their centroids coincide with each other.

Let $d(\cdot,\cdot)$ denote the Euclidean distance between two points in $\mathbb{R}^2$, and for each point $\mathbf{r}$ in A or B, let $\mathbf{n_r}$ be the unit normal vector directed into the interior of the corresponding shape. Then, our registration procedure can be summarized as follows

- **Correspondence Matching:** For each point $\mathbf{p} \in A$, we find from B a coarse set of candidate matches, given by

$$B_{\mathbf{p}} = \left\{ \arg \min_{\substack{\mathbf{r}\in B \\ \mathbf{n_p}\bullet\mathbf{n_r}\geq 0}} d(\mathbf{p},\mathbf{r}) \right\} \cup \left\{ \mathbf{q} \in B : \mathbf{p} = \arg \min_{\substack{\mathbf{r}\in A \\ \mathbf{n_q}\bullet\mathbf{n_r}\geq 0}} d(\mathbf{q},\mathbf{r}) \right\} \subset B$$

  which contains the nearest B-point to $\mathbf{p}$, together with other B-points, if there are any, for which $\mathbf{p}$ is the nearest A-point. We assume that there are no ties while comparing distances, and impose non-negative dot products of curve normals at $\mathbf{p}$ and at each $\mathbf{q} \in B_{\mathbf{p}}$, so that they lie on the same side of their respective shapes.

  Then, we let $\mathbf{q_p} = \arg\max_{\mathbf{q}\in B_{\mathbf{p}}} d(\mathbf{p},\mathbf{q})$ be the unique coarse match for $\mathbf{p}$ in B, again assuming no ties. The intuition behind designating $\mathbf{q_p}$ as the maximizer of $B_{\mathbf{p}}$ is depicted in Figure 4.6; it is to make the set $\{\mathbf{q_p} : \mathbf{p} \in A\} \subset B$, namely raw correspondences of A as uniform as possible in B, so that we can capture large deformations as well.

- **Deformation:** We first compute the raw displacement vectors $\mathbf{q_p} - \mathbf{p}$ for all $\mathbf{p} \in A$, and then smooth them spatially. To be precise, for each $\mathbf{p} \in A$ we compute a Gaussian weighted average $\mathbf{v_p}$ of raw displacements of nearby A-points. Finally, we do the update $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{v_p}$ for all points in A.

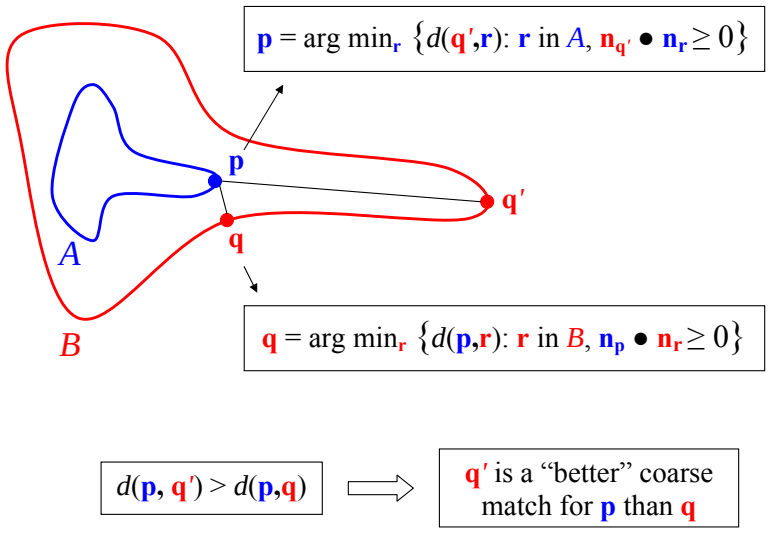

Figure 4.6: Selecting a coarse match for $\mathbf{p} \in A$ from B.

We continue alternating between these two steps until $\max_{\mathbf{p}\in A} \max_{\mathbf{q}\in B_\mathbf{p}} d(\mathbf{p}, \mathbf{q})$ is less than some threshold.

Figure 4.7 shows our registration scheme by iterative deformation of the template shape (blue curve), to an actual target sample (red curve), yielding our features as the displaced boundary points of the deformed shape (black dots). Using 200 regularly sampled curve points, our observed variables X_O become 400 dimensional, where each X_s, $s \in O$ corresponds to one coordinate (horizontal or vertical).

4.1.2.2 Simulations

We train models from NLVM-Gauss on these features using only 100 randomly selected images per digit class from MNIST database. Figure 4.8 shows the model

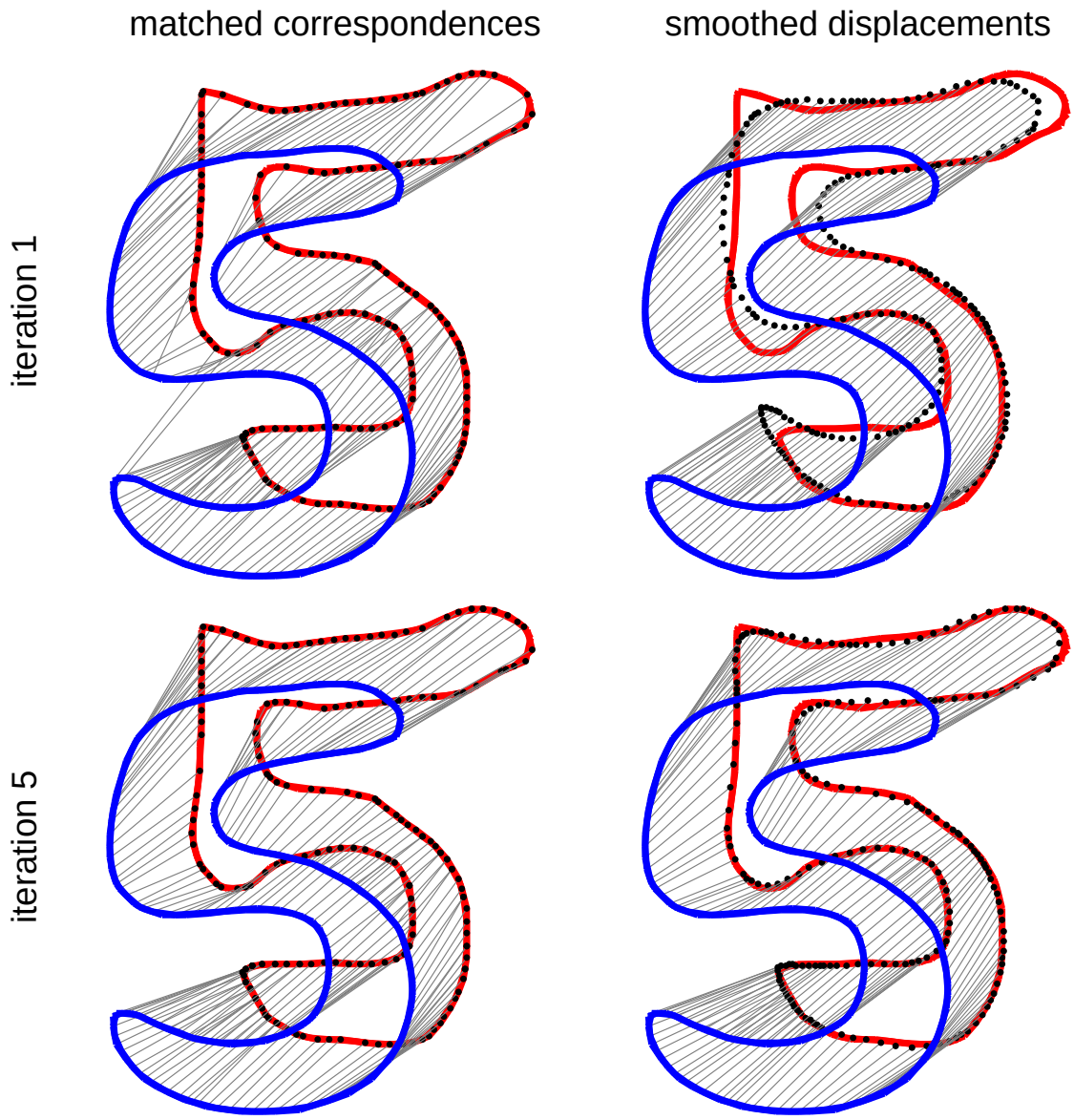

Figure 4.7: Deformable template registration for a sample "5" shape. Blue curves are the template shapes, whereas red ones are target samples, which are identical in all four plots (centroids are vis-a-vis shifted for better visualization). Black dots are matched/deformed curve points with correspondence/displacement vectors shown in gray lines. Left column: Matched correspondences. Right column: Deformations by smoothed displacements. Top row: First iteration. Bottom row: Fifth iteration.

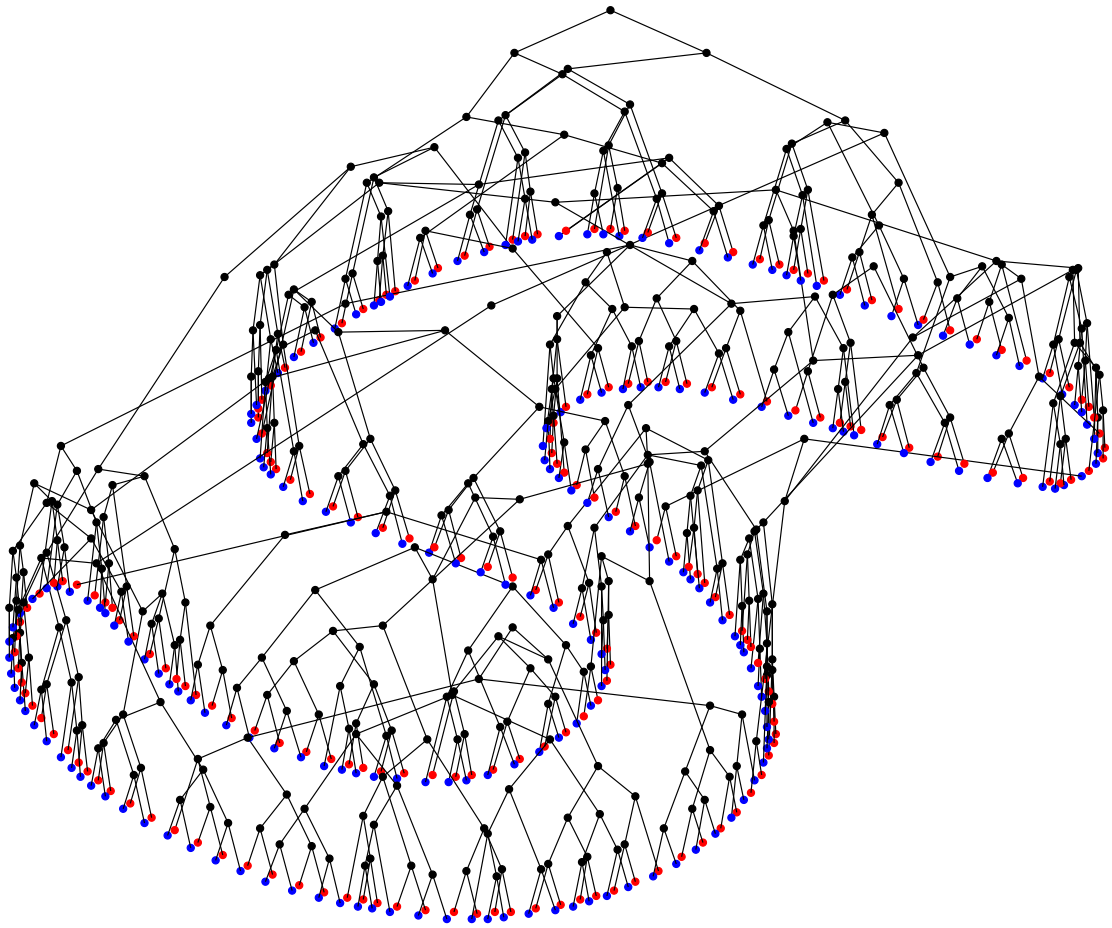

Figure 4.8: Forest structure learned from NLVM-Gauss for shape class "5". Colored dots stand for terminal observable variables, i.e. horizontal (red) and vertical (blue) coordinates of boundary points, regulated by the hierarchy of non-terminal hidden variables shown in black dots.

structure learned for class "5". To assess how well we can estimate shape densities, we generate random instances, as shown in Figure 4.9, by top-down Monte Carlo simulating each learned model.

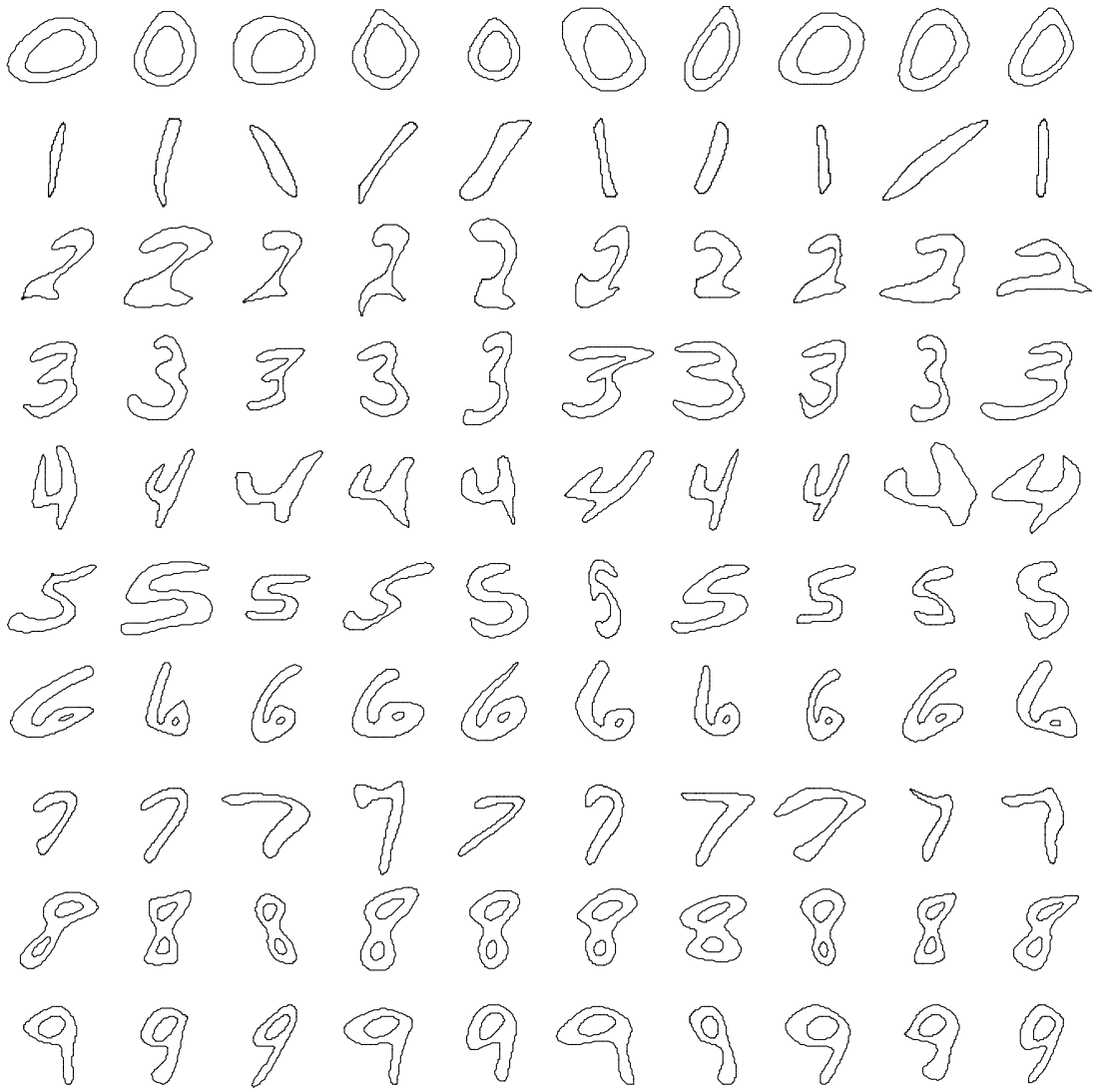

Figure 4.9: Some artificial samples generated from learned shape densities. Simulation is done with top-down Monte Carlo sampling the models, after training each of them on 100 actual data instances. Curves are smoothed after simulation for removing small random artifacts.

4.1.2.3 Captured Variations in the Digit Space

To better interpret learned dependencies, we examine the expected shapes for ranging values of corresponding latent root variables. Given a hidden root $X_u = x_u$

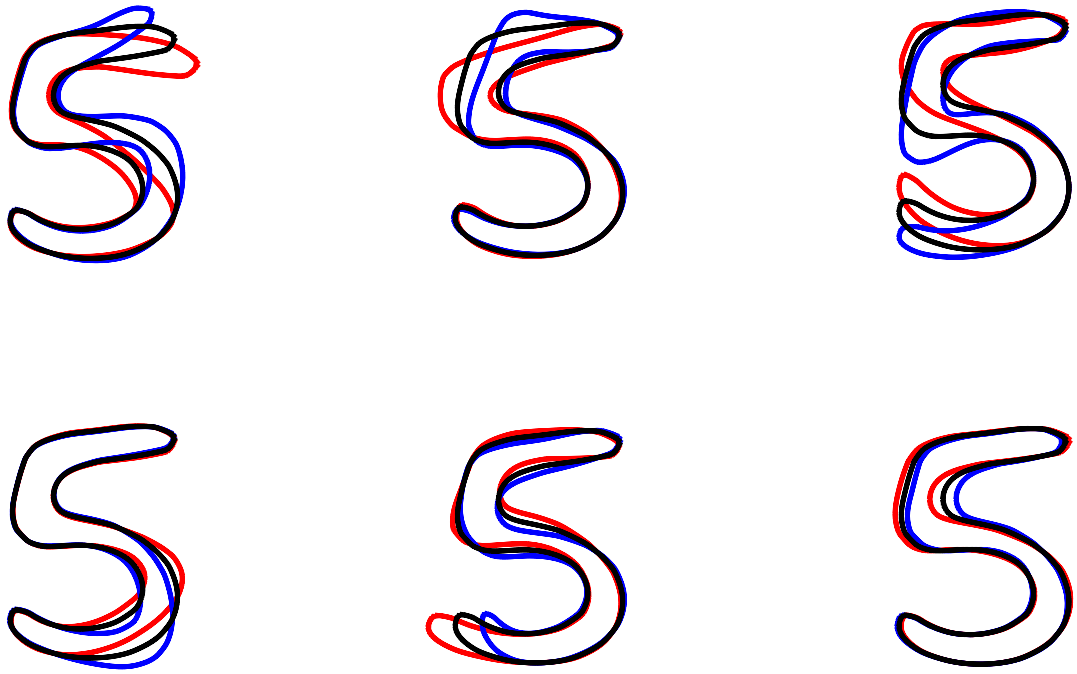

Figure 4.10: Captured variations within the digit class "5": Expected shapes conditional to ranging values of hidden root variables that regulate the six largest tree components of the model learned from NLVM-Gauss. Training is done over 100 actual samples. Each of the six plots corresponds to variations under a distinct tree component, i.e., a distinct hidden root, which hierarchically encodes a different dependency "mode" of the learned density. Black curves are the mean shape (identical in each sub-figure), superposed with blue and red curves, respectively evaluated with $+3\sigma_u$ and $-3\sigma_u$ deviates of the corresponding hidden root variable X_u. Since each such X_u acts locally rather than globally, curves are further smoothed to crease out induced discontinuities between values $E[X_s|X_u = \pm 3\sigma_u]$ and $E[X_t|X_u = \pm 3\sigma_u] = E[X_t]$ at spatially adjacent landmarks $s \in td(u)$ and $t \notin td(u)$.

and learned weights as in (3.33), the expected value of a terminal descendant X_s, $s \in td(u)$ is easily found by linear Gaussian assumption, as

$$E[X_s|X_u = x_u] = x_u \prod_{\substack{t \in an(s) \\ t \neq u}} w_t \tag{4.1}$$

namely, by multiplying with x_u, the weight parameters along the unique lineage from u to s, which contains ancestors $an(s)$ of s.

Playing with values x_u, we can achieve a visual interpretation of discovered dependencies amongst terminal descendants $X_{td(u)}$. Figure 4.10 shows expected shapes, which are found in this way from the learned MLVM-Gauss model for digit class "5". Sub-figures respectively correspond to six largest isolated tree components of the learned forest. Each of the six plots illustrates the variation of observable features conditional to a distinct root variable X_u. Superposed with the mean shape shown in black, blue and red curves respectively depict $E[X_O|X_u = 3\sigma_s]$ and $E[X_O|X_u = -3\sigma_s]$ evaluated by (4.3) for $\pm 3\sigma_u$ deviates of X_u (Note that X_u is constrained to have zero mean).

4.1.2.4 Low-Dimensional Reconstruction

We can further exploit the linear Gaussian interactions in NLVM-Gauss to demonstrate the coarse-to-fine nature of proposed hierarchical representation. Given a realization of observed features, we can infer the most likely configuration of hidden variables, evaluated efficiently as posterior means from (3.44). Then, given these predicted hidden values, which have a reduced dimensionality, we can compute the conditional expectation of observations as in (4.3), yielding their "reconstruction". Repeating this for inferred latent regulators of different hierarchical ranks, we can compare reconstructions to the original observed sample, and give a further visual

assessment on the scale and order of learned dependencies.

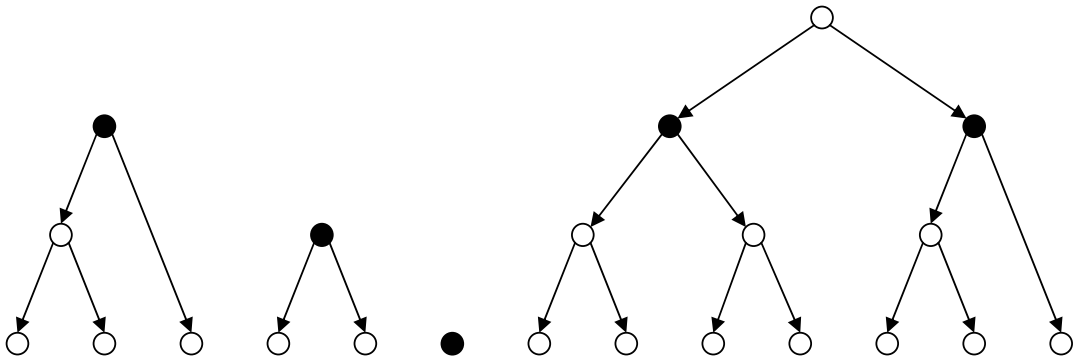

Figure 4.11: An example forest $G = (V, E) \in \mathcal{F}$, where $V_r \subset V$ for $r = 2$ is shown as the set of black nodes.

In particular, given forest $G = (V, E) \in \mathcal{F}$ and an observed realization x_O at terminal nodes, let

$$\widehat{x}_s(x_O) = E[X_s|x_O] = \begin{cases} x_s, & \text{if } s \in O; \\ \dot{M}_s(x_O), & \text{otherwise.} \end{cases} \tag{4.2}$$

denote the value of variable X_s inferred from observations. Also, let

$$V_r = \{s \in V : rk(s) = r\} \cup \{s \in V : rk(s) < r, pa(s) = \emptyset\}$$

be the disjoint union of nodes of rank r, and roots with ranks that are less than r. See Figure 4.11, where V_r is shown on an example forest. The second set in the union above is to incorporate relatively smaller trees in G, so that terminal descendants of V_r entirely cover the set O of leaves. Each terminal $s \in O$ has a unique ancestor in V_r, which we denote by $an_r(s)$. If $r = 0$, then X_{V_r} is simply the set of observed variables

X_O, which are all zero ranked. With $r > 0$, the collection X_{V_r} has fewer members than X_O and contains hidden variables that encode dependencies of order r (or less, if their trees are too small to contain any rank r node) as well as observed variables that are independent. Thus, one can interpret $\widehat{x}_{V_r}(x_O)$ as x_O's "projection to some reduced space", which is found by hierarchical discovery of dependencies.

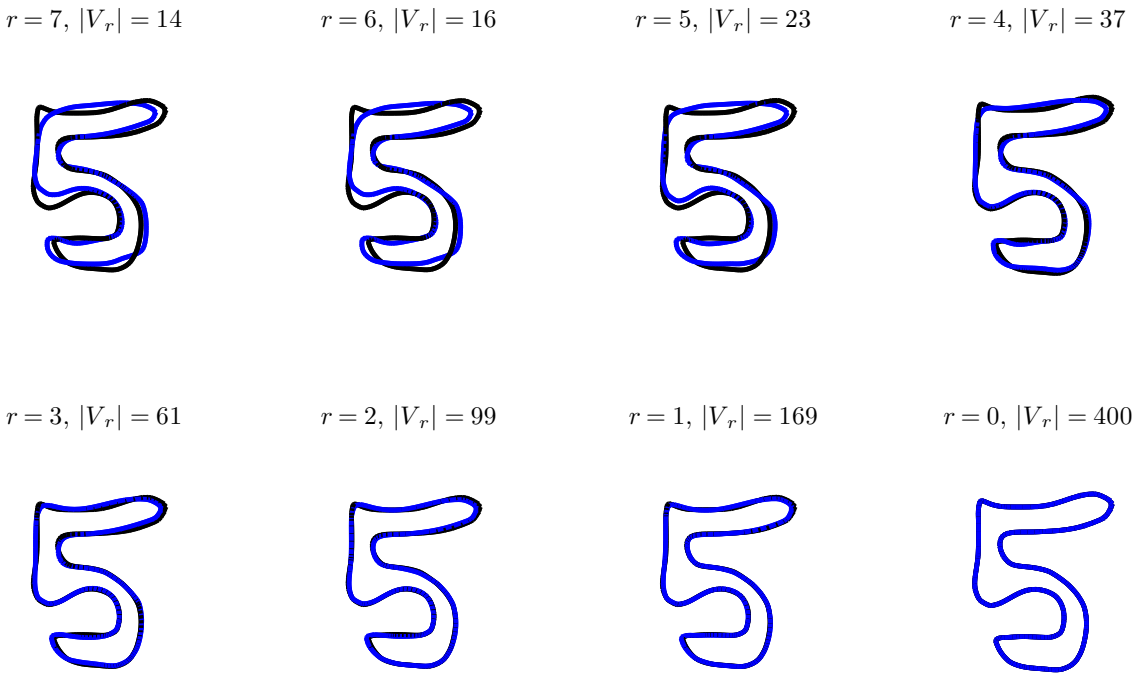

Figure 4.12: Rank r reconstructions of a 400 dimensional actual sample from class "5". Black and blue curves are the respective actual and reconstructed shapes, superposed for comparison. From top-left to bottom-right, rank r decreases from 7 to 0. $|V_r|$ is the dimension of representation, i.e., the number of components used for reconstruction.

Finally, as in (4.3), we can evaluate

$$\widehat{\widehat{x}}_s(x_O, r) = E[X_s|\widehat{x}_{V_r}(x_O)] = \widehat{x}_{an_r(s)} \prod_{\substack{t \in an(s) \\ t \neq an_r(s)}} w_t \tag{4.3}$$

from quantities in (4.2) and weight parameters learned along the unique lineage from V_r to s. We call $\widehat{\widehat{x}}_O(x_O, r)$ "the rank r reconstruction" of x_O. Figure 4.12 shows an example how a "5" shape is reconstructed with various r using $|V_r| < D = |O|$ variables. For example, when $r = 7$, the hidden variables, which regulate up to $2^7 = 128$ observable features, are first deduced with *bottom-up* inference and then used to obtain the reconstruction with *top-down* reasoning. In this way, the original sample can be approximated using fewer dimensions. Note that even fewer dimensional representations would be possible by relaxing the criteria for model selection and balance condition, and allowing fusions of larger sets of dependent variables.

4.2 Experiments with Cancer Profiles

With the application of high-troughput technologies to clinical studies, cancer prediction from microarray data has become an extensively studied statistical problem. Various well known machine learning methods have been reported to accurately detect patterns in gene expression profiles as indicators of a particular cancer or its subtypes. Examples include decision trees (DT), naïve Bayes (NB), k-nearest neighbors (k-NN), support vector machines (SVM), and prediction analysis of microarrays (PAM) based on shrunken centroids [111]. More recently, methods based on relative expression comparisons among few selected genes have also led to viable classifiers like top-scoring pairs (TSP) [27] and its multi-pair extension (k-TSP) proposed in [112], which provides a comprehensive comparative assessment of the previous techniques as well.

To demonstrate the merits of our proposed NLVM family, we apply our method to cancer profile classification, and compare our preliminary results with the aforementioned state-of-the-art. We use ten data sets summarized in 4.3, which all contain two classes, with labels given as either "tumor" (T) versus "normal" (N), or two certain subcategories of the disease, for example, "Acute Lymphocytic Leukemia (ALL)" versus "Acute Myelogenous Leukemia (AML)". Again, observations are microarray expression levels from thousands of genes, whereas sample sizes are typically less than few hundreds.

We evaluate our method's performance with leave one out cross validation (LOOCV).

Table 4.3: Cancer data sets used for evaluating classification performance of the proposed method

Data set	Platform	Number of Genes	Number of Samples	Reference
Colon	cDNA	2000	40(T) vs. 22(N)	[113]
Leukemia	Affy	7129	47(ALL) vs. 25(AML)	[114]
CNS	Affy	7129	25(C) vs. 9(D)	[12]
DLBCL	Affy	7129	58(D) vs. 19(F)	[115]
Lung	Affy	12533	31(M) vs. 150(A)	[116]
Prostate1	Affy	12600	52(T) vs. 50(N)	[117]
Prostate2	Affy	12625	38(T) vs. 50(N)	[118]
Prostate3	Affy	12626	24(T) vs. 9(N)	[119]
GCM	Affy	16063	190(C) vs. 90(N)	[120]
BRCA1	Affy	1658	25(M) vs. 93(N)	[121, 122]

In each validation loop, we first select a small set O of top ten marker genes to be used for both modeling and classification. This is done to alleviate the combinatorial burden of learning, but more importantly to cast our results comparable to interpretable methods like TSP, k-TSP and DT, which typically use 2 to 18 genes for classification. Based on samples seen, we perform this selection with Wilcoxon rank sum test, which is a typical filtering method for identifying differentially expressed genes [123, 124]. Then, we estimate for each of the two sub-populations, a model from NLVM-Gauss trained over the expression levels X_O. Eventually, our maximum likelihood classifier returns the phenotype, for which the corresponding model attains greater likelihood on the expression levels x_O of the left out test sample.

Table 4.4 summarizes our classification results compared to other methods. The accuracy is computed as the sum of true positives and true negatives divided by the number of samples in the corresponding data set. Note that all of the other approaches are discriminative and when there is a gene selection involved (as in TSP, k-TSP, DT, PAM), these methods do it simultaneously while optimizing their decision rules, whereas our approach is generative and filtering is done independent of the learning phase. In that sense, picking the top 10 differentially expressed genes may arguably be less than optimal in our case.

As the results suggest, NLVM-Gauss classifier trained on 10 most differentially expressed genes is among the best performing methods TSP, k-TSP, SVM and PAM. Excluded from Table 4.4, we also experimented with another combined breast cancer

data set [121, 122], containing 'BRCA1 mutation (M)" versus "BRCA1 normal (N)" profiles related to BRCA1 gene. We obtain 83% accuracy, whereas TSP and its triplet extension TST have been reported to perform with respective rates 66% and 77%, [124], where all three classifiers are restricted to operate on the top ten differentially expressed genes.

Figure 4.13 shows further results of the NLVM-Gauss classifier. This time accuracies are evaluated with 10-fold cross validation and given as a function of the number D of the top most differentially expressed genes that are used for modeling. Again, the gene selection is performed with Wilcoxon rank sum test repeatedly inside each validation loop. As can be seen from the plots, there are various trends for different data sets. For leukemia data [114], the rates slightly increase, while for CNS [12], prostate-1 [117] and prostate-2 [119] data sets, there is about 5 to 10% decrease in the accuracy as D is varied from 10 to 320. But note that CNS and prostate-3 data contain very few training samples (34 and 33, respectively), thus, this drop in rates corresponds to only one or two additionally misclassified test examples. For the rest of the experiments performances remain more or less steady, suggesting that cancer detection, or cancer type prediction can be robustly achieved with as few as 10 marker genes, which is also demonstrated with our LOOCV results in Table 4.4.

Table 4.4: LOOCV accuracy (%) of classifiers for binary class expression data sets. Best prediction rate for each data set is highlighted in a frame. Our maximum likelihood classifier NLVM-Gauss(10) is trained over 10 most differentially expressed genes, which are repeatedly selected in each CV loop using Wilcoxon rank sum test.

Method	Leukemia	CNS	DLBCL	Colon	Prostate1	Prostate2	Prostate3	Lung	GCM	Average
NLVM-Gauss(10)	95.83	76.47	90.91	90.32	95.10	81.82	96.97	98.90	82.50	89.87
TSP	93.80	77.90	98.10	91.10	95.10	67.60	97.00	98.30	75.40	88.26
k-TSP	95.83	97.10	97.40	90.30	91.18	75.00	97.00	98.90	85.40	92.01
DT	73.61	67.65	80.52	80.65	87.25	64.77	84.85	96.13	77.86	79.25
NB	100.00	82.35	80.52	58.06	62.75	73.86	90.91	97.79	84.29	81.17
k-NN	84.72	76.47	84.42	74.19	76.47	69.32	87.88	98.34	82.86	81.63
SVM	98.61	82.35	97.40	82.26	91.18	76.14	100.00	99.45	93.21	91.18
PAM	97.22	82.35	85.71	85.48	91.18	79.55	100.00	99.45	79.29	88.91

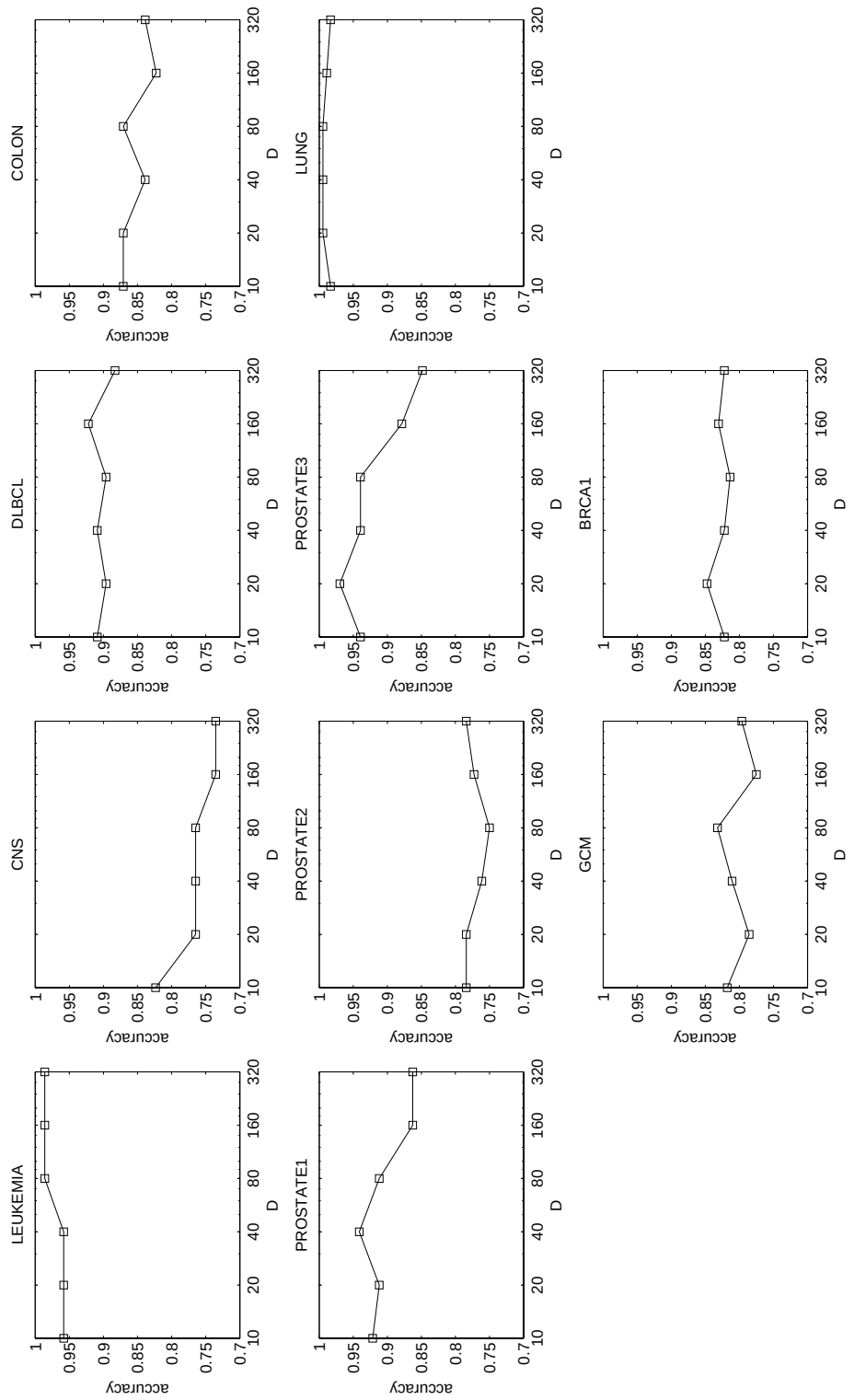

Figure 4.13: 10-fold CV accuracies with NLVM-Gauss as a function of number D of the top most differentially expressed genes used in modeling.

Chapter 5

Discussion and Conclusion

5.1 Model for Cell Signaling

Cell signaling processes play a central role in the etiology of many diseases, and signaling proteins provide a logical target for therapeutic intervention with numerous treatments under development [76]. Although the hopes for targeted therapy are high, the approach is limited to disrupting a single signaling protein. On the other hand, studies in glioblastoma multiforme have demonstrated that each individual tumor has a different set of aberrant signaling proteins [125, 126], making it essential to identify in each patient which proteins need to be targeted for treatment.

The logical method to identify an aberrant signaling protein is to look for changes in protein post-translational modifications, since most signal propagation takes the form of phosphorylation changes in proteins or cleavage events changing protein lo-

calization and structure. However, these measurements are presently very limited *in vivo*. An alternative approach is to use the mature microarray technology targeted at mRNA transcripts, since transcriptional changes resulting from activation or suppression of transcriptional regulators are primary endpoints for many signaling processes. Microarray data coupled with reasonable models of signaling networks provide a potential avenue for identification of individual signaling protein abnormalities.

We laid out a comprehensive statistical model for cell signaling that aims to recover patient dependent protein activities from microarray data of their transcriptional endpoints. Given this task, the prior information was indispensable due to greatly increased difficulties posed by small training samples, measurement noise and large proportion of hidden components to be inferred. Consequently, we considered a documented core signaling diagram (Figure 2.1) that is particular to our breast cancer study and available expression data.

Our model has two important realistic aspects. First, it accounts for biological heterogeneity, that is, cell-to-cell differences within the experimented tissue, and second it has a multi-level approach to elaborate the overall generative process starting from hidden phenotypes to final log-expressions with different statistical constructions in the cell, tissue, population and measurement levels. In particular, we considered a latent variable Bayesian network for each individual cell, taking the core wiring diagram as the underlying directed acyclic graph. Then, we modeled the microarray measurements as noisy logarithms of the total amount of RNA extracted from the

tissue. Relating those to the single cell abundances averaged over a large ensemble of cells, and using the law of large numbers, we were able to represent them as conditional expectations given patient specific phenotypes, where this expectation is taken with respect to the Bayesian network formulation defined for a single cell. Finally, we took the activation rates of signal initiators, namely cell receptors, as random phenotypic variables that are modeled at the population level and give rise to observations through this entire process. Learning from available microarray data can be robustly achieved using the SAEM algorithm, supporting rigorous statistical inference.

The RAS-RAF network we analyzed has 78 components where 40 of them (cell receptors, internal signaling proteins and transcription factors) are hidden and the remaining 38 represent single cell abundances of the transcripts that are indirectly observed as final log-expressions. For computational purposes, we constructed the corresponding cell level Bayesian network via parameter-free, linear and generic transition probabilities as given in Equation (2.2), which are used in a majority of our experiments. It may be argued that these linear transitions oversimplify the underlying chemical processes, but still, the overall model allows the user to incorporate his/her expert knowledge and to explain signaling dynamics with more complex, nonlinear choices. In that regard, we believe there is still room to improve the predictive accuracy of the method. In fact, without sacrificing efficiency, we assumed another linear formulation (2.24), which favors the known dominance of inhibition over activation at crossing pathways. Similarly, we discussed a nonlinear version as laid out

in Table 2.4. Both extensions maintain a limited complexity via parameter sharing and both demonstrated better prediction performances. Further alternatives can be explored as well; for instance, one can differentiate interactions at the signaling level from those at the level of transcription; or enrich the representation by introducing extra parameters that can be validated as more protein data becomes available.

We demonstrated model identifiability, reproducibility through simulations and robustness under biologically meaningful revisions of topology. Using two real patient data sets, one with complete ERα ground truths, the other with complete ground truths for both ERα and EGFR, we validated our method's ability to recover receptor status in a breast cancer study. As signaling plays a central role in the etiology of many diseases, identification of the aberrant proteins driving signaling errors will provide information for personalized therapeutic intervention. It is expected that this will improve patient prognosis and reduce undesirable side-effects during treatment.

As a future work, we wish to broaden the scope of learning for possible internal cell-level parameters as they may be involved in more general formulations for the signal transition function ϕ_v (2.2). One possibility is to alternate between SAEM, which is used for learning population and measurement parameters, with another hill-climbing procedure to estimate shared parameters of the cell level Bayesian network.

Another prospective important goal of our research is to be able to learn more complex topologies by growing them from prior core diagrams, such as the one in Figure 2.1. To do that, we can sequentially propose new non-terminal hidden nodes

at the cell level, append them to the current network and validate them with gains in data likelihood. Incorporating additional genes that were not present in the original core diagram but available in the expression data, we can identify such new hidden nodes by comparing their conjectured cliques against public domain protein-protein interaction databases.

5.2 Model Family for Discovery of Dependencies

For situations where prior knowledge is insufficient to circumscribe the difficulties, we argued for introducing carefully chosen biases to manage susceptibility to high variance, namely to avoid over-fitting scarcely available data, while still being able to recover complex variable interactions that may exist. We entertained this idea in a generative approach by formulating a nested family of hierarchical latent variable graphical models, denoted NLVM, which are forest structured probability distributions with observable variables represented at the terminal nodes.

We explored two particular parametric cases, one for binary variables with local dependencies given as Bernoulli conditionals, and the other formulated for real valued variables, with linear Gaussian interactions. For both choices, we established parametric identifiability, laid out a belief propagation based dynamic programming approach for exact inference, and provided details of the EM algorithm for maximum

likelihood estimation of model parameters.

Exploiting the nesting property, we formulated a structure learning algorithm that sequentially discovers dependencies and fuses corresponding substructures to a joint representation. In particular, this corresponds to a local search within the proposed family, where at each step the appropriate move among candidate merges is selected based on its BIC score. This criterion directly considers the achieved gain in data likelihood at the leaves, but with a penalty term, which not only incorporates the additional constant complexity but also the sample size, so that the algorithm can adapt itself when more data are available.

We entertained our generative models with applications which can assess its validity directly. Using binary edge features annotated with polarity and orientation, we applied our Bernoulli family NLVM-Bern to maximum likelihood classification of handwritten digit images from the extensively studied MNIST data set. Our classification rate is close to the state-of-the-art. It is worth noting that all of the methods reported to perform superiorly to ours, are discriminative and specifically adjusted for to the task at hand, whereas our approach is entirely generative and has a general construction without incorporating particular knowledge about the digit world. In order to demonstrate the generative property, we also did experiments on density estimation of digit shapes, this time using their discretized shape contours as real-valued features represented with our Gaussian family NLVM-Gauss. We generated artificial shapes from the learned densities, and, as can be seen from our results, our

simulations are very plausible even when training is done over a limited number of actual examples. Similarly, we also demonstrated how dependencies are encoded by visualizing variations of the jointly represented components in the feature space.

We applied NLVM-Gauss to maximum likelihood classification as well. We experimented with phenotype prediction, such as detecting "cancer" vs. "normal" or different subtypes of a specific cancer, using microarray gene expression data from corresponding clinical studies. Our LOOCV experiments demonstrated performances comparable to the state-of-the-art, which again involves only discriminative approaches unlike our method.

Motivated by our method's potential in the applications discussed so far, we intend to extend our research to various other directions involving more challenging tasks, both in vision and biology. In vision, for example, we plan to investigate the possibility of performing recognition for a large number of object categories in cluttered scenes. In this context, it seems unreasonable to learn a completely different model for each category, since it is clear that distinct object classes share features or even larger parts (e.g., dogs and cats, cars and bikes), as indicated in [28, 29]. Thus, we wish to train "reusable models" for such shared semantic tokens, probably even from an automatically generated alphabet, and then combine them to attain higher level structures.

We also wish to explore the potential of our generative models in other biological applications such as gene clustering, and recovering gene regulatory networks. For

such tasks, once the latent variable forest structure is estimated, the lengths of the undirected paths between the terminal gene expression variables, parameters learned along them, as well as the hierarchical arrangement of these paths, can be used collectively to estimate the strengths of co-regulation and to formulate hypotheses on causality.

Bibliography

[1] E. R. Dougherty, "Small sample issues of microarray-based classification," *Comparative and Functional Genomics*, no. 2, 2001.

[2] P. Sebastiani and M. Ramoni, "Statistical challenges in functional genomics," *Statistical Science*, vol. 18, pp. 33–70, 2003.

[3] R. Simon, M. D. Radmacher, K. Dobbin, and L. M. McShane, "Pitfalls in the use of dna microarray data for diagnostic and prognostic classification," *Journal of the National Cancer Institute*, vol. 95, no. 1, pp. 14–18, 2003.

[4] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, and R. Spang, "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proceedings of the National Academy of Sciences*, vol. 98, pp. 11 462–11 467, 2001.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 1999.

BIBLIOGRAPHY

[6] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, 2001.

[7] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, pp. 1–58, 1992.

[8] L. Breiman, "Statistical modeling: The two cultures," *Statistical Science*, no. 16, pp. 199–215, 2001.

[9] D. J. Duggan, M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent, "Expression profiling using cdna microarrays," *Nature Genetics*, vol. 21, no. 1, pp. 10–14, 1999.

[10] R. J. Lipshutz, S. P. Fodor, T. R. Gingeras, and D. J. Lockhart, "High density synthetic oligonucleotide arrays," *Nature Genetics*, vol. 21, no. 1, pp. 20–24, 1999.

[11] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend, "Functional discovery via a compendium of expression profiles," *Cell*, vol. 102, no. 1, pp. 109–126, 2000.

[12] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E.

McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, no. 6870, pp. 436–442, 2002.

[13] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3, pp. 601–620, 2000.

[14] I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20, pp. 248–256, 2004.

[15] A. Djebbari and J. Quackenbush, "Seeded bayesian networks: constructing genetic networks from microarray data," *BMC Systems Biology*, vol. 2, pp. 57–69, 2008.

[16] I. Ulitsky and R. Shamir, "Identifying functional modules using expression profiles and confidence-scored protein interactions," *Bioinformatics*, vol. 25, no. 9, pp. 1158–1164, 2009.

[17] K. Sachs, S. Itani, J. Carlisle, G. P. Nolan, D. Pe'er, and D. A. Lauffenburger,

"Learning signaling network structures with sparsely distributed data." *Journal of Computational Biology*, vol. 16, no. 2, pp. 201–212, 2009.

[18] F. Fleuret and D. Geman, "Coarse-to-fine face detection," *International Journal of Computer Vision*, no. 41, 2001.

[19] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[20] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Springer, 1999.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[23] D. M. Gavrila, "Multi-feature hierarchical template matching using distance transforms," in *International Confernece on Pattern Recognition*, 1998, pp. 439–444.

[24] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification," *Nature neuroscience*, vol. 5, no. 7, pp. 682–687, 2002.

[25] F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 524–531.

[26] Y. Amit and A. Trouvé, "POP: Patchwork of parts models for object recognition," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 267–282, 2007.

[27] S. Geman, D. F. Potter, and Z. Chi, "Composition systems," *Quarterly of Applied Mathematics*, vol. 60, no. 4, pp. 707–736, 2002.

[28] S. Krempp, D. Geman, and Y. Amit, "Sequential learning with reusable parts for object detection," in *USENIX Technical Conference*, 2002.

[29] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: Efficient boosting procedures for multiclass object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 762–769.

[30] D. A. Socolinsky, J. D. Neuheisel, C. E. Priebe, J. DeVinney, and D. Marchette, "Fast face detection with a boosted cccd classifier," 2002.

[31] Y. Amit, D. Geman, and X. Fan, "A coarse-to-fine strategy for multiclass shape detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 12, pp. 1606–1621, 2004.

BIBLIOGRAPHY

[32] G. Blanchard and D. Geman, "Hierarchical testing designs for pattern recognition," *The Annals of Statistics*, vol. 33, no. 3, pp. 1155–1202, 2005.

[33] S. Geman, K. Manbeck, and E. Mcclure, "Coarse-to-fine search and rank-sum statistics in object recognition," Tech. Rep., 1995.

[34] E. Bienenstock, S. Geman, and D. Potter, "Compositionality, mdl priors, and object recognition," in *Conference on Neural Information Processing Systems*. MIT Press, 1997, pp. 838–844.

[35] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[36] D. Heckerman, "A tutorial on learning with bayesian networks," in *Learning in Graphical Models*, 1999.

[37] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, pp. 1–305, 2008.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, no. 39, pp. 1–38, 1977.

[39] B. Delyon, M. Lavielle, and E. Moulines, "Convergence of a stochastic approx-

imation version of the EM algorithm," *Annals of Statistics*, vol. 27, no. 1, pp. 94–128, 1999.

[40] D. M. Chickering and D. Heckerman, "Efficient approximations for the marginal likelihood of bayesian networks with hidden variables," *Machine Learning*, vol. 29, no. 2-3, pp. 181–212, 1997.

[41] H. Akaike, "A new look at statistical model identification," *IEEE Transactions on Automatic Control*, no. 19, 1974.

[42] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, no. 6, pp. 461–464, 1978.

[43] N. Friedman, "The bayesian structural EM algorithm," in *Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 129–138.

[44] T. Verma and J. Pearl, "A theory of inferred causation," in *2nd International Coference on the Principles of Knowledge Representation and Reasoning*, 1991.

[45] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young, "Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks," in *Pacific Symposium on Biocomputing*, 2001, pp. 422–433.

[46] ——, "Combining location and expression data for principled discovery of ge-

netic regulatory network models," in *Pacific Symposium on Biocomputing*, 2002, pp. 437–449.

[47] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, no. 303, pp. 799–805, 2004.

[48] D. Pe'er, "Bayesian network analysis of signaling networks: A primer," *Science STKE*, no. 2005, p. 4, 2005.

[49] T. E. Ideker, V. Thorsson, and R. M. Karp, "Discovery of regulatory interactions through perturbation: inference and experimental design," in *Pacific Symposium on Biocomputing*.

[50] D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed expression profiles," *Bioinformatics*, vol. 17, no. 1, pp. 215–224, 2001.

[51] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano, "Combining microarrays and biological knowledge for estimating gene networks via bayesian networks," *Journal of Bioinformatics and Computational Biology*, vol. 2, no. 1, pp. 77–98, 2004.

[52] S. Mukherjee and T. P. Speed, "Network inference using informative priors," *Proceedings of the National Academy of Sciences*, vol. 105, no. 38, pp. 14 313–14 318, 2008.

[53] M. Liu, A. Liberzon, S. W. Kong, W. R. Lai, P. J. Park, I. S. Kohane, and

S. Kasif, "Network-based analysis of affected biological processes in type 2 diabetes models," *PLoS Genetics*, vol. 3, no. 6, p. e96, 2007.

[54] M. R. Birtwistle, M. Hatakeyama, N. Yumoto, B. A. Ogunnaike, J. B. Hoek, and B. N. Kholodenko, "Ligand-dependent responses of the ErbB signaling network: experimental and modeling analyses." *Molecular Systems Biology*, vol. 3, p. 144, 2007.

[55] J. C. Liao, R. Boscolo, Y. L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury, "Network component analysis: reconstruction of regulatory signals in biological systems," *Proceedings of the National Academy of Sciences*, vol. 100, no. 26, pp. 15 522–15 527, 2003.

[56] A. V. Kossenkov and M. F. Ochs, "Matrix factorization for recovery of biological processes from microarray data," *Methods in Enzymology*, vol. 467, pp. 59–77, 2009.

[57] M. Niepel, S. L. Spencer, and P. K. Sorger, "Non-genetic cell-to-cell variability and the consequences for pharmacology." *Current Opinion in Chemical Biology*, vol. 13, no. 5-6, pp. 556–561, 2009.

[58] P. Lazarsfeld and N. W. Henry, *Latent structure analysis*. Houghton Mifflin, 1968.

BIBLIOGRAPHY

[59] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

[60] D. Connolly, "Constructing hidden variables in bayesian networks via conceptual clustering," in *International Conference on Machine Learning*, 1993, pp. 65–72.

[61] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, vol. 2, no. 2, pp. 139–172, 1987.

[62] N. L. Zhang, "Hierarchical latent class models for cluster analysis," *Journal of Machine Learning Research*, vol. 5, pp. 697–723, 2004.

[63] N. Zhang and T. Kocka, "Efficient learning of hierarchical latent class models," in *16th IEEE International Conference on Tools with Artificial Intelligence*, 2004, pp. 585–593.

[64] Y. Wang, N. L. Zhang, and T. Chen, "Latent tree models and approximate inference in bayesian networks," *Journal of Artificial Intelligence Research*, vol. 32, no. 1, pp. 879–900, 2008.

[65] S. Harmeling and C. K. I. Williams, "Greedy learning of binary latent trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1087–1097, 2011.

[66] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[67] M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky, "Learning latent tree graphical models," 2010. [Online]. Available: http://arxiv.org/abs/1009.2722

[68] G. Elidan and N. Friedman, "The information bottleneck EM algorithm," in *Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2003, pp. 200–208.

[69] C. K. I. Williams, "A mcmc approach to hierarchical mixture modelling," in *Conference on Neural Information Processing Systems*. The MIT Press, 1999, pp. 680–686.

[70] R. M. Neal, "Density modeling and clustering using dirichlet diffusion trees," *Bayesian Statistics*, vol. 7, pp. 619–629, 2003.

[71] C. Kemp and J. B. Tenenbaum, "The discovery of structural form," *Proceedings of the National Academy of Sciences*, vol. 105, no. 31, pp. 10 687–10 692, 2008.

[72] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular Biology and Evolution*, vol. 4, no. 4, pp. 406–425, 1987.

[73] J. Felsenstein, *Inferring Phylogenies*, 2nd ed. Sinauer Associates, 2003.

[74] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

[75] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, "Latent hierarchical structural learning for object detection," in *IEEE Computer Vision and Pattern Recognition*, 2010.

[76] P. J. Roberts and C. J. Der, "Targeting the raf-mek-erk mitogen-activated protein kinase cascade for the treatment of cancer," *Oncogene*, vol. 26, no. 22, pp. 3291–3310, 2007.

[77] J. J. Yeh and C. J. Der, "Targeting signal transduction in pancreatic cancer treatment." *Expert Opinion on Therapeutic Targets*, vol. 11, no. 5, pp. 673–694, 2007.

[78] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya, "The kegg databases at genomenet," *Nucleic Acids Res*, vol. 30, no. 1, pp. 42–46, 2002.

[79] J. Lin, C. M. Gan, X. Zhang, S. Jones, T. Sjoblom, L. D. Wood, D. W. Parsons, N. Papadopoulos, K. W. Kinzler, B. Vogelstein, G. Parmigiani, and V. E. Velculescu, "A multidimensional analysis of genes mutated in breast and colorectal cancers," *Genome Research*, vol. 17, no. 9, pp. 1304–1318, 2007.

[80] K. Chin, S. DeVries, J. Fridlyand, P. T. Spellman, R. Roydasgupta, W. L. Kuo,

A. Lapuk, R. M. Neve, Z. Qian, T. Ryder, F. Chen, H. Feiler, T. Tokuyasu, C. Kingsley, S. Dairkee, Z. Meng, K. Chew, D. Pinkel, A. Jain, B. M. Ljung, L. Esserman, D. G. Albertson, F. M. Waldman, and J. W. Gray, "Genomic and transcriptional aberrations linked to breast cancer pathophysiologies," *Cancer Cell*, vol. 10, no. 6, pp. 529–541, 2006.

[81] H. Parkinson, U. Sarkans, M. Shojatalab, N. Abeygunawardena, S. Contrino, R. Coulson, A. Farne, G. G. Lara, E. Holloway, M. Kapushesky, P. Lilja, G. Mukherjee, A. Oezcimen, T. Rayner, P. Rocca-Serra, A. Sharma, S. Sansone, and A. Brazma, "Arrayexpress–a public repository for microarray gene expression data at the ebi," *Nucleic Acids Research*, vol. 33, no. Database issue, pp. 553–555, 2005.

[82] V. Matys, O. V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. E. Kel, and E. Wingender, "Transfac and its module transcompel: transcriptional gene regulation in eukaryotes," *Nucleic Acids Research*, vol. 34, no. Database issue, pp. 108–110, 2006.

[83] A. Kossenkov, F. J. Manion, E. Korotkov, T. D. Moloshok, and M. F. Ochs, "ASAP: automated sequence annotation pipeline for web-based updating of sequence information with a local dynamic database," *Bioinformatics*, vol. 19, no. 5, pp. 675–676, 2003.

[84] W. Huber, A. von Heydebreck, and M. Vingron, "Analysis of microarray gene expression data," in *Handbook of Statistical Genetics*. Wiley, 2003.

[85] T. Ideker, V. Thorsson, A. F. Siegel, and L. E. Hood, "Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data," *Journal of Computational Biology*, vol. 7, no. 6, pp. 805–817, 2000.

[86] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed, "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, no. 2, pp. 249–264, 2003.

[87] Y. Chen and E. Dougherty, "Ratio-based decisions and the quantitative analysis of cdna microarray images," *Journal of Biomedical Optics*, no. 2, pp. 364–374, 1997.

[88] E. Kuhn and M. Lavielle, "Coupling a stochastic approximation version of EM with an MCMC procedure," *ESAIM: Probability and Statistics*, vol. 8, pp. 115–131, 2004.

[89] Y. Wu, X. Zhang, J. Yu, and Q. Ouyang, "Identification of a topological characteristic responsible for the biological robustness of regulatory networks," *PLoS computational biology*, vol. 5, no. 7, 2009.

[90] J. Binder, D. Koller, S. J. Russell, and K. Kanazawa, "Adaptive probabilistic networks with hidden variables," *Machine Learning*, vol. 29, pp. 213–244, 1997.

[91] C. M. Hurvich and C. L. Tsai, "Regression and time series model selection in small samples," *Biometrika*, no. 76, pp. 297–307, 1989.

[92] V. Vapnik and A. Chervonenkis, *Theory of Pattern Recognition.* Nauka, Moscow, 1974.

[93] W. Zucchini, "An introduction to model selection," *Journal of Mathematical Psychology*, vol. 44, no. 1, pp. 41–61, 2000.

[94] K. P. Burnham and D. R. Anderson, *Model Selection and Inference: A Practical Information-Theoretic Approach.* Springer, 1998.

[95] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society*, no. 61, pp. 611–622, 1999.

[96] S. Roweis and Z. Ghahramani, "A unifying review of linear gaussian models," *Neural Computation*, no. 11, pp. 305–345, 1999.

[97] Y. Weiss and W. T. Freeman, "Correctnes of belief propagation in gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, 2001.

[98] A. Rothman, P. Bickel, E. Levina, and J. Zhu, "Sparse permutation invariant

covariance estimation," *Electronic Journal of Statistics*, vol. 2, pp. 494–515, 2008.

[99] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[100] O. Banerjee, Laurent, and E. Ghaoui, "First-order methods for sparse covariance selection," *SIAM Journal on Matrix Analysis and its Applications*, 2007.

[101] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[102] M. I. Jordan, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, pp. 181–214, 1994.

[103] C. M. Bishop and Tipping, "A Hierarchical Latent Variable Model for Data Visualization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 281–293, 1998.

[104] N. Lawrence and A. Hyvärinen, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[105] N. D. Lawrence, "Hierarchical gaussian process latent variable models," in *International Conference in Machine Learning*, 2007.

[106] C. Ek, P. Torr, and N. Lawrence, "Gaussian process latent variable models for human pose estimation," *Machine Learning for Multimodal Interaction*, pp. 132–143, 2008.

[107] C. Liu, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, no. 10, pp. 2271–2285, 2003.

[108] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Handwritten digit recognition with a committee of deep neural nets on GPUs," *ArXiv*, 2011. [Online]. Available: http://arxiv.org/abs/1103.4487

[109] F. Fleuret, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, no. 5, pp. 1531–1555, 2004.

[110] Y. Lecun and C. Cortes, "The MNIST database of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[111] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, pp. 6567–6572, 2002.

[112] A. C. Tan, D. Q. Naiman, L. Xu, R. L. Winslow, and D. Geman, "Simple decision rules for classifying human cancers from gene expression profiles," *Bioinformatics*, vol. 21, no. 20, pp. 3896–3904, 2005.

[113] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J.

Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745–6750, 1999.

[114] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, and M. L. Loh, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, pp. 531–537, 1999.

[115] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub, "Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, no. 1, pp. 68–74, 2002.

[116] G. J. Gordon, R. V. Jensen, L.-L. Hsiao, S. R. Gullans, J. E. Blumenstock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno, "Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma," *Cancer Research*, pp. 4963–4967, 2002.

[117] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, and J. P. Richie, "Gene expression correlates

of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[118] R. O. Stuart, W. Wachsman, C. C. Berry, J. Wang-Rodriguez, L. Wasserman, I. Klacansky, D. Masys, K. Arden, S. Goodison, M. McClelland, Y. Wang, A. Sawyers, I. Kalcheva, D. Tarin, and D. Mercola, "In silico dissection of cell-type-associated patterns of gene expression in prostate cancer," *Proceedings of the National Academy of Sciences*, vol. 101, no. 2, pp. 615–620, 2004.

[119] J. B. Welsh, L. M. Sapinoso, A. I. Su, S. G. Kern, J. Wang-Rodriguez, C. A. Moskaluk, H. F. Frierson, and G. M. Hampton, "Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer," *Cancer Research*, vol. 61.

[120] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub, "Multiclass cancer diagnosis using tumor gene expression signatures," *Proceedings of the National Academy of Sciences*, vol. 98, no. 26, pp. 15 149–15 154, 2001.

[121] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend,

"Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

[122] I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O. P. Kallioniemi, B. Wilfond, A. Borg, J. Trent, M. Raffeld, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Kononen, L. Bubendorf, W. Fehrle, S. Pittaluga, S. Gruvberger, N. Loman, O. Johannsson, H. Olsson, and G. Sauter, "Gene-expression profiles in hereditary breast cancer," *The New England Journal of Medicine*, vol. 344, no. 8, pp. 539–548, 2001.

[123] M. E. Troyanskaya, M. E. Garber, P. O. Brown, D. Botstein, and R. B. Altman, "Nonparametric methods for identifying differentially expressed genes in microarray data," in *Gene Selection via Discretized Gene Expression Profiles and Greedy Feature Elimination*, 2002, pp. 1454–1461.

[124] X. Lin, B. Afsari, L. Marchionni, L. Cope, G. Parmigiani, D. Q. Naiman, and D. Geman, "The ordering of expression among a few genes can provide simple cancer biomarkers and signal BRCA1 mutations," *BMC Bioinformatics*, vol. 10, 2009.

[125] D. W. Parsons, S. Jones, X. Zhang, J. C. Lin, R. J. Leary, P. Angenendt, P. Mankoo, H. Carter, I. M. Siu, G. L. Gallia, A. Olivi, R. McLendon, B. A. Rasheed, S. Keir, T. Nikolskaya, Y. Nikolsky, D. A. Busam, H. Tekleab, J. Diaz,

L. A., J. Hartigan, D. R. Smith, R. L. Strausberg, S. K. Marie, S. M. Shinjo, H. Yan, G. J. Riggins, D. D. Bigner, R. Karchin, N. Papadopoulos, G. Parmigiani, B. Vogelstein, V. E. Velculescu, and K. W. Kinzler, "An integrated genomic analysis of human glioblastoma multiforme," *Science*, vol. 321, no. 5897, pp. 1807–1812, 2008.

[126] TCGA, "Comprehensive genomic characterization defines human glioblastoma genes and core pathways," *Nature*, vol. 455, no. 7216, pp. 1061–1068, 2008.

Vita

Erdem Yörük received the B.S. degree in Electrical and Electronic Engineering in 2002, from Boğaziçi University, Turkey. He earned the M.S. degree from the same department in 2004, with his master's thesis titled "Shape-Based Hand Recognition". In 2005, he enrolled in the Department of Applied Mathematics and Statistics Ph.D. program at Johns Hopkins University, and was a member of Center for Imaging Science and Institute for Computational Medicine. He won the best student paper award in IEEE 12^{th} conference on Signal Processing and Communications Applications in 2004, and received H. Cohen Fellowship in 2009. His research focuses on computational biology, computer vision, machine learning and graphical Markov models.