Swiss Federal Institute of Technology Zurich

Seminar for Statistics

Department of Mathematics

Master Thesis *February* 2021 – *August* 2021

Marco Hassan

Parameter Learning in Bayesian Networks under Uncertain Evidence – An Exploratory Research.

Submission Date: 30/08/2021

Co-Adviser: *Markus Kalisch*
Adviser: *Radu Marinescu*

To my family, my beloved ones and my dear friends
that make difficult times easier to face.

ABSTRACT

This study deals primarily with the topic of parameter learning in Bayesian Networks. In its first sections it proposes a general literature overview of parameter learning in the case of Maximum Likelihood Estimation under complete evidence. A general introduction to the concept of likelihood decomposability is provided before rigorously exposing the EM-algorithm as a viable choice for parameter estimation when the decomposability property does not hold and the maximization problem is a nasty multimodal function. Given the properties of the EM-algorithm, we will argue that it is possible to generalize the algorithm to deal with the case of parameter learning in a full-bayesian learning setting by adjusting the maximization step. The second part of study leverages the classic theory exposed in the previous sections to deal with the case of parameter learning in the case of uncertain evidence. Augmenting the arguments proposed in Wasserkrug et al. (2021) the study will argue that upon finding a constrained joint distribution for the probabilistic graphical model that satisfies the constrains imposed by the uncertain evidence, the EM-algorithm might be used for obtaining a sensible network parameterization with slight adjustments as the virtual evidence method of Pearl (1987). The study shows then by example that that such methods are easily integrable in classical statistical software as in the extension of the merlin engine.

***K*eywords** Graphical Models · Iterative Methods · Bayesian Networks · Bayesian Statistics · Parameter Learning · Bayesian Learning · Uncertain Evidence · EM-algorithm · I-projection · Inference · Clique Algorithms

Contents

List of Figures

List of Algorithms

List of Tables

Notation

This section introduces some general notation style used in the script.

Sets - i.e. sets of random variables/nodes, edges - are expressed via the capital italic notation as $\mathscr{X}$, $\mathscr{Y}$. We denote a single realization of the random variables within the set as x, y respectively. Finally we denote a set of realizations via the following italic notation $\mathcal{X}$, $\mathcal{Y}$ respectively.

When expressing the probability of a set of realizations, say $P(\mathcal{D})$, we intend the likelihood of that set of realizations under the given probability function.

Random variables themselves are defined via capital letters, while the lower case of the respective random variables defines domain realization of the variables. For instance X_i represents a particular random variable while x_i represents the ralization of it. Multivariate random variables are expressed in bold face as **X**.

We denote a realization for the entire Bayesian Network, i.e. a realization containing evidence for each random variable present in the set $\mathscr{X}$ representing the network variables, by ξ.

Moreover, we denote as $\mathbf{PA}_i$ the multivariate random variable representing the parent random variables for a particular node X_i. Note that a parent random variable is defined as a random variable with a directed edge pointing to the node of interest. We denote a realization of such multivariate random variable as pa_i. Note as well that pa_{ij} represents the realization of the jth parent random variable of x_i, i.e. a realization of PA_{ij}.

In the case of multiple samples of the same random variable an index within square brackets denotes the particular sample of interest, say for instance $\xi[1], ..., \xi[M]$ in the case of M Bayesian Network samples.

Note that the $Val(\mathbf{X})$ operator represents the set of all of the possible realizations of the random variable **X**. When such operator is used on a set of realizations, say for instance a set of missing evidence H[m] = {H$_1$[m] = ?, . . . , H$_k$[m] = ?} - i.e. when using $Val(\mathcal{H}[m])$ - we do intend the set of possible domain realizations of the random variables in the set.

Finally when indexing the parameters of interest, the subscripts represent the index of the random variable as usual, while the superscripts denotes the realization for that particular variable. An example helps to illustrate the concept. $\theta_{x_1^0}, \theta_{x_2^1}$ represents the parameter governing the case of $x_1 = 0$ and $x_2 = 1$ respectively.

1 Bayesian Networks - Overview and Script Outline

Probabilistic models based on directed acyclic graphs (DAGs) have been consistently studied and researched since the first appearance in the field of genetics in Wright (1921).

An important moment in the history of such models and the Bayesian Networks subclass was, as argued by Pearl (2011), during the late 1970s when such models emerged as the method of choice when dealing with uncertainty in reasoning and expert systems, effectively starting to replace the usage of symbolic artificial intelligence and rule-based schema.

As argued in Pearl (2011), an especially significant property of such models that marked a turn-point in the modeling of real world systems, was the fact that such models shifted the focus of the modeling exercise from reasoning processes focusing on the flow of information (think at classical AI systems including neural networks) to direct world representation, where edges may represent real-world causal connections.

This, together with the possibility to associate probabilistic statements to events modeled by such graphs via bidirectional inference possibilities makes the point for the usage and research of such models in various subfields of artificial intelligence.

Given such properties and the fact that the mathematical machinery necessary to deal with such graphs spans multiple distinct subfields[1], the models attracted the interest of multiple researchers. It comes as no surprise the fact that there is a vibrant research community dealing with such topic. Alone in the last 10 (20) years the numbers of published papers on the topic increased by x4 (x24) factor[2], with a total of ~100'000 published papers on the topic of Bayesian Networks alone for the year 2020.

In this sense, a thorough review of the ongoing research in the field would be impossible. While important progress has been done in researching the computational aspects of inference, learning and representation of the networks, most of the research so far is focusing on an idealized world with complete or missing evidence. Hence, despite the wealth of research, fundamental issues when dealing with uncertain evidence for modeling Bayesian Networks have been only marginally addressed that far.

This will pose the focus to this script. By a rigorous literature review as well as some minor theoretical contributions we will try to partially address such a gap in the Bayesian Network modeling research. In this sense, we will start with a formal definition of Bayesian Networks in the next sub-section before turning to a rigorous definition of the different types of uncertain evidence from which we will expand on.

1.1 Bayesian Networks - Formal Definition

Before starting to dig into the material we provide a general definition of a Bayesian Network as in Pearl (2011).

Definition 1 *Bayesian Network: A Bayesian Network is a directed acyclic graph (DAG) $G(\mathscr{V}, \mathscr{X})$ whose nodes $\mathscr{X}$ represent random variables in the Bayesian sense - i.e. they can be observable quantities, latent variables, unknown parameters or hypotheses.*

[1] Think for instance to the necessary components of mathematical statistics, information theory, graph theory and optimization.

[2] Data from app.dimensions.ai.

On the top of it, edges $\mathscr{V}$ represent the conditional dependencies among the nodes; i.e. nodes that are not connected represent random variables that are conditionally independent.

Especially important is the characterization of the conditional independence relation among the random variables.

In order to see this, recall that when modeling a Bayesian Network the object of interest is the complete probabilistic model of the random variables domain - i.e. the probability of every possible event as defined by the values of all of the variables.

Consider the case where you would want to represent the joint distribution of a set of random variables $\mathscr{X} = \{X_1, ..., X_n\}$.

Then, given a parametric distribution on each random variable in the set it is possible to extract the functional form of the probability distribution for the joint distribution. However, there is an issue when trying to directly model the joint distribution without explicitly leveraging and modeling the conditional independence structures among the random variables. This due to the exponential number of possible events in the domain that would require an exponential number of parameters. Standard examples may be found in Koller and Friedman (2009).

Yet when considering the conditional structure among the variables in the set $\mathscr{X}$, it is possible to leverage the *global semantics* of the network to represent the joint-distribution in compact form as in Pearl (2011):

$$P(x_1, ..., x_n) = \prod_i P(x_i|pa_i)$$

where pa_i represents the realization for the parents nodes of node x_i.

It is then possible to see that as well framed by Pearl (2011): "Provided that the number of parents of each node is bounded, it is then immediate to see that the number of parameters required grows only linearly with the size of the network, whereas the joint-distribution itself grows exponentially."

This leads us to a second possible definition of Bayesian network as framed by Koller and Friedman (2009), making more concrete the idea of a Bayesian Network as a compact data structure representing the complete probabilistic model of the random variables domain:

Definition 2 *Joint-Density Factorization: Let $\mathscr{G}$ be a Bayesian Network graph over the variables $X_1, ..., X_n$.*

We say that a distribution P over the same factorization space factorizes according to $\mathscr{G}$ if P can be expressed as a product

$$P(x_1, ..., x_n) = \prod_i P(x_i|pa_i^{\mathscr{G}})$$

This equation is called the chain rule for Bayesian networks, and the $P(x_i|pa_i^{\mathscr{G}})$ terms are called conditional probability distributions (CPDs).

Definition 3 *Bayesian Network: A Bayesian Network is a pair $\mathscr{B} = (\mathscr{G}, P)$ where P factorizes over $\mathscr{G}$, and where P is specified as a set of CPDs associated with $\mathscr{G}$s nodes. The distribution P is often annotated $P_{\mathscr{B}}$.*

Given such a definition it is immediate to distinguish the three main puzzles that is necessary to solve when working with Bayesian Networks, respectively:

> (i) the structure riddle, where you specify either by domain knowledge or via data-driven learning the shape of the Bayesian Network graph $\mathscr{G}$. Both are challenging tasks especially in the case the user aims to fulfill both completeness and soundness of I-maps as defined in Koller and Friedman (2009).
>
> (ii) the parameterization learning riddle, where given some evidence we will try to specify the parameters of the CPDs of the network in the most sensible way such that we well describe the probabilistic structure of the random variables domain.
>
> (iii) the inference riddle, where given the network structure and parameterization you would apply Bayes Theorem for performing bidirectional inference to get to the probabilistic occurrence of some random variables evidence. Again a very demanding task, given that it was proven that the exact solution of such problem is NP-hard, such that it is often necessary to rely on approximate inference methods as in Pearl (1987) or variational methods as in Jordan et al. (1999).

In this script, we will focus on (ii) taking (i) as given and leveraging standard results from the literature as in Koller and Friedman (2009) for (iii).

We turn next to a formal definition of evidence and especially of *uncertain evidence* necessary for the correct outline of the research presented in this script.

1.2 Types of Evidence

The most basic case of evidence is the one of complete evidence. This occurs when we are provided with complete observations of the network, i.e. when it is possible to observe a certain realization for each random variable in the domain of the network.

One more interesting case is the one treated by Mrad et al. (2015), Wasserkrug et al. (2021). The argument posed by the authors is that under many settings complete evidence is not possible.

In many cases there might be a hiding mechanism active that might hide some of the realizations. Think for instance at a malfunctioning sensor that sporadically measures input. Or think for instance at medical settings where different patients might be measured different variables.

Albeit the case of missing evidence greatly alters the way through which it is possible to learn the parameters of the network, there are multiple possible solutions to estimate parameters and come to local maxima. We will address one of such methods in the next chapter.

A more interesting case is posed by *uncertain evidence* as introduced by Mrad et al. (2015). The authors distinguish three types of non-complete evidence:

(i) likelihood evidence

(ii) fixed probabilistic evidence

(iii) non-fixed probabilistic evidence

We will use throughout this document the definition as in Mrad et al. (2015) which we will briefly summarize next.

Definition 4 *Hard evidence: A finding on a variable commonly refers to an instantiation of the variable. This can be represented by a vector with one element equal to 1, corresponding to the state the variable is in, and all other elements equal to zero. This type of evidence is usually referred to as hard evidence.*

Definition 5 *Uncertain evidence: evidence that cannot be represented by a vector as in the hard evidence case.*

Definition 6 *Likelihood evidence: in such type of evidence there is uncertainty about the veracity of an observation, such as, for example, the information given by an imperfect sensor. Such uncertainty is expressed in terms of relative likelihood of observing one realization vis à vis another one.*

Definition 7 *Probabilistic evidence: we talk about probabilistic evidence when we have a set of probabilistic finding on one or multiple random variables X_i in the network. The structure of the probabilistic finding is specified by a local probability distribution $R(X_i)$.*

Note that a probabilistic finding $R(X_i)$ on a variable $\mathrm{X_i}$ of a Bayesian network replaces any prior belief or knowledge on $\mathrm{X_i}$. As a consequence, the prior $P(X_i)$ resulting from Bayesian Network inference is not used in the propagation of $R(X_i)$, and any previous finding or belief on $\mathrm{X_i}$ is lost.

Moreover, note the following distinction between *fixed* and *non-fixed* probabilistic evidence:

Definition 8 *Fixed (Non-fixed) Probabilistic evidence: A probabilistic finding is fixed (non-fixed) when the distribution $R(X_i)$ can not be (can be) modified by the propagation of other findings.*

Such that it is all about how the *arrival of evidence*, as depicted in the following schema from Mrad et al. (2015), can update the cognitive state:

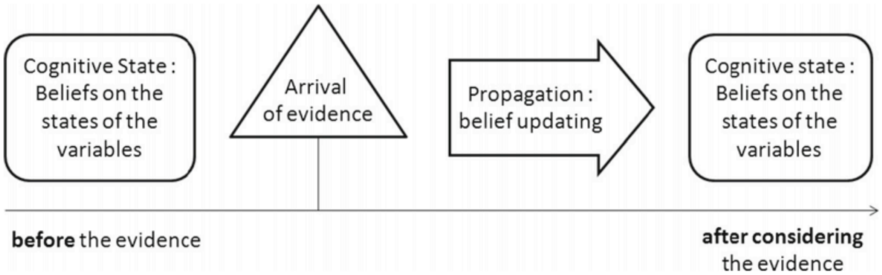

Cognitive State : Beliefs on the states of the variables
Arrival of evidence
Propagation : belief updating
Cognitive state : Beliefs on the states of the variables
before the evidence
after considering the evidence

Figure 1: Inference Loop as in Mrad et all.

Summarizing, in simple terms, we differentiate the following three cases for the above:

1. In fixed probabilistic evidence we specify a probabilistic evidence *all things considered*. This means that even after new evidence is observed on any other random variable in the network, we do not update the cognitive state specified by the fixed probabilistic evidence.

2. In non-fixed probabilistic evidence we consider the current structure of the tree such that for the current state of the network, the conditional probability distribution is specified by the specified probabilistic evidence. Further in-coming evidence that will alter the network probabilistic structure will affect the cognitive state of the current node.
3. In likelihood evidence we do not consider any prior information. I.e. we simply specify a local likelihood ratio for a particular evidence and we still have to run the inference step for the current state to get the final cognitive state. I.e. as mentioned by Mrad et al. (2015) in contrast to probabilistic evidence which remains unchanged by updating the observed variables, likelihood evidence has to be combined with previous beliefs in order to update the belief in the observed variable(s).

Given the general understanding on Bayesian Networks and the different types of uncertain evidence we will provide in the next section a literature review for the case of parameter learning in Bayesian Networks in the case of complete, missing and uncertain evidence. We will finally propose the research outline for this script.

1.3 Literature Review and Research Outline

Irrespective of the type of uncertainty, we distinguish among two class of methods in order to estimate parameters in Bayesian Networks, namely the Maximum Likelihood Estimation (MLE) Spiegelhalter and Lauritzen (1990) and Bayesian Learning Smith and M (2001). The theoretical framework on which the two models rely is different. While in MLE we assume the parameter of interest to be a fixed but unknown variable, in Bayesian Learning we treat the parameter of interest themselves as a random variable.

Both such methods have been extensively studied in statistical research.

For MLE we have to guarantee the existence of such an estimator and then perform the optimization exercise in order to find the maximum likelihood estimator. In this sense a particular important result that we will leverage in the script is the realization that in the case of an exponential family as treated by Barndorff-Nielsen (1978), we can obtain the MLE by solving for the reverse I-projection (or M-projection) as defined in Csiszár (1975).

In contrast to that in the case of Bayesian Learning the situation is more complex. On the one hand it is necessary to define a prior probability distribution on the parameter of interest before the information arising from the observed evidence is propagated. This can be done either with the help of subject matter experts by expressing some degree of knowledge gained over the years, or by the usage of non-informative priors as in Syversveen (1998). On the other hand, it is necessary to decide on a point estimator for the parameter given its posterior distribution. Standard point estimators of choice are the maximum a posteriori estimator or the first moment of the posterior distribution. Important is to realize that we might not be able to integrate over every posterior such that it will be necessary to rely on approximate methods for computing the moments of interest. Or, a viable different solution, is to restrict the modeling approach to exponential families with a sensible choice of conjugate priors, as defined in Schlaifer and Raiffa (1961), such that the posterior will be a standard distribution which moments and mode can be expressed by functions and coded efficiently in statistical software.

Given the general theoretical framework for the two parameter estimation techniques mentioned above, we are left with a generalization of the above to the probability function of Bayesian Networks. Seminal was the work of Spiegelhalter and Lauritzen (1990) who laid the foundation of the learning problem by expressing the assumption of global parameter independence coming to the central notion of decomposability of the likelihood function that we are going to treat in the next sections. Based on such work multiple papers were published expanding the theory for different classes of Bayesian Networks such as tree-CPDs Buntine (1993), linear Gaussian BN Heckerman et al. (1995) as well as different learning settings such as learning with parameter sharing networks, hierarchical Bayesian models and non-parametric estimation.

The theoretical foundation for the research mentioned that far was the one of complete evidence. Leaving such an ideal case and turning to missing evidence where missing evidence occurs for some network random variables, some further reasoning is necessary. Rubin (1976) and Little (1976) started to reason about the notion of *missing completely at random* and *missing at random* data, a topic we will re-encounter in later section of this script. Given such a distinction it is possible to set out the fundamentals for the parameter learning task. We will show that in the case of *missing completely at random* data some of the decomposability properties of the likelihood function under complete data are lost such that it is necessary to rely on some tailored learning technique such as the EM-algorithm first presented by Dempster et al. (1977) and its application to graphical models as in Lauritzen (1995). We will show then that under missing evidence such an algorithm will be correct and converge to a local maxima of the joint distribution of the Bayesian Network.

Moving to the case of uncertain evidence, we will first build on the theory of Wasserkrug et al. (2021) to augment the EM-algorithm in order to deal with *likelihood evidence* keeping the correctness and convergence properties of the algorithm. This will be done by exploiting the idea of augmenting Bayesian Network via virtual evidence nodes as outlined by Pearl (2014).

Then, in a second step, we will generalize the theory of Wasserkrug et al. (2021) in order to deal with the Bayesian Learning setting in the case of maximum a posteriori estimator. We will show there that it is possible to apply the EM algorithm in such a setting by adjusting the maximization step in a way that its correctness and convergence properties are guaranteed.

Finally, when considering the case of *probabilistic evidence* it is important to reason on the iterative proportional fitting procedure and its possible extensions for the case of parameter learning via the EM-algorithm. In this sense we will reason on how we might generalize EM algorithm to the case of an arbitrary number of probabilistic evidence by leveraging the work of PENG et al. (2010), Meng (2016) and by trying to incorporate such algorithms into the EM-algorithm.

The script continues as follows. Section 2 treats the most basic and fundamental case of parameter learning under complete evidence. It will outline the global decomposability property of the likelihood such that the maximization task of the MLE is facilitated. Section 3 exposes the case of missing evidence and discusses how in such case the global decomposability property is lost such that the resulting likelihood for the Bayesian Network is a multi-modal function that is difficult to optimize. Moreover, it introduces and discusses the mathematics of the EM-algorithm as a way to deal with such a case and reach a local optima. Section 4 discusses the case of Bayesian Learning and reasons about the EM-algorithm and its applicability to deal with the case of maximum a posteriori estimators. Section 5 discusses few examples of the application of the algorithms to the case of exponential families Bayesian Networks CPDs. Section 6 discusses about some practical concerns of applying the algorithms of section 3, pointing to the possibility of applying some numerical techniques for the M-step of the EM-algorithm. The remaining sections deal with the case of parameter learning under uncertain evidence. Section 7 discusses the theoretical fundamentals for parameter learning in the case of likelihood evidence. It treats both the case of plain MLE parameter learning as well as the case of maximum a posteriori parameter learning in a Bayesian Learning setting. Finally section 8 deals with the case of parameter learning in the case of probabilistic evidence both in the case of single and multiple probabilistic evidence statements. Section 9 wraps up and concludes.

2 Learning under Complete Evidence

In this section we will lay out the fundamental characteristics of parameter learning under complete evidence. On the one hand we will propose the global decomposition property as in Spiegelhalter and Lauritzen (1990). On the other hand we will introduce the parameter independence condition; this is a fundamental property that Bayesian Networks need to display in order to fulfill the decomposition properties. In fact, we will see in the next section that in the case of missing evidence such a property is not fulfilled such that the decomposability of Bayesian Networks is lost.

The major result of Spiegelhalter and Lauritzen (1990), was the realization that in the case of complete data the likelihood of the Bayesian Network decomposes over the set of local likelihood functions of the individual nodes X_i. Hence, despite the fact that the likelihood function of Bayesian Networks is a product of multiple chained CPDs, it is possible to estimate the parameters of each CPDs locally in order to get an overall network parameterization fulfilling the functional requirements of the estimation technique.

In order to see that, consider the following network $\mathscr{B} = (\mathscr{G}, P)$ with θ parameterization and a set of complete evidence $\mathcal{D}$ consisting of sample instances $\xi[1], ..., \xi[M]$.

We note here, that in this script we will work under the assumption that the parameters in the network are disjoint. Meaning that there is no *global parameter sharing* as well as no *local parameter sharing* in the network as described in Koller and Friedman (2009).

Given such network structure it is possible to express the network overall probability via chain rule as:

$$P_\theta(x_1, ..., x_n) = \prod_i \prod_m P_{\theta_i}(x_i[m] | pa_i^{\mathscr{G}}[m])$$

Noting now that it is possible to invert the order of multiplication we get the following global likelihood decomposition for the Bayesian Network decomposition.

$$\begin{aligned} L(\theta : \mathcal{D}) &= \prod_m P_\theta(x_1[m], ..., x_k[m]) \\ \prod_m P_\theta(x_1[m], ..., x_k[m]) &= \prod_m \prod_i P_{\theta_i}(x_i[m] | pa_i^{\mathscr{G}}[m]) \\ \prod_m P_\theta(x_1[m], ..., x_k[m]) &= \prod_i [\prod_m P_{\theta_i}(x_i[m] | pa_i^{\mathscr{G}}[m])] \\ &= \prod_i L(\theta_i : \mathcal{X}_i | \mathcal{PA}_i) \end{aligned}$$

It is then immediate to see that when solving for the MLE parameterization of the above you might well get the parameters by locally solving for the MLE estimators of the local likelihoods $L(\theta_i : \mathcal{X}_i | \mathcal{PA}_i)$.

A similar reasoning holds for the case of Bayesian Learning. In such a case we require on the top of complete data the property of *a priori independent* parameters defined as follows as in Koller and Friedman (2009)

Definition 9 *A Priori Global Parameter Independece: Let G be a Bayesian network structure with parameters* $\theta = (\theta_{X_1|\boldsymbol{PA}_{X_1}}, ..., \theta_{X_n|\boldsymbol{PA}_{X_n}})$.

A prior $P(\theta)$ *is said to satisfy global parameter independence if it has the form:*

$$P(\theta) = \prod_i P(\theta_{X_i|\boldsymbol{PA}_{X_i}})$$

Such that with *a-priori global parameter independence* knowing the value of one parameter in a CPD does not add any information regarding another CPD parameter value.

With this notion in mind and the global likelihood decomposition property it is possible to see that for Bayesian Learning under complete data it holds that:

$$\begin{aligned}
\prod_m P_\theta(x_1[m], ..., x_k[m]) &= \prod_i L(\theta_i : \mathcal{X}_i|\mathcal{PA}_i) \\
P(\theta) &= \prod_i P(\theta_{X_i|\mathbf{PA}_{X_i}}) \\
\prod_m P(\theta|d[m]) &= \prod_i L(\theta_i : \mathcal{X}_i|\mathcal{PA}_i) * L(\theta_{X_i|\mathbf{PA}_i}) * \prod_m \frac{1}{P(d[m])}
\end{aligned}$$

Such that once more we might for instance be able to compute the maximum a posteriori parameterization for the entire network, by taking the maximum a posteriori parameterization of individual CPDs.

As a final note, we stress here the point that, in a similar line of reasoning, if when observing the data the parameterization of a *local* CPD is d-separated - such that, after observing complete data, $\theta_{Y|pa_{ij}}$ is independent from $\theta_{Y|pa_{il}}$ for all local parents $j, l = 1, ..., p$ with $i \neq l$ - then a network satisfies the *local decomposability property*. In such a case it would then hold for the parameter local independence:

$$\begin{aligned}
P(\theta) &= \prod_i P(\theta_{X_i|\mathbf{PA}_i}) \\
&= \prod_i \prod_j P(\theta_{X_i|PA_{ij}})
\end{aligned}$$

And correspondingly for your likelihood

$$
\begin{aligned}
L(\theta : \mathcal{D}) &= \prod_{i}^{K}[\prod_{m}^{M}[\prod_{j}^{P} P_{\theta_{ij}}(x_i[m]|pa_{ij}^{\mathscr{G}}[m])]] \\
&= \prod_{i}^{K}[\prod_{j}^{P}[\prod_{m}^{M} P_{\theta_{ij}}(x_i[m]|pa_{ij}^{\mathscr{G}}[m])]] \\
&= \prod_{i}\prod_{j} L(\theta_{ij} : \mathcal{X}_i|\mathcal{PA}_{ij})
\end{aligned}
$$

Such that it would ultimately hold for Bayesian Learning

$$
\prod_{m} P(\theta|d[m]) = \prod_{i}\prod_{j} L(\theta_{ij} : \mathcal{X}_i|\mathcal{PA}_{ij}) * L(\theta_{X_i|PA_{ij}}) * \prod_{m} \frac{1}{P(d[m])} \tag{1}
$$

It is then clear that in such a case it is then possible to maximize at an even more narrow local level reducing the difficulty of the optimization task.

We turn next to the case of missing evidence where we are going to see that such neat properties fade away such that it will be necessary to rely on more sophisticated techniques in order to perform the learning task. In fact, as we will see it will not be possible anymore to perform local operations for obtaining a global solution.

3 On Missing Evidence

In the case of missing evidence we have two types of findings for the random variables in our network $G(\mathscr{V}, \mathscr{X})$.

Once more, consider $m = 1, ..., M$ instances of your network. Then, on the one hand, you will have observed random variables realizations - hard evidence - $d[m]$ for a subset of variables $\mathscr{D} \subset \mathscr{X}$. On the other hand you will have missing or non-observed findings $h[m]$ for a subset of variables $\mathscr{H} := \mathscr{X} \setminus \mathscr{D}$.

As both of the parameter learning techniques as presented in 1.3 involve a likelihood term, the question is on the way such likelihood term can be represented in the case of missing evidence.

In order to answer such a question we will shortly distinguish among data *missing completely at random* and data *missing at random* as reasoned in Little (1976), Rubin (1976).

We start by defining a data hiding mechanism $P_\psi(O_{X_i}|X_i)$ where O_{X_i} is a binary random variable representing whether the random variable X_i is observed or missing. It then follows that it is possible to express the probability of the random variable X_i realization through $P_{missing}(X_i, O_{X_i}) = P_\theta(X_i) * P_\psi(O_{X_i}|X_i)$.

Given such a model we turn to the definition of the two essential hiding mechanisms leveraging once more the work of Koller and Friedman (2009):

Definition 10 *Missing Completely at Random: A missing data model $P_{missing}$ governing a random variable X_i is missing completely at random (MCAR) if $P_{missing} \models (X_i \perp O_{X_i})$. I.e. in the case of marginal independence among the observation mechanism and the random variable.*

Given such property it is immediate to see that the likelihood decomposes on terms depending on the parameters of interest θ and on terms governing the data hiding mechanism ψ. If the ultimate interest of your study is on the parameterization of the data governing mechanism of the random variables X_i, i.e. on θ, it is then obvious that it is possible to focus on such portion of the likelihood function forgetting about the data hiding mechanism.

Defining then the set of random variables $\mathscr{Y} = \{Y_1, ..., Y_n\}$, where $Val(Y_i) = Val(X_i) \cup \{?\}$, where $\{?\}$ represents a missing evidence, we can define the following data hiding mechanism:

Definition 11 *Missing at Random: A data model $P_{missing}$ is missing at random (MAR) if for all observations with $P(y) > 0$, i.e. for all possible realizations, and for all $h \in Val(\boldsymbol{H})$, we have with $d \in Val(\boldsymbol{D})$ observed evidence, that $P_{missing} \models (h \perp o_X|d)$.*

Or in other words, we talk about data *missing at random* when conditioning on the observed evidence we have conditional independence among the hidden/non-observed variables, and the hiding mechanism.

As pointed out by Koller and Friedman (2009), MAR is a powerful condition as it is a necessary condition in order to write the likelihood function under missing evidence as a product of terms involving the parameters governing the probabilistic structure of the random variables of interest X_i and the hiding mechanism O_{X_i}. It is then possible, as in the case of *missing completely at random*, to distinguish between the two likelihood terms and just focus on the likelihood of the observed variables when estimating the θ parameters.

In order to see this note first that in the case of MAR, the observation pattern o_X gives no additional information about the hidden variables given the observed variables, that is:

$$P_{missing}(h|d, o_X) = P_{missing}(h|d)$$

It holds then that

$$\begin{aligned}
P_{missing}(y) &= \sum_h P_\theta(h, d) * P_\psi(o_X|h, d) \\
&= \sum_h P_\theta(h, d) * P_\psi(o_X|d) \\
&= P_\psi(o_X|d) * \sum_h P_\theta(h, d) \\
&= P_\psi(o_X|d) * P_\theta(d)
\end{aligned}$$

Such that you can easily see that if $P_{missing}$ is MAR then $L(\theta, \psi : \mathcal{D})$ decomposes into two terms $L(\theta : \mathcal{D}), L(\psi : \mathcal{D}, \mathcal{O}_X)$.

Noting now that as we can always reach the *MAR* condition by expanding a Bayesian Network, we will assume for the theory in this script that the Bayesian Network of interest presenting missing evidence satisfies *MAR*, such that the question of interest will be the functional form of the likelihood $P_\theta(d)$ in the case of missing data, the topic we will address in the next section.

3.1 On the Observed Variables Likelihood under Missing Data

This section sets the focus on the likelihood of the observed data in the case of missing evidence.

We know in fact that in a network with missing data satisfying *MAR*, it is possible to just focus on such a term forgetting the parameters governing the data hiding mechanism in order to estimate the parameterization θ that maximizes the likelihood of the random variables of interest X_i.

Starting from this principle it holds that for a set of observed variables $\mathcal{D}$ we have:

$$L(\theta : \mathcal{D}) = \prod_m^M P_\theta(d[m])$$

The above looks similar to the case of complete data observations. We might be tempted to say that the learning task does not differ. However, that is not the case, as the subtle difference lies in the fact that under missing evidence we loose the parameter independence property. This because, as we will reason next, in the case of missing evidence, the trails among parameters in the networks are not anymore d-separated such that information on one node will not only yield information for the particular node parameters observation but rather yield information for other local and global network parameters as well.

In order to understand why the decomposition property is gone think for instance at the following basics network structure with table-CPDs and binary random variables: $\mathscr{G}_{X_1 \rightarrow X_2}$. It follows then that you have six parameters governing the random variables realizations: $\theta_{x_1^0}, \theta_{x_1^1}, \theta_{x_2^1|x_1^1}, \theta_{x_2^0|x_1^1}, \theta_{x_2^1|x_1^0}, \theta_{x_2^0|x_1^0}$.

To see why the local decomposition is lost in the above graph consider the case:

$$\theta_{X_2|x_1^1} \rightarrow X_2 \leftarrow \theta_{X_2|x_1^0}$$

It is then straightforward to see that observing both X_2 and X_1, $\theta_{X_2|x_1^0}$ and $\theta_{X_2|x_1^1}$ are d-separated as we can rule out the arcs that are not active. However, when X_1 is missing with X_2 being observed the above will be d-connected due to the common effect factor. In this sense local decomposability is lost and we will have to *sum* up the likelihoods of both the case x_1^1 and x_1^0.

An analogous case emerges for the case of the global decomposition. Think for instance at the network:

$$X_2 \leftarrow H \rightarrow X_1$$

Then in the case of missing H; X_1 and X_2 would be d-connected due to the common factor and no-inactive arcs. It follows once more that the likelihood would be given by the sum of all of the possible likelihood realizations of the missing variable H, such that the likelihood would be given in general by the following expression:

$$L(\theta : \mathcal{D}) = \prod_{m} \sum_{h[m] \in Val(\mathcal{H}[m])} P_\theta(d[m], h[m])$$

It is immediate to see that it is not anymore possible to invert the order of the multiplication due to the interaction of summing and multiplication operations. Moreover, it is also immediate to see that the above will require an inference step to get to the probabilities of the observations.

In this sense both the *local* as well as the *global* likelihood decomposition properties are lost under missing evidence. The computational difficulty of the learning task increases, as it is necessary to deal with multimodal likelihood arising from the sum of unimodal distributions.

We will cover in the next section the idea of the EM-algorithm. This emerged as a powerful algorithm in order to deal with the difficulties that arise from such a complex multinomial likelihood function. We will see that due to an expectation step we will restore an *expected* likelihood decomposability property. Moreover, we will see by reviewing the EM-theory that convergence to local maxima is guaranteed such that we know that such a method will reach one of the local maxima of the likelihood distribution of the observed data.

3.2 The Mathematics of the EM

As discussed by Koller and Friedman (2009) it is possible to frame the EM as a coordinate ascent optimization of an energy function we will define next. Given such perspective we will be able to prove the following theorem

Theorem 1 *When applying the EM-algorithm it holds that the likelihood function increases at each iteration step:*

$$l(\theta^{t+1} : \mathcal{D}) \geq l(\theta^{t} : \mathcal{D}) \qquad \forall\, t$$

In order to prove this, we propose the theory developed in Koller and Friedman (2009). Consider the following energy function:

$$F[P(\mathbf{X}), Q] = E_Q[log(\tilde{P}(\mathbf{X}))] + H_Q(\mathbf{X}) \tag{2}$$

Where $\tilde{P}$ is an unnormalized state probability $P = \frac{\tilde{P}}{Z}$ and H_Q is the entropy of the observed particles.

Using such energy functional 2 it is possible to re-express the logarithm of the normalizing constant Z as follows:

$$log(Z) = F[P, Q] + D(Q||P) \tag{3}$$

where $D(Q||P)$ is the Kullback–Leibler divergence, or relative entropy.

We will choose next the following distribution for the particle distribution:

$$P(\mathbf{H}|\mathbf{D}, \theta) = \frac{P(\mathbf{H}, \mathbf{D}|\theta)}{P(\mathbf{D}|\theta)} \tag{4}$$

With this choice it becomes clear that $Z = P(\mathbf{D}|\theta)$ and $\tilde{P} = P(\mathbf{H}, \mathbf{D}|\theta)$. **D** and **H** represents multivariate random variables and will later represent multivariate random variables composed of observed and missing random variables respectively.

Moreover, recall that using our notation it holds:

$$L(\theta : \mathcal{D}, \mathcal{H}) = P(\mathcal{H}, \mathcal{D}|\theta) \tag{5}$$

$$L(\theta : \mathcal{D}) = P(\mathcal{D}|\theta) \tag{6}$$

where $\mathcal{D}$ represents observed evidence and $\mathcal{H}$ represents missing evidence.

Such that using 3 we can get to the log-likelihood function of the observed data by:

$$l(\theta : \mathcal{D}) = F_D[\theta, Q] + D(Q(\mathcal{H})||P(\mathcal{H}|\theta, \mathcal{D})) \tag{7}$$

$$l(\theta : \mathcal{D}) = E_Q[l(\theta : \mathcal{D}, \mathcal{H})] + H_Q(\mathcal{H}) + D(Q(\mathcal{H})||P(\mathcal{H}|\theta, \mathcal{D})) \tag{8}$$

The above are two fundamental equations. It is in fact straightforward to see that as both the relative entropy as well as the entropy are non-negative the log-likelihood on the left hand side above is an upper bound for the energy functional and the expected log-likelihood relative to Q, for any choice of Q.

Moreover it is straightforward to see in the above that choosing the Q-measure as $P(\mathbf{H}|\mathbf{D}, \theta)$ the relative term fades away such that the entropy term is the overall measure on the difference between the expected log-likelihood and the real log-likelihood. It is in fact clear that in such a case the log-likelihood and the energy functional are the one and the same thing.

In this sense the relation between the energy functional and the log-likelihood is clear and we can think of the EM-algorithm as a coordinate ascent optimization of the energy functional. To see this consider the E-step and M-step as follows.

3.2.1 The Expectation Step

Consider the first coordinate ascent - Q, keeping θ fixed. We look for $\text{argmax}_Q F_D[\theta, Q]$. It is then immediate that:

$$Q^* = P(\mathbf{H}|\mathbf{D}, \theta) \tag{9}$$

$$F_D[\theta, Q^*] = l(\theta : \mathcal{D}) \tag{10}$$

$$F_D[\theta, Q^*] \geq F_D[\theta, Q] \tag{11}$$

The reason because the above is the actual searched maximum argument is the following: You have in general an upper bound on the energy functional given by the log-likelihood. If you now choose the distribution Q in the way described above you know that you have reached the upper bound and that such upper bound is tight. I.e. it is straightforward to see that your are at the maximum for a given θ.

Note that choosing Q^* you are in fact choosing the probability density by which you are going to weight the synthetically created complete data sets in your E-step. In such a way you can interpret the E-step as the step involving the maximization of the energy functional along the Q coordinate.

3.2.2 The Maximization Step

This is the second coordinate ascent - θ. Here we look towards $\text{argmax}_\theta F_D[\theta, Q]$.

It follows then quoting from Koller and Friedman (2009):

"Suppose Q is fixed, because the only term in F that involves θ is $E_Q[l(\theta : \mathcal{D}, \mathcal{H})]$, the maximization is equivalent to maximizing the expected log-likelihood."

This is in fact exactly the standard M-step of the EM algorithm so that we can interpret the M-step as the coordinate ascent along the second axis.

Summarizing, by the fact that at each step we choose Q^* such that 9 holds and by the fact that at each step the energy functional is optimized and increases, it follows from equation 7 that the log-likelihood increases. This immediately proves theorem 1.

4 Bayesian Parameter Learning

A natural question that arises is whether it is possible to generalize the EM-algorithm and apply it also in the case of Bayesian Parameter Learning.

Recall that in Bayesian statistics rather than treating the parameters of interest as fixed but unknown variables you treat them as random variables themselves.

You would then specify a prior, i.e. a probability distribution, as the governing process of the parameters. This can be either a non-informative prior or a prior based on your domain knowledge expertise.

Such prior distribution would then be updated upon the arrival of new observations according to the well known Bayes Rule. The result is an updated posterior distribution from which you can compute your statistics of interest.

$$P(\theta|\mathbf{D}) = \frac{P(\mathbf{D}|\theta) * P(\theta)}{P(\mathbf{D})} \tag{12}$$

It is straightforward to see that that the posterior is proportional to a likelihood term $P(\mathbf{D}|\theta)$ multiplied by the prior distribution.

It follows that depending the choice of point estimate of the posterior a different mathematical exercise is necessary. I.e. in the case of the choice of the first moment as point estimate an integration exercise would be necessary and similar reasonings can be done for the other metrics.

As argued in the introductory chapter, another important point estimator of choice for choosing the parameterization out of the posterior is the most likely point estimate *(MAP)*. This is the point estimate maximizing your posteriori likelihood, i.e. mathematically it is expressed as follows:

$$\begin{aligned}\tilde{\theta} &= \underset{\theta}{\operatorname{argmax}} \frac{P(\mathcal{D}|\theta) * P(\theta)}{P(\mathcal{D})} \\ \tilde{\theta} &= \underset{\theta}{\operatorname{argmax}} P(\mathcal{D}|\theta) * P(\theta) \\ \tilde{\theta} &= \underset{\theta}{\operatorname{argmax}} log(P(\mathcal{D}|\theta)) + log(P(\theta))\end{aligned} \tag{13}$$

$$score_{MAP}(\theta : \mathcal{D}) = log(P(\mathcal{D}|\theta)) + log(P(\theta))$$

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}}\, score_{MAP}(\theta : \mathcal{D}) \tag{14}$$

Where the last equation in 13 follows immediately from the properties of the logarithm function. And the second equation in 13 from the fact that the normalizing constant does not depend on the parameter of interest.

Given the above it is possible to understand that the conclusions from the previous chapter about the EM algorithm apply. The first term of $score_{MAP}$ is exactly the likelihood term of the previous section. The only difference will be in the prior distribution term.

We will show next that it is possible to adjust the M-step of the EM algorithm in order to have a properly working EM algorithm maximizing the score map of 14. This will be the main exercise of the next section.

4.1 Bayesian Parameter Learning - EM Generalization

Maximum a posteriori Bayesian Parameter Learning is a straightforward generalization of the discussion of 3.2.

In fact noting that the score of the MAP estimator is defined as

$$score_{MAP}(\theta : \mathcal{D}) = \ log(P(\mathcal{D}|\theta)) + log(P(\theta)) \tag{15}$$

it is possible to see that the previous results apply.

In order to see that define the following adjusted energy functional:

$$\tilde{F}[\theta, Q] = E_Q[log(\tilde{P}(\mathbf{X}))] + H_Q(\mathbf{X}) + log(P(\theta)) \tag{16}$$

Such that:

$$l(\theta : \mathcal{D}) + log(P(\theta)) = \tilde{F}_D[\theta, Q] + D(Q(\mathcal{H})||P(\mathcal{H}|\theta, \mathcal{D})) \tag{17}$$

It follows immediately that choosing Q as $P(\mathbf{H}|\mathbf{D}, \theta)$ and maximizing the adjusted energy functional we are in fact maximizing the score-map such that the results of the previous section apply.

The only question remaining is on how to optimize the adjusted energy functional via coordinate ascent optimization.

Here it is straightforward to see that the adjusted metric does not affect E-step (we still choose Q in the very same way) but the M-step needs to be reformulated taking the effect of the prior into account.

In order to see this consider our discussion in the previous chapter. The way you choose the Q distribution is unaffected and we will need to perform the same exercise in order to get the $\operatorname{argmax}_Q \tilde{F}_D[\theta, Q]$.

However, what is affected is the optimization along the other coordinate. That is the computation of $\operatorname{argmax}_\theta \tilde{F}_D[\theta, Q]$ keeping Q fixed. In this case the terms depending on θ is not limited to the expected likelihood $E_Q[l(\theta : \mathcal{D}, \mathcal{H})]$, as was the case before, but it is rather important to also consider the prior distribution $P(\theta)$.

5 Hands On - An Exponential Family Example

This section provides an application of theory presented above for the general case of exponential families. The idea is to crystallize the theory developed so far in the general setting of exponential families CPDs.

Given such a procedure it will be possible for the user to apply the presented theory to a general class of distribution allowing a rich modeling possibility for probabilistic graphical models. Moreover, we will leverage such models for implementing the algorithms proposed into the open source merlin engine.

In order to see this define at first the set $\mathscr{Q}$ of parametric distributions belonging to the exponential family $\mathrm{P}_\theta(\mathbf{X})$, defined as:

$$P_\theta(\mathbf{X}) = \frac{1}{Z(\theta)} exp[\sum_i c(\theta_i)\tau(X_i)] * A(\mathbf{X}) \tag{18}$$

where, $Z(\theta)$ is a normalizing term and $\tau(\mathbf{X}) = (\tau(X_1), ..., \tau(X_K))$ is the sufficient statistic.

You can then see that multiple distributions belong to such class of distributions.

Consider for instance the most basic case when modeling Bayesian Networks: the one of multinomial table-CPDs. You can see that such distributions belong to the exponential family.

Recall that for the multinomial table-CPDs with binary random variables X_i the local probability function is given by:

$$P(X_i|\theta) = \prod_{x_i \in Val(X_i), pa_{ij} \in Val(\mathbf{PA}_i)} \theta_{x_i|pa_{ij}} \tag{19}$$

It is now possible to frame the above in the exponential family form by defining the sufficient statistics as $\tau(x_i|pa_{ij}) = \mathbb{1}_{\{X=x, \mathbf{PA}_i = pa_{ij} : x \in Val(X), pa_{ij} \in Val(\mathbf{PA}_i)\}}$ and $c(\theta_{x_i|pa_{ij}}) = ln(\theta_{x_i|pa_{ij}})$.

Given that it is immediate to see that

$$P(X_i|\theta) = exp[\sum_{x_i \in Val(X_i), pa_{ij} \in Val(\mathbf{PA}_i)} c(\theta_{x_i|pa_{ij}}) * \tau(x_i|pa_{ij})] \tag{20}$$

Another of such examples are linear Gaussian Bayesian networks. In such networks each node is described as follows:

$$X_i = \beta_{i0} + \beta_{i1} * pa_{i1} + ... + \beta_{ip} * pa_{ip} + \epsilon \tag{21}$$

where $\epsilon \sim N(0, \sigma^2)$.

Such that, the local probability model is defined as:

$$P(X_i|\theta_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} exp[-\frac{1}{2\sigma_i^2}(x_i - (\beta_{i0} + \beta_{i1} * pa_{i1} + ... + \beta_{ip} * pa_{ip}))^2] \tag{22}$$

You can then see by expanding the square that the sufficient statistics for such local exponential distribution is: $\tau(x_i|pa_i) = (1, x_i, pa_{i1}, ..., pa_{ip}, x^2, x_i pa_{i1}, ..., x_i pa_{ip}, pa_{i1}^2, pa_{i1}pa_{i2}, ..., pa_{ip}^2)$.

Leaving such examples and going back to the general definition of exponential family distributions it is immediate to see that if the local CPDs are exponential family distributions, the global probability function over the entire network will be an exponential family distribution.

5.1 Complete Evidence

Given the definition of local CPD it follows from the theory of the previous section that in the case of *complete data*, we can solve for the global MLE by locally maximizing individual CPDs. You can then get the MLE of the CPDs by either deriving the MLE by standard analytical theory or by means of M-projection theory and moment matching as argued by Koller and Friedman (2009).

This means that in the case of a exponential family distribution with *M* complete instances, you can individually maximize the different:

$$\prod_m P(x_i[m]|\theta_i) = \prod_m^M \frac{1}{Z(\theta_i)} exp[c(\theta_i)^\intercal \tau(x_i[m])] * A(x_i[m]) \tag{23}$$

5.2 Missing Evidence

Consider now the case of *missing evidence*. Here again it is possible to apply the theory exposed in the previous section in a straightforward way. In this sense, we showed how alternating an M-step that maximizes the likelihood of missing and observed values and an E-step that performs some inference for a given parameterization, we are guaranteed to reach a local maxima for the likelihood of the observed data.

In the case of missing evidence, for each instance we might have both observed evidence $d_i[m]$ as well as missing evidence $h_i[m]$.

Given the inference step on the current network parameterization where we compute the probabilistic realization of possible synthetically complete data, we can express and maximize the following expected local likelihood function.

$$
\begin{aligned}
E_Q(l(\theta_i : D_i, H_i)) = & -Mlog(Z(\theta_i) + \sum_m^M \sum_{h_i[m]\in Val(\mathbf{H}_i[m])} Q(h_i[m]) * \mathbf{c}(\theta_\mathbf{i})^\intercal \tau(d_i[m], h_i[m]) \\
& + \sum_m^M \sum_{h_i[m]\in Val(\mathbf{H}_i[m])} Q(h_i[m]) * log(A(d_i[m], h_i[m]))
\end{aligned} \tag{24}
$$

$$
E_Q(l(\theta_i : D_i, H_i)) = -Mlog(Z(\theta_i)) + \sum_m^M E_Q[\mathbf{c}(\theta_\mathbf{i})^\intercal \tau(d_i[m], h_i[m])] + E_Q[log(A(d_i[m], h_i[m]))] \tag{25}
$$

And generally, for the global likelihood involving K factors

$$
\begin{aligned}
E_Q(l(\theta : D, H)) &= \prod_i^K E_Q(l(\theta_i : D_i, H_i)) \\
&= \prod_i^K -Mlog(Z(\theta_i)) + \sum_m^M E_Q[\mathbf{c}(\theta_\mathbf{i})^\intercal \tau(d_i[m], h_i[m])] + E_Q[log(A(d_i[m], h_i[m]))] \\
&= \prod_i^K -Mlog(Z(\theta_i)) + \mathbf{c}(\theta_\mathbf{i})^\intercal \sum_m^M E_Q[\tau(d_i[m], h_i[m])] + E_Q[log(A(d_i[m], h_i[m]))]
\end{aligned} \tag{26}
$$

Hence, it is possible to see that due to the linearity of the expectation we have global decomposability of the expected likelihood function such that we can estimate the global MLE of the expected likelihood of the network by estimating the local MLE of the CPDs expected likelihoods.

Performing this exercise for the two examples above we get the following.

Starting with the multinomial table CPDs and defining a random variable $\mathbf{Y}$ representing the *synthetically completed data* $< \mathbf{H}, \mathbf{D} >$, we have that

$$
\begin{aligned}
\tilde{\theta}_{y_i|pa_{ij}} &= \underset{\theta_{y_i|pa_{ij}}}{\operatorname{argmax}} \prod_m \prod_{y_i\in Val(Y_i)} P_{\theta_{y_i|pa_{ij}}}(y_i[m]|pa_{ij}[m]) \\
\tilde{\theta}_{y_i|pa_{ij}} &= \underset{\theta_{y_i|pa_{ij}}}{\operatorname{argmax}} \sum_m ln(\theta_{y_i|pa_{ij}}) * \sum_{h[m]\in Val(\mathbf{H}[m]):y_i=y_i[m]} Q(h[m]) * \mathbb{1}_{\{y_i=y_i[m], pa_{ij}=pa_i[m]\}}
\end{aligned} \tag{27}
$$

With the additional constraints that $\sum_{y_i\in Val(Y_i), pa_{ij}\in Val(\mathbf{PA}_i)} \theta_{y_i|pa_{ij}} = 1$.

Solving this constrained optimization problem by standard Lagrange method you get:

$$
\tilde{\theta}_{y_i|pa_{ij}} = \frac{\bar{M}[y_i, pa_{ij}]}{\sum_j \bar{M}[y_j, pa_{ij}]} \tag{28}
$$

With $\bar{M}[y_i, pa_{ij}] = \sum_m^M \sum_{h[m]\in Val(\mathbf{H}[m]):y_i=y_i[m]} Q(h[m]) * \mathbb{1}_{\{y_i=y_i[m], pa_{ij}=pa_{ij}[m]\}} = E_Q(M(y, pa)),\ M(y, pa) = \sum_m \tau(y, pa)$.

Algorithmically it is then possible to write the EM-application for the above case as in 1.

Algorithm 1 EM-Learning: the classical EM algorithm for learning with missing evidence

```
Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset $S$
procedure EM($\mathcal{B}$, $S$)
    Initialize $\mathcal{B}$'s parameters $\theta \leftarrow \theta^0$
    for all $t = 1, \ldots$ until convergence do
        $\left\{ \bar{M}_{\theta^t}[x_i, u_i] \right\}$ ←COMPUTE-ESS($\mathcal{B} = (G, \theta^t)$, $S$)
        for all $i = 1, \ldots, n$ do
            for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
                $\theta^{t+1}_{x_i|u_i} = \frac{\bar{M}_{\theta^t}[x_i, u_i]}{\bar{M}_{\theta^t}[u]}$
            end for
        end for
    end for
end procedure

function COMPUTE-ESS($\mathcal{B} = (G, \theta)$, $S$)
    for all $i \in 1, \ldots, n$ do
        for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
            $\bar{M}[x_i, u_i] \leftarrow 0$
        end for
    end for
    for all example $S_j \in S$ do
        Run inference on $(G, \theta)$ with evidence $d_j$
        for all i= $1, \ldots, n$ do
            for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
                $\bar{M}[x_i, u_i]$ += $P_{(G,\theta)}(x_i, u_i | d_j)$
            end for
        end for
    end for
end function
```

Turning to the second example, the one of linear Gaussian CPDs we have for the local CPD

$$P(X|\theta) = \prod_{m} \prod_{y_i \in Val(Y_i), pa_i \in Val(\mathbf{PA}_i)} \prod_{h[m] \in Val(\mathbf{H}[m])} \frac{1}{\sqrt{2\pi\sigma^2}} exp[-\frac{1}{2\sigma^2}(Q(h[m]) * y[m] - (\beta_0 + \beta_1 * pa_1[m] + ... + \beta_K * pa_K[m]))^2] \tag{29}$$

such that once more we have an exponential family, which likelihood we aim to optimize.

In order to perform such a task we refer to the M-projection theory. As proved by Koller and Friedman (2009), the M-projection of an arbitrary distribution on the exponential family is given by the parameterization where the expected sufficient statistics of the two distributions match.

Moreover, given the fact that it is possible to prove that the MLE of an exponential family is nothing else than the M-projection of the empirical distribution on the exponential distribution of interest, it follows immediately that we can find the MLE parameterization by finding the M-projection through moment-matching.

In the specific to solve such MLE problem we need to find the parameterization such that the empirical average of the sufficient statistics corresponds to the one of the expected sufficient statistics given the exponential family parameterization.

Given the above results from information theory it is generally possible to compute the MLE of exponential families in the presence of missing data by firstly computing a map

$$ess(\theta) = E_{P_\theta}(E_Q(\tau(\mathbf{Y})))$$

Then, if possible, inverting such map

$$\theta = ess^{-1}$$

and finally inserting the empirical moments of the expected sufficient statistics.

Note that due to the synthetically completed dataset you work with the expected - expected sufficient statistics. Where the double expectation has to account on the one hand the expectation of the synthetically completed evidence and, on the other hand, the moment matching expectation given the exponential family parameterization from the M-projection theory.

Doing the above exercise for a simple linear Gaussian CPD with a single parent would result in the following picture:

$$\begin{aligned} ess(\theta) &= ess\begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix} \\ &= \begin{pmatrix} E_{P_\theta}(E_Q(\mathbf{Y})) = \beta_0 + \beta_1 E_{P_\theta}(\mathbf{PA}_1) \\ E_{P_\theta}(E_Q(\mathbf{Y} * \mathbf{PA}_1)) = \beta_0 E_{P_\theta}(\mathbf{PA}_1) + \beta_1 E_{P_\theta}(\mathbf{PA}_1^2) \end{pmatrix} \end{aligned}$$

Such that inverting such a map and inserting the empirical moments we get

$$\hat{\theta} = \begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix} = \begin{pmatrix} E_D(E_Q(\mathbf{Y})) - \frac{E_D(E_Q(\mathbf{Y}*\mathbf{PA}_1)) - E_D(E_Q(\mathbf{Y}))E_D(\mathbf{PA}_1)}{E_D(\mathbf{PA}_1^2) - E_D(\mathbf{PA}_1)^2} * E_D(\mathbf{PA}_1) \\ \frac{E_D(E_Q(\mathbf{Y}*\mathbf{PA}_1)) - E_D(E_Q(\mathbf{Y}))E_D(\mathbf{PA}_1)}{E_D(\mathbf{PA}_1^2) - E_D(\mathbf{PA}_1)^2} \end{pmatrix} \tag{30}$$

where the empirical moments are given by $E_D(E_Q(\mathbf{Y})) = \frac{1}{M}\sum_m \sum_{h[m] \in Val(\mathbf{H}[m])} Q(h[m])y[m]$ and similar.

It is now clear that such an approach can be used in the general case of exponential families. You can for instance easily get to the MLE result of the multinomial case achieved via Lagrange method through the moment matching idea presented above.

In general the methodical frame for exponential families CPDs is the following; you substitute the inference step in line 27 of Algorithm 1 with an inference step calculating the expected sufficient statistics *of interest* given the exponential family distribution of choice. You then insert in the M-step of line 6-9, the M-projection parameterization obtained by the moment-matching of expected sufficient statistics as discussed above. Finally you iterate until convergence.

5.3 Bayesian Parameter Learning - A CPT example

An example for the extension of the EM algorithm to compute the maximum a posteriori parameter in the case of missing evidence is treated in this section.

The theory proceeds with the most classic network structure. The one of table conditional probability distributions where the realizations are distributed according to a multinomial distribution given the $\theta_{\mathrm{X_i}\,|\,\mathbf{PA}_{\mathrm{X_i}}}$ local parameters and where possible realizations are binary, $Val(X_i) = \{0, 1\}$.

Specifying a Dirichlet distribution as the prior of such parameters we can compute the maximum a posteriori estimator.

As from the reasoning of the previous chapter we know that the EM algorithm properties of convergence and correctness apply and that the algorithm will iteratively converge to a local maximum.

While as mentioned the E-step will be unaffected by the introduction of the prior, we need to adapt the M-step to account for the influence of the latter.

Consider in this sense the unnormalized probability for the Dirichlet-Multinomial posterior distribution:

$$P(\theta|X) = \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_i^K \theta^{x_i}_{x_i|\mathbf{PA}_i} * \frac{1}{B(\alpha)} \prod_{i=1}^K \theta^{\alpha_i - 1}_{x_i|\mathbf{PA}_i} \tag{31}$$

And consider the adjusted energy functional 16. We can derive the new likelihood expression in the case of missing evidence by defining a new multivariate random variable $\mathbf{Y}$ expressing synthetically completed data observations $< \mathbf{H}, \mathbf{D} >$:

$$\tilde{F}[\theta, Q] = E_Q[P_\theta(\mathbf{Y})] + H_Q(\mathbf{Y}) + log(P_{hyperparameters}(\theta)) \tag{32}$$

Such that taking the argument maximizing the likelihood of the adjusted energy functional $\operatorname{argmax}_\theta \tilde{F}[\theta, Q]$ we are left with the following, where y[m] represents synthetically created complete observation <h[m], d[m]>:

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_m E_Q[log(\prod_i^K \prod_{pa_{ij} \in Val(\mathbf{PA}_i)} \frac{\Gamma(\sum_i [y[m]_i, pa[m]_{ij}] + 1)}{\prod_i \Gamma([y[m]_i, pa[m]_{ij}] + 1)} \theta^{[y[m]_i, pa[m]_{ij}]}_{y_i|pa_{ij}} * \frac{1}{B(\alpha)} \theta^{\alpha_i - 1}_{y_i|pa_{ij}})] + H_Q(y[m]) \tag{33}$$

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_m E_Q[log(\prod_i^K \prod_{pa_{ij} \in Val(\mathbf{PA}_i)} \theta^{[y[m]_i, pa[m]_{ij}]}_{y_i|pa_{ij}} * \theta^{\alpha_i - 1}_{y_i|pa_{ij}})] \tag{34}$$

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_m E_Q[log(\prod_i^K \prod_{pa_{ij} \in Val(\mathbf{PA}_i)} \theta^{[y[m]_i, pa[m]_{ij}] + \alpha_i - 1}_{y_i|pa_{ij}})] \tag{35}$$

It follows given that by the linearity of the expectation and that $y[m]_i = \{0, 1\}$, we can re-express the above as:

$$\tilde{\theta} = \underset{\theta}{\text{argmax}} \sum_{i}^{K} \sum_{pa_{ij} \in Val(\mathbf{PA}_i)} (\sum_{m}^{M} E_Q[M[y_i, pa_{ij}]] + \alpha_i - 1) * log(\theta_{y_i|pa_{ij}})] \tag{36}$$

where it holds

$$\bar{M}[y_i, pa_{ij}] = \sum_{m}^{M} E_Q[M[y_i, pa_{ij}]] \tag{37}$$

$$\bar{M}[y_i, pa_{ij}] = \sum_{m}^{M} \sum_{h[m] \in Val(\mathbf{H}[m]) : y_i = y_i[m]} Q(h[m]) \mathbb{1}_{\{y_i = y[m]_i, pa_{ij} = pa[m]_{ij}\}} \tag{38}$$

$$\bar{M}[y_i, pa_{ij}] = \sum_{m}^{M} P(y_i, pa_{ij} | d[m], \theta) \tag{39}$$

So that ultimately:

$$\tilde{\theta} = \underset{\theta}{\text{argmax}} \sum_{i}^{K} \sum_{pa_{ij} \in Val(\mathbf{PA}_i)} (\bar{M}[y_i, pa_{ij}] + \alpha_i - 1) * log(\theta_{y_i|pa_{ij}})] \tag{40}$$

Given the additional restriction that $\sum_i \sum_{pa_{ij} \in Val(\mathbf{PA}_i)} \theta_{y_i|pa_{ij}} = 1$, we can obtain the necessary condition for finding the optimum by using the Lagrange method

$$\frac{\partial}{\partial \theta_{y_i|pa_{ij}}} \sum_{i}^{K} (\bar{M}[y_i, pa_{ij}] + \alpha_i - 1) * log(\tilde{\theta}_{y_i|pa_{ij}})] - \lambda(\sum_{i} \tilde{\theta}_{y_i|pa_{ij}} - 1) \overset{!}{=} 0 \tag{41}$$

$$\lambda = \frac{\bar{M}[y_i, pa_{ij}] + \alpha_i - 1}{\tilde{\theta}_{y_i|pa_{ij}}} \tag{42}$$

And inserting this in the first order condition and solving for $\tilde{\theta}_{y_i|pa_{ij}}$

$$\tilde{\theta}_{y_i|pa_{ij}} = \frac{\bar{M}[y_i, pa_{ij}] + \alpha_i - 1}{\sum_j (\bar{M}[y_j, pa_{ij}] + \alpha_j - 1)} \tag{43}$$

This will be the way you update the parameters in the M-step.

It is straightforward to see from the above that it is possible to perform the same exercise in similar settings possibly leveraging the M-projection theory in the case of exponential family posterior distributions.

We conclude by noting that as long as the prior distribution $P(\theta)$ is well behaved in the sense that the resulting posterior (i) is concave (ii) is differentiable (iii) is smooth such that it is possible to exchange differentiation and integration; then the MAP estimator will exists, the correctness and convergence properties of EM apply to the score of the maximum a posteriori point estimate. A local maximum point estimator for the likelihood of the observed data will ultimately result.

5.4 Bayesian Parameter Learning - An Exponential Family Generalization

This section generalizes the exercise of the above section for general exponential family distributions. As discussed in Barndorff-Nielsen (1978), Geiger et al. (1998), Lauritzen (1996) and as well known from standard statistical theory such distributions are particularly well suited for statistical analysis due to their properties.

Albeit the only restriction for the choice of the prior distribution are the one mentioned at the end of the previous section a particularly sensible selection for the prior distribution is the one of using conjugate priors as defined by Schlaifer and Raiffa (1961). This because, when using conjugate priors the data is incorporated into the posterior distribution only through the sufficient statistics such that there will exist relatively simple formulas for updating the prior into the posterior Fink (1997).

Moreover, through such a property it will be easy to compute the MLE according to the sufficient statistics in the *complete data* case, or according to the expected sufficient statistics in the case of *missing data* evidence. Finally, the fact that conjugate priors of exponential family distributions will often be well known exponential family distributions will further help in the parameter estimation given that the maxima of such posterior distributions are well documented in many statistical textbooks.

In fact, it is possible to note that the Dirichlet prior chosen in the previous section is nothing else then the conjugate prior to the multinomial distribution. However, note that the resulting posterior is not an exponential distribution such that you cannot apply the M-projection theory to get the result above.

Turning to the linear Gaussian parametric model presented in this script it is possible to see that the conditional distribution of local nodes in the network arises by a multivariate normal distribution of the parents, see for instance Koller and Friedman (2009).

It follows therefore that one way for performing Bayesian parameter learning in linear Gaussian Bayesian networks is by specifying a normal-inverse Wishart prior distribution on the multivariate mean and co-variance matrix of the local nodes of the parents.

After obtaining the new posterior hyperparameters depending on the prior hyperparameters and the (expected) sufficient statistics in the case of complete (missing) data, it is possible to obtain the maximum by getting the mode of the resulting multivariate t-distribution parameterized according to the hyperparameters and expected sufficient statistics.

Similar reasonings are possible by specifying accordingly the CPDs and prior distributions, such that a rich modeling set is available and easily implementable in statistical software.

6 On Numerical EM

As argued in the previous section when working with conjugate prior we might easily get to closed form solutions for the maximum of the posterior.

However, as was previously discussed it might be limiting to restrict the prior specification to conjugate priors of exponential distributions.

To tackle this issue and address the possibility of using a richer class of likelihood-priors instantiations we propose in this section some arguments for iteratively computing the maximum of an arbitrary well behaved distribution as discussed in section 5.3.

One classical tool to perform the task is the one of leveraging stochastic simulation methods to sample from the posterior distribution of interest and leverage asymptotic theory to get to the statistics of interest.

Another option, which we will focus next, is to apply a numerical solution to the M-step of the EM algorithm leveraging the theory presented in Ruud (1989).

Hence, in this section, we will aim to generalize the theory presented that far such that it is possible to implement general statistical software without having to limit the end-user to very specific pre-defined cases, where the algorithm running in the background has necessarily to know the closed-form analytical solution of the M-step.

Note that this will come at costs. We will need in fact to compute the Hessian of our expected log-likelihood which is one of the most computationally intensive tasks. This especially in highly dimensional problems. One of the major benefits of using the EM over gradient based methods would be lost in this sense.

6.1 Numerical EM for MLE estimator

In order to understand how to compute the M-step according to an iterative method, think at the following.

Consider that in the E-step you set $Q = P(\mathbf{H}|\mathbf{D}, \theta_0)$, such that you can reformulate 7 as follows

$$l(\theta : \mathcal{D}) = H_Q(\mathcal{H}) + \sum_{h \in Val(\mathcal{H})} P(h|\mathcal{D}, \theta_0) * l(\theta : \mathcal{D}, h) \tag{44}$$

$$Q(\theta, \theta_0 : \mathcal{D}) =: \sum_{h \in Val(\mathcal{H})} P(h|\mathcal{D}, \theta_0) * l(\theta : \mathcal{D}, h) \tag{45}$$

$$H(\theta_0, \theta : \mathcal{D}) =: Q(\theta, \theta_0 : \mathcal{D}) - l(\theta : \mathcal{D}) \tag{46}$$

It follows

$$
\begin{aligned}
\frac{\partial}{\partial\theta} l(\theta:\mathcal{D}) = l_1(\theta:\mathcal{D}) = \frac{\partial}{\partial\theta} Q(\theta,\theta_0,\mathcal{D}) - \frac{\partial}{\partial\theta} H(\theta,\theta_0,\mathcal{D}) \\
= Q_1(\theta,\theta_0:\mathcal{D}) - H_1(\theta,\theta_0:\mathcal{D}) \qquad (47)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2}{\partial\theta\partial\theta'} l(\theta:\mathcal{D}) = \frac{\partial^2}{\partial\theta\partial\theta'} Q(\theta,\theta_0,\mathcal{D}) - \frac{\partial^2}{\partial\theta\partial\theta'} H(\theta,\theta_0,\mathcal{D}) \\
= Q_{11}(\theta,\theta_0:\mathcal{D}) - H_{11}(\theta,\theta_0:\mathcal{D}) \qquad (48)
\end{aligned}
$$

Moreover given the following condition

$$
H_1(\theta_0,\theta_0:\mathcal{D}) = 0 \qquad \forall\theta_0 \qquad (49)
$$

we have for 47 that:

$$
l_1(\theta_0:\mathcal{D}) = Q_1(\theta_0,\theta_0:\mathcal{D}) \qquad \forall\theta_0 \qquad (50)
$$

Such that ultimately it holds using the classical derivation of the Newton-Raphson Method as in Storvik (2007):

$$
\theta_{EM} = \theta_0 - Q_{11}^{-1} Q_1 + o(||\theta_{EM} - \theta_0||) \qquad (51)
$$

where both Q_{11}, Q_1 are evaluated at θ_0.

It follows immediately that for log-concave functions each iteration of 51 increases the likelihood. It is therefore possible to apply the above numerical M-step by inserting the numerical computed Hessian and gradient. The convergence of the EM algorithm will not be hampered.

It is as well possible to set a predefined amount of iterations before switching to the next E-step in the EM-algorithm. Due to the increased computational cost of performing new inferences as well as computing new Hessian matrices such second option is not recommended; albeit being theoretically viable.

As a final remark, note that methods to improve the computational speed of such numerical M-step have been proposed, such in Louis (1982). As uphill steps cannot be guaranteed under all circumstances in such algorithm, we just refer the interested reader to the literature and do not consider this as a viable option for our solution. In that case the EM theory would collapse and there is no guarantee to reach a local maximum.

6.2 Numerical EM for MAP estimator

This section generalizes the arguments of the previous section to the case of MAP estimator in the case of Bayesian Parameter Learning.

Using 16 it follows immediately using the notation of the last section that:

$$l(\theta : \mathcal{D}) + log(P(\theta)) = H_Q(\mathcal{H}) + log(P(\theta)) + \sum_{h \in Val(\mathcal{H})} P(h|\mathcal{D}, \theta_0) * l(\theta : \mathcal{D}, h) \tag{52}$$

$$Q(\theta, \theta_0 : \mathcal{D}) =: log(P(\theta)) + \sum_{h \in Val(\mathcal{H})} P(h|\mathcal{D}, \theta_0) * l(\theta : \mathcal{D}, h) \tag{53}$$

$$H(\theta_0, \theta : \mathcal{D}) =: Q(\theta, \theta_0 : \mathcal{D}) - l(\theta : \mathcal{D}) \tag{54}$$

The idea is that as long as the likelihood and the prior are concave such that the sum of two concave functions will yield a Q function that is concave, we might apply the very same Newton-Raphson method to get iteratively to the maximum of the function.

$$\theta_{EM} = \theta_0 - Q_{11}^{-1} Q_1 + o(||\theta_{EM} - \theta_0||) \tag{55}$$

where Q_{11}, Q_1 are defined as in the previous section with the difference that they now account for the prior distribution influence.

6.3 On the numerical computation of the derivatives of expected log-likelihoods

This section concludes the chapter on Bayesian Parameter Learning by substituting the M-step of 4, by a numerical estimation of the maximum.

Note, that as argued in the previous sections this has the benefit of allowing a general algorithm that is not bounded to the analytical derivation of the maximum in the M-step.

Algorithm 2 Replace M-step for Bayesian Parameter Learning

```
Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset $S$, Current Parameterization $\theta_0$, Threshold $\epsilon$
function M-STEP($\mathcal{B}$, $S$)
    Numerically Compute $Q_1$
    Numerically Compute $Q_{11}$

    for all $t = 0, \dots$ until convergence do

        $\theta^{t+1} = \theta_t - Q_{11}^{-1} Q_1$

        convergence if $||\theta^{t+1} - \theta_t|| < \epsilon$
```

We keep the above formulation in general terms but we note that a sensible choice for computing the Hessian and Gradient of the expected log-likelihood function are the methods proposed in Meng (2016), where the author proposes several computational efficient methods for estimating the Fisher Information matrix in the case of an EM-algorithm application.

7 On Likelihood Evidence

Recall that as defined in Mrad et al. (2015), in the case of likelihood evidence an observation is uncertain due to unreliable source of information.

Then the evidence of a finding is expressed as a vector containing the relative likelihood of a random variable realization. Consider for instance a random variable **X** then its likelihood evidence is defined as:

$$L(X) = (L(X = x_1) : ... : L(X = x_k)) \tag{56}$$

Or when normalized you can express the likelihood-evidence as

$$L(X) = (P(obs|x_1) : ... : P(obs|x_k)) \tag{57}$$

Note that here the relative likelihoods do not have to sum to one. Thus they cannot be interpreted as probabilities.

Moreover, the key take-away that distinguish likelihood evidence from probabilistic evidence is, as mentioned, the fact that a likelihood evidence vector as in 56 is specified without a prior. This means that the prior encoding the probabilistic structure of the network for the local random variable realization is not taken into account. I.e. the information resulting from $P(X_i|\mathbf{PA}_{X_i})$ is not considered when expressing such an evidence.

This means, that when updating the belief on the realization of the random variable $\mathbf{X_i}$, i.e. at inference time, the likelihood evidence provided by the unreliable source of information must be combined with the prior probability resulting from the probabilistic structure implied by the network.

We will turn next to the task of performing such inference and the task of parameter learning under likelihood evidence describing the approach proposed in Wasserkrug et al. (2021).

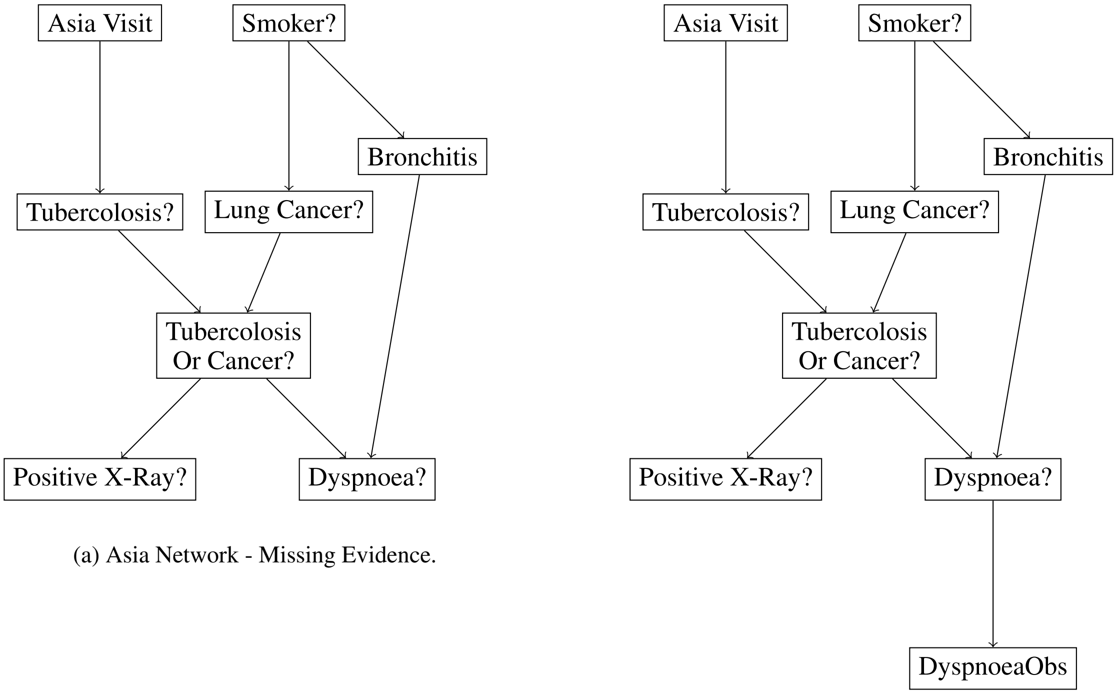
7.1 Adjusted EM - Likelihood Evidence

One of the most widespread ways to deal with likelihood evidence was introduced by Pearl (2014). The idea is to remodel the network structure $G(\mathscr{V}, \mathscr{X})$ in order to represent the likelihood evidence as a hard-finding on a newly created *virtual-node*.

Consider the Asia Network of Figure 2, as in Wasserkrug et al. (2021), Mrad et al. (2015). On the left hand side the core network is presented. Given hard findings or missing evidence we can estimate the parameters of the network via the standard EM-algorithm.

Consider now the right hand side of Figure 2. Assume, as in Wasserkrug et al. (2021) that likelihood evidence is obtained for the Dysponea node via an NLP tool [NLP] analyzing historical medical records. Then as proposed by Pearl (2014) we augment the network as on the right hand side of Figure 2 by creating a child node of the Dysponea node. Such a child node will encode the likelihood evidence as hard finding by specifying the relation between Dysponea and Dysponea Observed of interest; i.e. it will encode the likelihood evidence via the CPD of $P(DysponeaObs|Dysponea)$.

Figure 2: Asia Network - Virtual Evidence Comparison



(a) Asia Network - Missing Evidence.

(b) Asia Network - Expanded as by Pearl's Virtual Evidence.

Concretely assume as in Wasserkrug et al. (2021) that the NLP correctly characterizes Dysponea 70% of the times, when this does in fact occurs. Note that the NLP tool does not consider any prior information resulting from the probabilistic structure of our network. Then you might encode such likelihood evidence of the NLP as in Table 1.

Dysponea? DysponeaObs	yes	no
True	0.7	0.3
False	0.3	0.7

Table 1: DysponeaObs - Virtual Evidence Node CPT

Given such a CPT, encoding the likelihood evidence, it is possible to set the DyspnoeaObs to true as a hard finding. In such a way you will work with a standard network that is just composed of missing and hard evidence. You can then update the cognitive state of your network by standard inference techniques, and compute the parameters of interest by a standard EM-algorithm.

Given such explanation it follows that it is possible to rewrite the EM-step by adjusting the E-step such that it will perform its inference step on the virtual evidence augmented network that respects and incorporates the likelihood evidence information. This was the intuition and contribution of Wasserkrug et al. (2021) and such an algorithm, with the corresponding modification of the E-step, is presented in 3.

We continue the next section by modifying such algorithm such that it is possible to perform MAP estimation in Bayesian settings.

2**Algorithm 3** EM-Likelihood: an EM algorithm for learning with likelihood evidence

```
Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset $S$
procedure EM($\mathcal{B}$, $S$)
    Initialize $\mathcal{B}$'s parameters $\theta \leftarrow \theta^0$
    for all $t = 1, \ldots$ until convergence do
        $M - step\ as\ in\ Algorithm\ 1$
    end for
end procedure

function COMPUTE-ESS($\mathcal{B} = (G, \theta)$, $S$)
    for all $i \in 1, \ldots, n$ do
        for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
            $\bar{M}[x_i, u_i] \leftarrow 0$
        end for
    end for
    for all example $S_j \in S$ do
        Let $O_j$ be the observations induced by $S_j$
        $(G', \theta') \leftarrow$ AUGMENT-BN($\mathcal{B} = (G, \theta)$, $O_j$)
        for all $o \in O_j$ do
            Set the value of $o_V$ to $true$
        end for
        Run inference on $(G', \theta')$ with evidence $d_j$
        for all i= $1, \ldots, n$ do
            for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
                $\bar{M}[x_i, u_i] \mathrel{+}= P_{(G',\theta')}(x_i, u_i | d_j)$
            end for
        end for
    end for
end function

function AUGMENT-BN($\mathcal{B} = (G, \theta)$, $O$)
    Initialize $G' \leftarrow G$, $\theta' \leftarrow \theta$
    for all $o \in O$ do
        $G'_{\mathbb{V}} \leftarrow G'_{\mathbb{V}} \cup o_V$, $G'_{\mathbb{E}} \leftarrow G'_{\mathbb{E}} \cup (V, o_V)$  ▷ Add a new observation node to the graph and connect it to the relevant node
        for all $c_i \in Conf$ do                                  ▷ $Conf$ actual likelihood values provided for a node
            $\theta' \leftarrow \theta' \cup \theta_{O_V = true | v_i} = c_i$      ▷ Set the relevant CPT entry to be $Pr(obs | V = v_i)$
        end for
    end for
    return $(G', \theta')$
```

7.2 Bayesian Learning MAP - Adjusted EM for Likelihood Evidence

The idea of this section is to extend 3 in order to obtain the MAP estimator in a Bayesian Learning setting with Likelihood Evidence.

We discussed in the previous section how likelihood evidence requires augmenting the core network by virtual evidence nodes as in Pearl (2014) and consequently perform the inference step on such augmented networks.

Such procedure was outlined by the modification of the E-step in comparison to the standard EM algorithm with missing evidence.

Moreover, we discussed in section 4, how it is possible to adjust the M-step of the EM-algorithm to perform the task of MAP estimation. Both correctness and convergence properties will apply such that we will converge to a local maximum for our posterior distribution.

Combining the two steps it is immediate to see that it is possible to perform Bayesian Parameter Learning under likelihood evidence by replacing line 4 of 3 with

2**Algorithm 4** Replace M-step for Bayesian Parameter Learning

```
Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset $S$
function M-STEP($\mathcal{B}$, $S$)
    $\theta^{t+1}_{x_i|u_i} = \frac{\bar{M}_{\theta^t}[x_i,u_i]+\alpha_i-1}{\sum_j \bar{M}_{\theta^t}[x_j,u_i]+\alpha_j-1}$
return $(\theta^{t+1})$
```

Given such a computation it is possible to get to a local maximum for the MAP estimator.

8 On Probabilistic Evidence

In this section we extend the theory presented that far by introducing some techniques in order to deal with parameter learning for the case of *non-fixed probabilistic evidence*.

In the previous chapter we showed how it is possible to rephrase a likelihood evidence as a hard evidence by means of augmenting the network via *virtual evidence*.

We could then perform the inference step and propagate the information by means of Bayes Rule updating the probabilistic structure of the network.

By contrast, with probabilistic evidence such an approach is not viable. This because, as argued by PENG et al. (2010), propagating a probabilistic finding on $X_i \in \mathbf{X}$ requires a revision of the probability distribution of the network $P_\theta(\mathbf{X})$ on X_i by a local probability distribution defined by the probabilistic evidence statement $R(X_i)$. Given that $R(X_i)$, although acting as a condition for the update, is not itself an event, Bayes Rule and standard inference based on message passing algorithms fail. Hence, as mentioned in Mrad et al. (2015), a probabilistic finding $R(X_i)$ requires a reconsideration of the entire joint probability distribution $P_\theta(\mathbf{X})$ because it replaces the existing *prior* on the variable X_i.

In simple words, in the presence of probabilistic evidence it is not possible to propagate evidence by standard message passing algorithms. The solution proposed by Jeffrey (1990), is then to replace the initial probabilistic structure of the network $P_\theta(\mathbf{X})$ by a new probabilistic structure $Q_\theta(\mathbf{X})$ that reflects the beliefs in the variables of the model *after accepting the probabilistic evidence*.

In the specifics, as well outlined by Mrad et al. (2015), according to what is usually referred as Jeffrey's *probability kinematics*, Q must satisfy the following requirements:

1. the posterior probability distribution considering the network structure on the observed variable $Q(X_i)$ is unchanged: $Q(X_i) = R(X_i)$. This is in fact the functional requirement of the probabilistic evidence.
2. the conditional probability distribution of other variables given X_i remains invariant under the observation: $Q(\mathbf{X} \backslash X_i | X_i) = P(\mathbf{X} \backslash X_i | X_i)$. This essentially means that even if P and Q disagree on X_i, they agree on the consequences of X_i on other variables Mrad et al. (2015).

With the above specification of a new probabilistic structure satisfying the functional requirements of probabilistic evidence it is possible to compute the probability of a given event by means of Jeffrey's rule:

$$Q(\mathbf{Z} = z) = \sum_{x_i \in Val(X_i)} P(\mathbf{Z} = z | X = x_i) R(X = x_i) \tag{58}$$

Albeit being theoretically compelling, Jeffrey's formula above 58, should not be directly applied in Bayesian Networks. In fact such a formula requires the specification and functional form of the full probabilistic structure of the network in any state of the network in order to compute $P(\mathbf{Z} = z | X_i = x_i)$. I.e. in order to compute the new probabilistic structure Q$_\theta$ you would need to perform an inference step over all possible states combinations. A very computationally intensive task.

The solution to this problem as suggested by Chan and Darwiche (2005) and PENG et al. (2010) is to frame probabilistic evidence into likelihood evidence by computing the likelihood ratio as defined by:

$$L(X_i) = (\frac{R(x_{i_1})}{P(x_{i_1})} : ... : \frac{R(x_{i_k})}{P(x_{i_k})}) \tag{59}$$

It is then possible to prove that propagating such likelihood evidence by means of Pearl's method as described in the previous section, is equivalent to propagating and obtaining the probabilistic structure by means of Jeffrey's method 58.

It is in fact possible to prove, as in PENG et al. (2010), that with such an approach the posterior probability of X_i after propagating $L(X_i)$ by Pearls method, is equal to $R(X_i)$.

Given such theory it is straightforward to understand that in the case of a single probabilistic evidence we can easily learn the parameters of the Bayesian Network via the following adjustment of the **AUGMENT-BN** function of 3

Algorithm 5 EM-Single Probabilistic Evidence: an EM algorithm for learning in the case of a single probabilistic evidence

```
Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset $S$, Observations $O$
function AUGMENT-BN($\mathcal{B} = (G, \theta), O$)
    Initialize $G' \leftarrow G, \theta' \leftarrow \theta, Conf \leftarrow \emptyset$
    for all $r_{j_i} \in ProbEv(x_j)$ do        ▷ $ProbEv$ is the passed probabilistic evidence. r are the states for the Node.
        $Conf \leftarrow Conf \cup \frac{r_{j_i}}{\mathbf{P}_{x_{j_i}}}$
    end for
    for all $o \in O$ do
        $G'_{\mathbb{V}} \leftarrow G'_{\mathbb{V}} \cup o_V, G'_{\mathbb{E}} \leftarrow G'_{\mathbb{E}} \cup (V, o_V)$        ▷ Add a new observation node to the graph and connect it to the relevant node
        for all $c_i \in Conf$ do        ▷ $Conf$ computed likelihood for a probabilistic node
            $\theta' \leftarrow \theta' \cup \theta_{O_V=true|v_i} = c_i$        ▷ Set the relevant CPT entry to be $Pr(obs|V = v_i)$
        end for
    end for
    return $(G', \theta')$
```

It is important to mention, to this point that in the case of multiple probabilistic evidence on different nodes, the above approach does not apply.

This because, as shown by example in PENG et al. (2010), the algorithm above is not commutative and does not guarantee the functional requirement $R(X_i) = Q(X_i)$ for at least one node i. This intuitively because it just guarantees the functional property for the last virtual node for which inference is propagated using Pearl virtual evidence method.

This is for instance what happens in PENG et al. (2010) with two probabilistic evidence, $R(X_1), R(X_2)$ and the property that $Q(X_1) \neq R(X_1)$ or $Q(X_2) \neq R(X_2)$, depending on the order of propagation.

In order to solve such an issue, and guarantee the functional requirement of probabilistic evidence, the *Iterative Proportional Fitting Procedure (IPFP) algorithm* was proposed by Valtorta et al. (2002).

The algorithm is based on the following theorem as reported in PENG et al. (2010):

Theorem 2 *Let $Q_i(\boldsymbol{X})$ be the distribution resulting from updating $P(\boldsymbol{X})$ by $R(X_i)$, $X_i \subset \boldsymbol{X}$ using Jeffreys rule described above. Then $Q_i(\boldsymbol{X})$ is the I-projection of $P(\boldsymbol{X})$ on $\boldsymbol{P}_{R(X_i)}$, where $\boldsymbol{P}_{R(X_i)}$ is the set of distributions whose marginal over X_i equal $R(X_i)$.*

The idea of the IPFP algorithm is then the one of allowing multiple probabilistic evidence by leveraging the theorem above.

As well outlined in PENG et al. (2010), the idea is in fact to modify $P(\mathbf{X})$ incorporating the multiple constraints arising from the multiple probabilistic evidence conditions passed by the user. Consider $j = 1, ..., J$ restrictions, then you can perform such an exercise by iteratively projecting the distribution resulted from the previous iteration $\mathbf{P}_{R(X_j)}$ on the next set of constraints $R(X_{j+1})$.

Formally the IPFP would look as follows:

Algorithm 6 IPFP Algorithm

```
Require: Probabilistic Evidence $R(X_j, ..., X_J)$, intial distribution $P(\mathbf{X})$, necessary condition $R(X_j) << Q_{k-1}(X_j) \ \forall \ k$
function IPFP-DISTRIBUTION($Q_k(\mathbf{X})$)
    Initialize $Q_0(\mathbf{X}) \leftarrow P(\mathbf{X})$
    for $k = 1, ..., m = 1 + (k-1) \ mod \ J$ do
        $Q_k(x) = \begin{cases} Q_{k-1}(x) * \frac{R(x_j)}{Q_{k-1}(x_j)} & Q_{k-1}(x_j) \geq 0 \\ 0 & else \end{cases}$    ▷ note that $j$ represent the constraint used at each iteration
    end for
```

As proved by several authors such method converge. Moreover, given the new probabilistic structure $Q_k(\mathbf{X})$ that reflects the beliefs in the variables of the model *after accepting the probabilistic evidence*, we can learn the parameters of the network using the standard EM-algorithm.

It holds in fact that given complete or missing evidence we can leverage the theory developed in the previous chapters to learn the parameters of a network displaying multiple probabilistic evidence by leveraging the updated network probabilistic structure $Q_k(\mathbf{X})$ at the inference step in the E-step.

This is summarized in algorithm 7.

We note two problems, the first being that the algorithm needs to perform IPFP at each E-step given the new parameterization. You run in fact a chicken-egg problem that makes the solution to such problem very expensive. IPFP requires the parameterization of the node to be known in order to perform inference and get to the updated probabilistic structure of the network after consider probabilistic evidence. But we desire to learn such a parameterization such that we need to repeat the above until convergence.

Moreover, we note that IPFP requires to run inference and compute the new probabilistic structure for all possible realizations in the network $x \in Val(\mathbf{X})$. In order to perform such task the full joint probability distribution P($\mathbf{X}$) is be necessary. It is obvious that for big Bayesian Networks the task of estiamting the full-joint would be often infeasible to compute without the help of cost efficient inference algorithms such as the clique tree/ junction tree algorithms.

In this sense, further refinements were propose such as the *big clique* and the *modified junction tree* as discussed in Valtorta et al. (2002).

Both of such algorithms leverage the ideas of the clique trees algorithms as proposed by Shafer and Shenoy (1990). As well explained in Koller and Friedman (2009), using clique algorithms it is possible to run the inference and get the entire joint P($\mathbf{X}$) in an efficient way by propagating up- and downstreams the messages with the factor updates of interest. In such a way it is possible to get the entire joint-proability of the network in a computationally efficient way.

Algorithm 7 EM-Proabilistic: an EM algorithm for learning with probabilistic evidence

```
Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset $S$
procedure EM($\mathcal{B}$, $S$)
    Initialize $\mathcal{B}$'s parameters $\theta \leftarrow \theta^0$
    for all $t = 1, \ldots$ until convergence do
        $M - step\ as\ in\ Algorithm\ 1$
    end for

    function COMPUTE-ESS($\mathcal{B} = (G, \theta)$)
        Run IPFP given current parameterization        ▷ Note - you have to perform such iteration at each iteration
        $Q \leftarrow$ Return convergence distribution of algorithm IPFP above
        for all $i \in 1, \ldots, n$ do
            for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
                $\bar{M}[x_i, u_i] \leftarrow 0$
            end for
        end for
        for all example $S_j \in S$ do
            Run inference on $(G, Q_k, \theta)$ with evidence $d_j$
            for all i$= 1, \ldots, n$ do
                for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
                    $\bar{M}[x_i, u_i] \mathrel{+}= Q_{(G,\theta)}(x_i, u_i | d_j)$        ▷ Note that inference is based on the adjusted distriution Q obtained
  above
                end for
            end for
        end for

```

However, note that the clique tree inference algorithms as proposed by Shafer and Shenoy (1990) are not applicable in the general case of proabilistic evidence. This because, as mentioned in Valtorta et al. (2002), in the case of probabilistic evidence ‘after updating using all evidence, we still require that all appropriate marginals of the updated distribution be equal to the evidence entered. The deservedly celebrated junction tree algorithm for probability update was not designed to satisfy this requirement and in fact it does not‘. A proof of that is provided by example in Valtorta et al. (2002) and we refer to it for further details.

In order to deal with that issue in the case of multiple probabilistic evidence, Valtorta et al. (2002) proposed two extensions of the classical junction tree with different benefits in terms of time and space efficiencies. These are as mentioned the *big clique* and the *modified junction tree*, which will be briefly discussed next.

Starting with the *modified junction tree*, the idea is to guarantee the first property of Jeffrey’s probability kinematics - i.e. the property of the unchanged posterior probailistic evdience - by embedding an IPFP step into the classical junction tree algorithm. Such IPFP step at propagation time together with the iteration of the procedure until convergence, will guarantee that the converged propbability satisfies the two functional requirements of Jeffrey’s probability kinematics as proved in Csiszár (1975).

Important is, once more, to note the high computational costs of applying such an algorithm given the necessary IPFP step embedded at propagation time and the need to repeat the cycle until convergence as outlined in Valtorta et al. (2002). Two iterations cycles are involved in such an algorithm that would then be embedded in the E-step cycles of the EM-algorithm necessary for performing the parameter learning task.

A better solution might be the one of leveraging the *big clique* algorithm of Valtorta et al. (2002). There a single iteration procedure is involved - i.e. a single IPFP run - at the cost of a less space efficient procedure involving the creation of a *big clique* containing all of the specified probabilistic evidences.

Note that both of the techniques proposed by Valtorta et al. (2002) build on the theoretical fundamentals of IPFP and junction trees. However, both involve refactoring the existing junction tree inference engine to either embed an IPFP cycle into it or by creating the necessary big-cliques of interest by adding undirected edges connecting all of the variables involving probabilistic evidence.

In this sense, two further algorithms were proposed by PENG et al. (2010). These do not impose refactoring of the junction tree inference process but rather leverage a likelihood evidence reformulation. These algorithms are then especially useful as they allow a quick implementation in statistical software such as merlin. The inference step based on junction trees algorithm would not be altered in this case but a rather simple adjustment takes place before the inference step by creating a synthetic network embedding the probabilistic evidence information.

Finally, it is also important to realize that the two algorithms increases the set of time-space complexity mixtures and may therefore be the best inference solution depending on the number of probabilistic evidences as well on the network size of interest.

For the exact algorithmic formula of the two algorithm we refer to the paper of PENG et al. (2010). We note that, as mentioned by the authors, BN-IPFP-2 updates the joint distribution of the variable involving probabilistic evidence, while and BN-IPFP-1 updates the belief of the whole bayesian network such that the trade-offs are similar in spirit to the one of the standard IPFP and junction-tree based algorithms.

We conclude by expressing algorithm 8 in order to perform parameter learning in the case of multiple probabilistic evidence R(**Y**) with $\mathbf{Y} = Y_1, ..., Y_J$ - i.e. a multivariate random variable composed of *j* probabilistic evidences.

Algorithm 8 EM-Proabilistic: an EM algorithm for learning with multiple probabilistic evidence

```
Require: Bayesian network $\mathcal{B} = \langle \mathbf{X}, \mathbf{Y}, \mathbf{D}, G, \mathbf{P} \rangle$, dataset $S$
procedure EM($\mathcal{B}$, $S$)
    Initialize $\mathcal{B}$'s parameters $\theta \leftarrow \theta^0$
    for all $t = 1, \ldots$ until convergence do
        $M - step\ as\ in\ Algorithm\ 1$
    end for

    function COMPUTE-ESS($\mathcal{B} = (G, \theta)$)
        $Q \leftarrow$ BN-IPFP-1($\mathcal{B}, R(X_1, ..., X_j, ..., X_J), P(\mathbf{X}), S$)  ▷ You can also alternatively use BN-IPFP-2 as in PENG et. all paper.
        for all $i \in 1, \ldots, n$ do
            for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
                $\bar{M}[x_i, u_i] \leftarrow 0$
            end for
        end for
        for all example $S_j \in S$ do
            for all i$= 1, \ldots, n$ do    ▷ Run inference on $(G, Q_k, \theta)$ with evidence $D$ and compute ESS
                for all $x_i, u_i \in Val(X_i, \mathbf{PA}^{\mathcal{B}}_{X_i})$ do
                    $\bar{M}[x_i, u_i]$ += $Q_{(G,\theta)}(x_i, u_i | d_j)$    ▷ Note that inference is based on the adjusted distriution Q obtained above
                end for
            end for
        end for
    end function

    function BN-IPFP-1($\mathcal{B}, R(X_1, ..., X_j, ..., X_J), P(\mathbf{X}), S$)
        $Q_0 = P(\mathbf{X}), k = 1$

        while no convergence in the Q distribution do
            $j = 1 + (k-1)\ mod\ m; l = 1 + \left\lfloor \frac{k-1}{m} \right\rfloor$
            $L_{j,l}(Y^j) = \frac{R(y^j_{(1)})}{Q_{k-1}(y^j_{(1)})} : \ldots : \frac{R(y^j_{(j_s)})}{Q_{k-1}(y^j_{(j_s)})}$    ▷ Construct virtual evidence with likelihood ratio
            where $y^j_{(1)}, \ldots, y^j_{(j_s)}$ are state configurations of $Y^j$

            Obtain $Q_k$ by updating $Q_{k-1}$ with $L_{j,l}(Y^j)$ using Pearl's virtual evidence method

            $k$ += 1
        end while

```

9 Conclusion, Criticalities and Further Work

In this script we exposed the theoretical background in order to deal with the parameterization riddle of Bayesian Networks.

The theory was elaborated bottom up. Starting with the most basic case of *complete evidence* we showed the property of local and global likelihood decomposition in Bayesian Networks. It was then straightforward to see that both in the case of Maximum Likelihood Estimation as well as in the case of Bayesian Learning it was possible to estimate the parameters by optimizing the individual CPDs.

Turning to the case of *missing evidence* we showed that the above properties are lost. Despite this, we formally proved that it is possible to apply the EM-algorithm in order to reach a local optimum. Moreover, we argued how after the inference step the global *expected* likelihood decomposition is satisfied such that it is possible to maximize the parameters of the individual CPDs independently in such a step.

We continued by introducing the case of parameter learning under likelihood evidence as presented in Wasserkrug et al. (2021). The key there was to recognize the possibility of extending the bayesian network by incorporating the likelihood evidence by the means of virtual evidence nodes as proposed by Pearl (1987). Given such a method it was possible to frame likelihood evidence as hard-evidence, such that by the "elimination" of it, the network evidence would just consists of missing-evidence and hard-evidence. In that case, the classical EM-algorithm as discussed in section 3 applies.

On the top of the arguments proposed by Wasserkrug et al. (2021) we showed that it is possible to extend the EM-algorithm to the case of maximum a posterior Bayesian parameter learning without loosing the correctness and convergence properties of the algorithm. In the specific we showed that the inference step; i.e. the expectation is unaltered by the introduction of the priors in Bayesian Networks. However, the M-step must be adjusted in order to take the influence of the prior into account. In this sense we extended the merlin engine to deal with case of Bayesian Learning for the case of Multinomial-Dirichlet CPDs both in the case of missing evidence and likelihood evidence. Moreover, we exposed the general theory for generalizing the modeling possibilities to the case of exponential families CPDs and we exposed the possibility of leveraging M-projection in order to estimate the parameters of choice in that case.

Finally, we turned to the case involving probabilistic evidence. There again the epiphany comes with the realization that after the specification of a probabilistic network structure taking the probabilistic evidence constraints into account it is possible to choose the parameterization maximizing such an adjusted constrained probabilistic structure via the classical EM-algorithm. In this sense, the difficulty consists in finding ways to economically reach the new network probabilistic structure satisfying the probabilistic evidence constraints. This especially considering the fact that at each step of the EM algorithm, a new probabilistic structure for the entire network under the given parameterization has to be computed. We briefly introduced the IPFP theorem that poses the theoretical fundamental to obtain the network probabilistic structure of choice. We then exposed multiple modeling possibilities to incorporate the IPFP step - or derivate of it - into the EM algorithm. We mentioned the algorithms of Valtorta et al. (2002) offering good performance and different space-time complexity trade-off. We noted then that implementing such algorithms in statistical software would require to refactor the entire code-base of the junction tree algorithm used at inference step. Moreover, they are restricted by definition to the usage of junction tree algorithms at inference time. Given the above, we proposed and partially implemented the algorithms of PENG et al. (2010). These are particularly interesting both for their characteristic of expanding the time-space complexity trade-off of the models of Valtorta et al. (2002) as well due to their characteristic of working via Pearl's virtual evidence method. Especially the latter property is useful as it allows both to quickly implement the algorithm in statistical software, as we performed in merlin, as well as perform the inference step with any algorithm of choice.

Overall, we therefore extended the theory introduced by Wasserkrug et al. (2021) by noting that extensions of the EM algorithm are possible and easily integrable in statistical software as was demonstrated in merlin. We both proposed extension of the algorithm to deal with maximum a posteriori parameter estimation for the case of Bayesian Learning as well as extensions dealing with probabilistic evidence. For both we argued that the theory on which the EM-algorithm relies is satisfied such that we continue to hold the correctness and convergence properties of the algorithm.

We conclude by noting that upon the realization that the EM-algorithm consists of an inference and maximization step given a probabilistic joint distribution specified by the Bayesian Network it comes as no surprise that once the probabilistic structure implied by the network satisfies all of the constrains implied by uncertain evidence, the EM-algorithm will find a local maxima parameterization of it. When dealing with uncertain evidence the difficulty consists therefore much more in computing such constrained joint-probability in a computational way that is time and space efficient rather than in a fundamental re-elaboration of the EM-algorithm itself.

9.1 Criticalities and Further Work

The general flaw with this script, is the fact that it focuses on the theoretical possibilities of the parameterization learning task in Bayesian Networks without an in depth analysis of the computational burden to obtain it. In this sense, a deeper analysis of the computational effort of applying each algorithm is necessary. This both at an algebraic level as well as in an empirical way performing some simulation exercise as in Wasserkrug et al. (2021)[3].

Starting with the case of maximum a posteriori distribution for the case of Bayesian Learning we discussed the limitation of working with well known conjugate priors such that the posterior distribution will be a well known distribution. On the other hand, we briefly touched on the possibility of relying on simulation techniques in order to get to the maximum a posteriori parameterization. Finally, we mentioned the possibility of finding the maximum of the posteriori distribution via gradient-based numerical methods. An in depth study of such algorithms and especially of the performance of Meng (2016) would be of particular interest.

Turning to the case of the application of the EM-algorithm in the case of uncertain evidence, an in depth computational analysis of the algorithm is especially important. This because it is necessary to obtain a constrained joint-distribution Q, taking the uncertain evidence into account, *at each step of the EM-algorithm* until convergence. You therefore see that the costs of obtaining a new constrained joint-distribution is scaled linearly in the numbers of necessary iterations steps. Therefore, it is especially important to find the right combination of time-space computational effort such that the overall effort of obtaining the constrained joint-distribution and performing inference in the network is minimized for the network of interest. In this sense, it might be useful to elaborate some rule of thumb theory to choose the most suitable algorithm for different classes of Bayesian Networks. Finally, due to such especially important issues, it might be meaningful to further implement the algorithms of Valtorta et al. (2002) into statistical software as merlin in

[3] Consider in this sense the hugin engine.

order to increase the time-space computational mix available and leverage the best possible algorithm for each network configuration.

We conclude by noting that due to the vast literature and modeling possibilities available for Bayesian Networks the presented theory just scratches the surface of the possible algorithms available for obtaining the most suitable network parameterization. A further avenue of research that we completely ignored in the script is the one of performing the inference step via approximate inference methods and the inspection of the behaviour of the EM-algorithm method in such a case. Introductory reading material and relevant literature may be found in Koller and Friedman (2009). We close by noting that an analysis of the benefits of combining such methods in computational involving cases such as in the case of the algorithms dealing with uncertain evidence might be particularly interesting.

References

Ole Barndorff-Nielsen. Hyperbolic distributions and distributions on hyperbolae. *Scandinavian Journal of statistics*, pages 151–157, 1978.

Wray Buntine. Tree classification software. 1993.

Hei Chan and Adnan Darwiche. On the revision of probabilistic beliefs using uncertain evidence. *Artificial Intelligence*, 163(1):6790, Mar 2005. ISSN 0004-3702. doi: 10.1016/j.artint.2004.09.005. URL `http://dx.doi.org/10.1016/j.artint.2004.09.005`.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158, 1975.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Daniel Fink. A compendium of conjugate priors. *See http://www. people. cornell. edu/pages/df36/CONJINTRnew% 20TEX. pdf*, 46, 1997.

Dan Geiger, David Heckerman, and Christopher Meek. Asymptotic model selection for directed networks with hidden variables. In *Learning in Graphical Models*, pages 461–477. Springer, 1998.

David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

Richard C Jeffrey. *The logic of decision*. University of Chicago press, 1990.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Steffen L Lauritzen. The em algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, 19(2):191–201, 1995.

Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

Roderick JA Little. Inference about means from incomplete multivariate data. *Biometrika*, 63(3):593–604, 1976.

Thomas A. Louis. Finding the observed information matrix when using theemalgorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):226233, Jan 1982. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1982.tb01203.x. URL `http://dx.doi.org/10.1111/j.2517-6161.1982.tb01203.x`.

Lingyao Meng. Method for computation of the fisher information matrix in the expectation-maximization algorithm. *arXiv preprint arXiv:1608.01734*, 2016.

Ali Ben Mrad, Véronique Delcroix, Sylvain Piechowiak, Philip Leicester, and Mohamed Abid. An explication of uncertain evidence in bayesian networks: likelihood evidence and probabilistic evidence. *Applied Intelligence*, 43 (4):802824, Jun 2015. ISSN 1573-7497. doi: 10.1007/s10489-015-0678-6. URL `http://dx.doi.org/10.1007/s10489-015-0678-6`.

Judea Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32(2):245–257, 1987.

Judea Pearl. Bayesian networks. 2011.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

YUN PENG, SHENYONG ZHANG, and RONG PAN. Bayesian network reasoning with uncertain evidences. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(05):539564, Oct 2010. ISSN 1793-6411. doi: 10.1142/s0218488510006696. URL `http://dx.doi.org/10.1142/s0218488510006696`.

Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

Paul A Ruud. A comparison of the em and newton-raphson algorithms. 1989.

Robert Schlaifer and Howard Raiffa. *Applied statistical decision theory*. 1961.

Glenn R Shafer and Prakash P Shenoy. Probability propagation. *Annals of mathematics and Artificial Intelligence*, 2 (1):327–351, 1990.

José M Bernardo Smith and Adrian F M. Bayesian theory. *Measurement Science and Technology*, 12(2):221222, Jan 2001. ISSN 1361-6501. doi: 10.1088/0957-0233/12/2/702. URL `http://dx.doi.org/10.1088/0957-0233/12/2/702`.

David J Spiegelhalter and Steffen L Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, 1990.

Geir Storvik. Numerical optimization of likelihoods: Additional literature for stk1120. 2007.

Anne Randi Syversveen. Noninformative bayesian priors. interpretation and problems with construction and applications. *Preprint statistics*, 3(3):1–11, 1998.

Marco Valtorta, Young-Gyun Kim, and Jií Vomlel. Soft evidential update for probabilistic multiagent systems. *International Journal of Approximate Reasoning*, 29(1):71106, Jan 2002. ISSN 0888-613X. doi: 10.1016/s0888-613x(01)00056-1. URL `http://dx.doi.org/10.1016/s0888-613x(01)00056-1`.

Segev Wasserkrug, Marinescu Radu, Sergey Zeltyn, Evgeny Shindin, and Yishai Feldman. Learning the parameters of bayesian networks from uncertain data. 2021.

Sewall Wright. Correlation and causation. *J. agric. Res.*, 20:557–580, 1921.