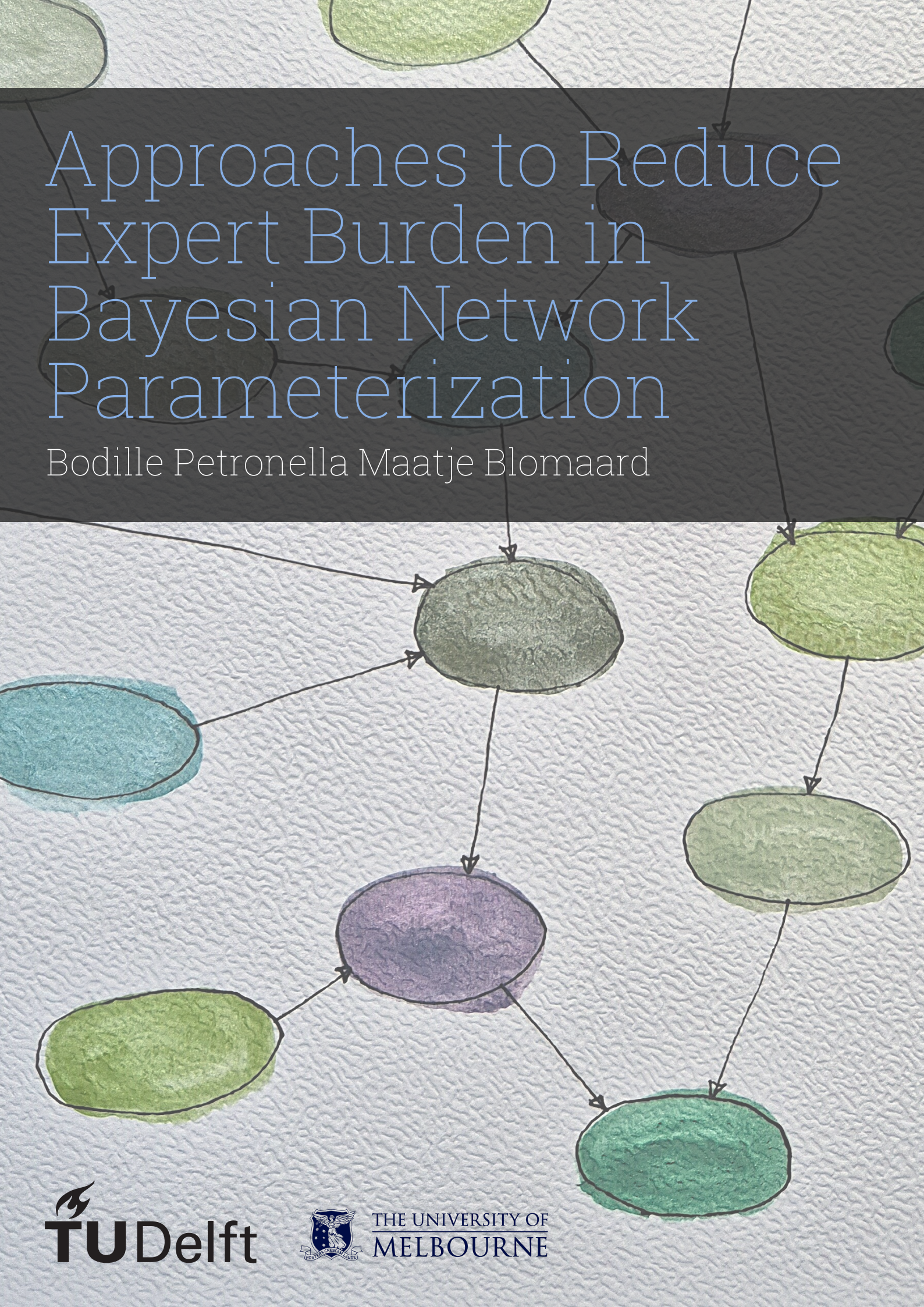
Approaches to Reduce Expert Burden in Bayesian Network Parameterization

Bodille Petronella Maatje Blomaard



TU Delft

THE UNIVERSITY OF MELBOURNE

APPROACHES TO REDUCE EXPERT BURDEN IN BAYESIAN NETWORK PARAMETERIZATION

by

BODILLE PETRONELLA MAATJE BLOMAARD

to obtain the degree of Master of Science in Applied Mathematics
at the Delft University of Technology,
to be defended publicly on Thursday July 11, 2024 at 10:30 a.m.

Project duration:	December 11, 2023 - July 11, 2024	
Thesis committee:	Dr. Ir. G. F. Nane,	Delft University of Technology
	Dr. A. M. Hanea,	University of Melbourne
	Dr. M. Vittorietti,	Delft University of Technology

How can I know that, when I've never tried?

Pippi Longstocking, Astrid Lindgren

ABSTRACT

Bayesian Networks (BNs) are popular models that represent complex relationships between variables, which can be quantified by Conditional Probability Tables (CPTs) in the discrete case. If data are not sufficient, experts can be involved to assess the probabilities in the CPTs through Structured Expert Judgment (SEJ), which is often a burdensome task. To lighten the elicitation burden, several methods have been developed previously to construct CPTs using a limited number of input parameters, such as the Ranked Nodes Method (RNM), InterBeta and Functional Interpolation. These methods are first analyzed theoretically, where limitations and potential improvements are determined, which were used as inspiration to develop extensions to the methods. The methods and newly developed extensions, including "ExtraBeta" and "AutoRNM", were applied to reconstruct fully elicited CPTs. Finally, simulation studies are performed to find best practices for InterBeta. InterBeta with parent weights is determined as the best-performing method, and the AutoRNM and ExtraBeta extensions are worth exploring further.

ACRONYMS

BN Bayesian Network

cdf cumulative density function

CPT Conditional Probability Table

DAG Directed Acyclic Graph

DBN Dynamic Bayesian Network

EPT Elicited Probability Table

EWDM Equal Weights Decision Maker

GJP Good Judgment Project

IF Interpolation Factor

KL Kullback-Leibler

pdf probability density function

pmf probability mass function

PWDM Performance-based Weights Decision Maker

RNM Ranked Nodes Method

SEJ Structured Expert Judgment

SHELF SHeffield ELicitation Framework

CONTENTS

1

INTRODUCTION

A Bayesian Network (BN) is a well-known modeling tool that can effectively represent complex systems through a graphical interface. Applications range from modeling volcano eruptions (Christophersen et al., 2018), biosecurity (Hanea et al., 2022) and healthcare (Kyrimi et al., 2021) to modeling nuclear applications (Cooke & Goossens, 1999). One of the possible ways to parameterize Bayesian Networks (BNs) is to ask domain experts in a structured manner. However, an important drawback to involving experts in studies is the resources needed. The combination of many questions of interest, structured protocols, limited time, and preferably more than one expert involved, often requires concessions to be made. This thesis explores methods to minimize resource demands by alleviating the burden on experts involved in BN parameterization.

BNs are graphical models that can represent full complex relationships between multiple variables in an organized way. A BN is specified by an acyclic graph with nodes for all variables and directed arcs representing the dependence relationships between variables. Arcs are directed from parent nodes to child nodes, and the presence of such arcs defines the dependence between the variables. The nodes are specified by (conditional) probability distributions, for the static discrete case, these can be given in the shape of a Conditional Probability Table (CPT). For a child node, this is a table filled with conditional probabilities for each combination of states that the child node can take and the parent nodes can take.

The conditional probabilities needed to fill CPTs can be found using data, or if insufficient data exists, experts can be involved. Structured Expert Judgment (SEJ) is the technique that enables data to be collected from experts in a structured way, that accounts for uncertainty. Different methods exist, such as the Delphi method (Brown, 1968), the Classical Model (Cooke, 1991), and the IDEA protocol (Hanea et al., 2017), which aim to collect unbiased data from experts. These methods pose a large burden on experts when many questions of interest need to be answered, which is the case when CPTs are elicited.

Case studies often have limited resources, which may force BNs to be simplified such that timely parameterization by the experts is ensured. For instance, reducing the number of states for certain nodes, as demonstrated by Barons et al., 2018, lowers the number of probabilities that need to be assessed. However, this simplification implies that less detail can be captured by the model. Instead of modifying the structure of a BN, other solutions are needed to decrease the number of parameters experts need to assess.

For this purpose, CPT construction methods have been developed that only require fewer values to be elicited. These methods include, but are not limited to, the Noisy-OR method (Pearl, 1988), the Ranked Nodes Method (RNM) (Fenton et al., 2007), InterBeta (Mascaro & Woodberry, 2022), and Functional Interpolation (Podofillini et al., 2014). Each of the methods requires a different set of input parameters to be elicited from the experts, these include CPT rows, different types of weights, and variance parameters.

Although some of these CPT construction methods have been available for a while already, there exists limited literature on the applications of these methods. A selection of methods has been previously tested and compared to data (Knochenhauer et al., 2013; Mkrtchyan et al., 2016; Zio et al., 2022), but these comparisons do not include InterBeta or are tested against fully expert-elicited CPTs. InterBeta was separately tested on fully expert-elicited CPTs. Thus, at the moment of writing, no guidelines exist that help modelers choose the most appropriate method for specific situations.

1

This thesis aims to investigate and provide methodological insights and empirical evidence on what CPT construction methods to use. To both maximize the accuracy with which CPTs can be reconstructed and minimize the expert burden. This will be done by examining the methods; finding possible limitations and points of improvement; comparing the performance of the different methods, when tested on previously expert-elicited CPTs; and finally testing on simulated CPTs. The research questions that are investigated in this thesis are:

- How do existing methods, such as RNM, InterBeta and Functional Interpolation compare against each other, in terms of accuracy and elicitation burden, when applied to reconstructing existing fully elicited CPTs?
- How can each method be improved, to offer more flexibility, improve accuracy, or limit the expert burden?
- How should the InterBeta method be tailored given the network structure, underlying correlation structure, or other factors?

The first part of the thesis aims to establish a solid foundation of background knowledge which will later be applied to answer the research questions. Starting with providing a theoretical background on Bayesian Networks (BNs) in Chapter 2, where some elementary theory is discussed, definitions are stated, and an example network is given. The thesis continues with an extensive overview of Structured Expert Judgment (SEJ) in Chapter 3. First, common heuristics and biases that exist in human thinking are discussed, and then a set of well-known SEJ methods are introduced, such as the Delphi method, Bayesian methods, the Classical Model, the SHeffield ELicitation Framework (SHELF), and the IDEA protocol. Chapter 3 also contains a brief discussion on what a group of experts should look like, and what elicitation practices exist. Chapter 4 contains the final literature part of this thesis and discusses various methods for constructing Conditional Probability Tables (CPTs) for BN, including the full CPT elicitation, the Noisy-OR model, the Ranked Nodes Method (RNM), InterBeta, and Functional Interpolation. For each of the methods, the technical computation details are discussed, as well as the elicitation framework and possible limitations and improvements to the methods.

Once the theoretical foundation has been laid, the thesis continues with an overview of fully elicited CPTs concerning the abundance of pollinators in the UK (Barons et al., 2018), household food security in Australia (Kleve & Barons, 2021), and the future of the polar bear population (Atwood et al., 2016) in Chapter 5. After which, the implementations of CPT construction methods - RNM, InterBeta, and Functional Interpolation - are detailed in Chapter 6. This chapter also includes extensions to each of the methods which aim to improve the accuracy of the methods and to lighten the elicitation burden. In Chapter 7 the results of the applications of RNM, InterBeta, and Functional Interpolation are presented. First, all methods are discussed separately, and afterward, the methods are compared, where both the accuracy and elicitation burden are taken into account. Following the application of the methods on expert-elicited data, the InterBeta method and its extensions are also applied to simulated CPTs. In Chapter 8, the general CPT simulation strategy, the different simulation studies, and its results are set out.

Finally, Chapter 9 contains a discussion of the thesis and recommendations for future research, and Chapter 10 contains the conclusion. In the final chapter each of the research questions is answered.

2

BAYESIAN NETWORKS

This chapter will provide background theory on Bayesian Networks (BNs). A BN is a type of graphical model, which represents a set of variables and their conditional dependencies with the help of a graph. This chapter will start with a discussion of graph theory, this includes some structural properties which can be linked to conditional dependencies between variables. When this theoretical basis is laid, dependence between variables is defined.

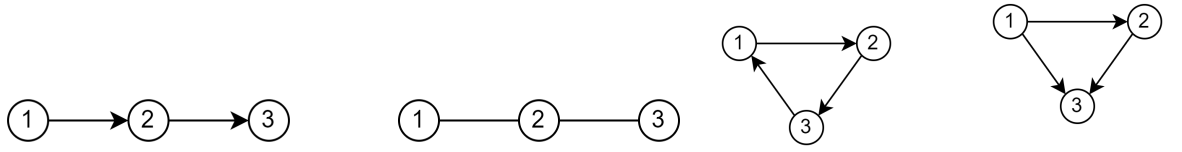
Then, the BN is defined in Section 2.3, after which the structure of BNs is linked to dependencies between variables in the network. The section continues with an overview of the differences between static and dynamic networks, and the differences between discrete and continuous networks. Moreover, the size of a discrete BN is defined and it is discussed how structural changes to the graph can lead to a reduction in size. An example of a BN is given that will be referenced throughout the thesis. Finally, the chapter concludes with an overview of applications of BNs in the available literature.

2.1. GRAPH THEORY

This section will contain the theory that is necessary for understanding BNs. A BN is a type of **graphical model**, which represents a joint probability distribution whose structure is described by a graph. This section contains an overview of the necessary graph theory for the definition of BNs.

A **graph** is a pair $G = (V, E)$, where $V = \{v_1, .., v_n\}$ is a finite set of vertices/nodes, and $E = \{\{v_i, v_j\}, v_i, v_j \in V, v_i \neq v_j\}$ is a finite set of edges/arcs. There are two types of edges, undirected edges $:= \{v_i, v_j\}$, which do not point in a particular direction; and directed edges $:= (v_i, v_j)$, which do have a direction. When a graph contains only directed edges it is called a **directed graph**; an example is given in Figure 2.1a. Figure 2.1b gives an example of an **undirected graph**, where each edge is undirected. When a graph contains both types of edges, it is called partially-directed. A **path** between edges v_1 and v_k is defined as a set of nodes $v_1, ..., v_k$ where each $(v_i, v_{i+1}) \in E$ for $i = 1, ..., k-1$. If $v_1 = v_k$, then it is called a **cycle**. In Figure 2.1c, a cycle can be found: 1,2,3,1.

If edge (v_i, v_j) is directed from vertex v_i to v_j, vertex v_i is called the **parent** of v_j. Similarly, v_j is the **child** of v_i. For example, in Figure 2.1a, node 1 is the parent of node 2, and node 2 is the child of node 1. Related to parent and child nodes, ancestors and descendants can be defined. If there exists a path from v_i to v_j, then v_i is an **ancestor** of v_j and v_j is a **descendant** of v_i. Furthermore, we recognize the following sets of vertices:



(a) Directed graph. (b) Undirected graph. (c) Cyclic graph. (d) Directed Acyclic Graph.

Figure 2.1: Four examples of graphs.

- $pa(v_i)$: set of parents of v_i,
- $ch(v_i)$: set of children of v_i,
- $an(v_i)$: set of ancestors of v_i,
- $de(v_i)$: set of descendants of v_i.

If $pa(v_i) = \emptyset$, then v_i is called a **root node**. If $ch(v_i) = \emptyset$, then v_i is called a **leaf node**.

For example, in Figure 2.1a, node 1 is an ancestor of node 3 and node 3 is a descendant of node 1. For node 3 the following sets can be found, $pa(3) = \{2\}$, $ch(3) = \emptyset$, $an(3) = \{1,2\}$, and $de(3) = \emptyset$, which means that node 3 is a leaf node.

2.1.1. DIRECTED ACYCLIC GRAPHS

Graphically, a BN is a Directed Acyclic Graph (DAG), which is a directed graph that does not contain any cycles. This is equivalent to each parent node coming before the child node in a certain ordering. For the formal definition of a DAG, first, it's needed to define an ordering.

Definition 2.1.1. *There exists a **complete ordering** of the graph if there exists a relationship '$<$' on the elements of $V = \{v_1, ..., v_n\}$, such that for all $v_i, v_j, v_l \in V$:*

(i) $v_i < v_j$ *or* $v_i > v_j$, *and*

(ii) $v_i \not< v_i$, *and*

(iii) if $v_i < v_j$ *and* $v_j < v_l$ *then* $v_i < v_l$.

Using this definition, the equivalence in Theorem 2.1.2 is found.

Theorem 2.1.2. *For a directed graph, the following two conditions are equivalent:*

(i) There is no directed cycle.

(ii) There exists a complete ordering of the graph.

A DAG is a graph that contains only directed edges and contains no directed cycle. Using the results from Theorem 2.1.2, this is equivalent to a directed graph that can be completely ordered.

2.1.2. D-SEPARATION

To study the dependencies between the variables encoded in the nodes of a a graph, the structure can be analyzed. In this section, d-separation is introduced. A **trail** between two nodes $X = v_1$ and $Y = v_k$ is a set of nodes $v_1, ..., v_k$ such that either $\{v_i, v_{i+1}\} \in E$ or $\{v_{i+1}, v_i\} \in E$. Thus, a trail does not have to follow the directions of the arcs. A node Z which is on the trail, somewhere in between nodes X and Y, can be classified as **serial**, **diverging**, or **converging**. Figure 2.2 gives examples of each of these types.

(a) Serial. (b) Serial. (c) Diverging. (d) Converging.

Figure 2.2: Four types of node connection directions.

Definition 2.1.3. *A trail between nodes X and Y is blocked by a set $\mathcal{Z}$ if*

(i) the trail contains a node $Z \in \mathcal{Z}$ and the connection at Z is either serial or diverging, or

(ii) the trail contains a node W such that $W \notin \mathcal{Z}$, $de(Z) \notin \mathcal{Z}$, and the connection at W is a converging.

Definition 2.1.4 (d-Separation)**.** *Two nodes X and Y are d-separated by a set $\mathcal{Z}$ if all trails between X and Y are blocked by $\mathcal{Z}$.*

This definition of d-separation is later used to link the structure of a graph to the dependencies between variables.

2.2. PROBABILITY THEORY

An important part of BN theory is the probability theory behind it. In this section, the fundamental theory about (conditional) independence is set out. This theory will be linked to the graphical structure of BNs in the next section.

The random variables are described by a probability mass function (pmf), for discrete, or probability density function (pdf), for continuous random variables. To avoid confusion, a simplified notation for the pmf and pdf of a random variable X is used: f_X. The joint probability density (or mass) function of X and Y is then written as $f_{X,Y}$.

Definition 2.2.1 (Independence). *Two random variables X_i and X_j with pdf (pmf) $f_{X_i}(x_i)$ and $f_{X_j}(x_j)$ respectively, are independent if*

$$f_{X_i,X_j}(x_i,x_j) = f_{X_i}(x_i)\cdot f_{X_j}(x_j).$$

This is denoted as $X_i \perp X_j$.

Following the definition of independence, also conditional independence can now be defined. Two random variables can be conditionally independent given a set of random variables.

Definition 2.2.2 (Conditional independence). *Two random variables X_i and X_j are conditionally independent given random variable Y ($X_i \perp X_j|Y$) if*

$$f_{X_i,X_j|Y}(x_i,x_j|y) = f_{X_i|Y}(x_i|y)\cdot f_{X_j|Y}(x_j|y). \tag{2.1}$$

Equivalently, when $X_i \perp X_j|Y$:

$$f_{X_i|X_j,Y}(x_i|x_j,y) = f_{X_i|Y}(x_i|y). \tag{2.2}$$

2.3. BAYESIAN NETWORKS

It is now time to define the BN. The definition will first be made in general, later in this section the distinction will be made between static and dynamic BN and between discrete and continuous BN.

Definition 2.3.1 (Bayesian Network). *A Bayesian Network BN is a graphical model which is specified by:*

(i) a Directed Acyclic Graph (DAG) $G = (V,E)$, where vertices $v_1,...,v_n \in V$ represent random variables $X_1,...,X_n$. The directed edges $e_1,...,e_k \in E$ represent the dependence relationships between variables. If there is no arc between vertices v_i and v_j, and $v_i < v_j$ then $X_i \perp X_j \,|\, X_{pa(v_j)}$, or if $pa(v_j) = \emptyset$ there is independence between the variables $X_i \perp X_j$,

(ii) a set of conditional probabilities $f_{X_i\,|\,X_{pa(v_i)}}(x_i|x_{pa(v_i)})$.

The joint pdf or pmf $f_{X_1,...,X_n}(x_1,...,x_n)$ can be represented by a BN as the product over the conditional densities (or probabilities) of all variables:

$$f_{X_1,...,X_n}(x_1,...,x_n) = \prod_{i=1}^{n} f_{X_i\,|\,X_{pa(v_i)}}(x_i|x_{pa(v_i)}). \tag{2.3}$$

In case $pa(v_i) = \emptyset$ (i.e. node v_i has no parents) the marginal pdf or pmf is used instead: $f_{X_i}(x_i)$. For the example in Figure 2.1a this would lead to the following joint pdf or pmf:

$$f_{X_1,X_2,X_3}(x_1,x_2,x_3) = f_{X_3\,|\,X_2}(x_3\,|\,x_2) f_{X_2\,|\,X_1}(x_2\,|\,x_1) f_{X_1}(x_1).$$

2.3.1. INDEPENDENCE IN BAYESIAN NETWORKS

The graphical structure of a BN is not only visually pleasing but also represents the dependence structure between the variables in the model. In Definitions 2.2.1 and 2.2.2, it is defined what (conditional) independence means for random variables. Using Definition 2.1.4, which defines the concept for d-separation, the structure of a BN can be linked to the dependence structure between the variables in the model.

First, the joint probability of the variables represented by BN G will be denoted as $P_G(\mathbf{X})$, where $\mathbf{X}$ is the set of nodes in G.

Lemma 2.3.2. *Suppose that G is a Bayesian network with leaf node Y, and G_0 is the BN resulting from G when Y is removed. Let $\mathcal{X}$ be the set of all nodes in G_0, then*

$$P_G(\mathcal{X}) = P_{G_0}(\mathcal{X}). \tag{2.4}$$

Thus, leaf nodes can be removed from a model without changing the distribution of the remaining nodes. This can be extended to removing all of the nodes which are not ancestors of a set of nodes. The set of nodes $\mathcal{X}$ is called **ancestral** if for all nodes in the set, the ancestors are included.

Lemma 2.3.3. *Suppose that G is a Bayesian network, and $\mathcal{X}$ is an ancestral set of nodes. G_0 is the BN resulting from G when all nodes outside of $\mathcal{X}$ are removed, then*

$$P_G(\mathcal{X}) = P_{G_0}(\mathcal{X}). \tag{2.5}$$

In the next theorem, the link between d-separation and independence is made.

Theorem 2.3.4. *Let X, Y be two nodes in a Bayesian network, and let $\mathbf{Z}$ be a set of nodes that does not contain X, or Y. If $\mathbf{Z}$ d-separates X and Y, then*

$$X \perp Y | \mathbf{Z}.$$

To illustrate the concept of d-separation and independence of variables in a BN, Figure 2.3 can be used. The nodes A, B, C, D, E, and Z represent the random variables A, B, C, D, E, Z. For example, nodes A and B are not d-separated by node Z, thus variables A and B are not conditionally independent given Z, in fact they are independent. On the other hand, nodes A and C are d-separated by node Z, meaning that variables A and C are conditionally independent given Z, so $A \perp C | Z$. This is an example of the Local Markov property, which states that child nodes are independent of their ancestors given their parents. In the following definition, the formal statement of the Local Markov property is given.

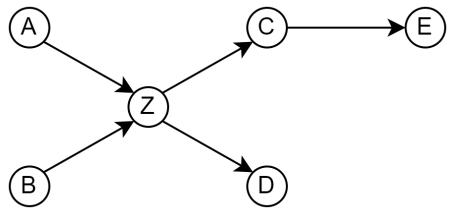


Figure 2.3: Example BN for the illustration of d-separation.

Definition 2.3.5 (Local Markov property)**.** *Let X be a node in a Bayesian network, $pa(X)$ is the set of parent nodes of X, and $an(X) \backslash pa(X)$ is the set of ancestors of X excluding the parent nodes. Then according to the local Markov property,*

$$X \perp an(X) \backslash pa(X) | pa(X).$$

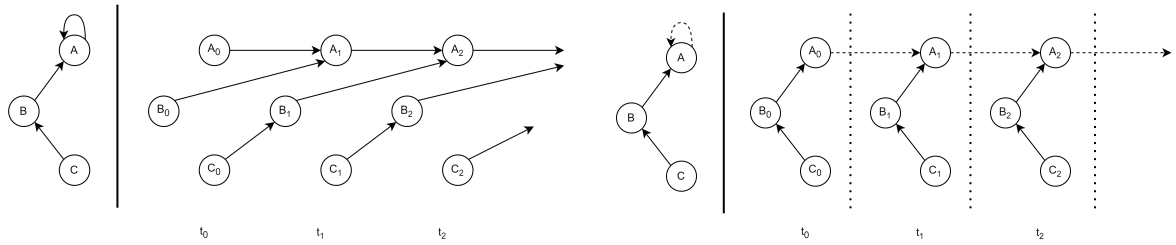
2.3.2. STATIC VS. DYNAMIC

There are different types of BNs, which can be distinguished by various characteristics, one of which is whether the BN is static or dynamic. A static BN does not change over time and is determined by a fixed/atemporal joint probability distribution. Dynamic Bayesian Networks (DBNs) are dependent on time. This overview is based on the descriptions from Mihajlovic and Petkovic, 2001.

Like static BNs, DBNs use graphs to show the structure of the model. Two approaches for representing the time in BNs can be compared, the first only consists of influences between time steps, as depicted in Figure 2.4a. Left of the bar the temporal BN is presented, and on the right, the equivalent DAG is shown over time. Note that for temporal BNs cycles are allowed to represent the causal effect within one node over time, unlike for static BNs. In addition, for static BNs an arc represents a dependence relation between variables, however, for this type of DBNs the arcs represent causal effects. This representation does not allow arcs between nodes at the same point in time, all arcs are between different points in time.

The other approach is shown in Figure 2.4b, where time slices are used to represent the time dependencies of the DBN. In the figure, the time slices are separated by dashed vertical lines. The DBN consists of

sub-models representing the system at different points in time. In this approach, there may be dependence relations between variables in one time slice (solid arcs) and temporal relations between variables in consecutive time slices (dashed arcs). The left figure once again shows a cycle around node A, note that this is only allowed for the dashed arcs.



(a) Example of a DBN represented by a temporal model.

(b) Example of a DBN represented by time slices.

Figure 2.4: Two examples of different types of Dynamic Bayesian Networks. For both figures, the left shows the structure of the DBN, and the right is the model over time.

In this thesis, dynamic BNs will not be further explored. The inclusion of this overview is purely provided for the sake of completeness.

2.3.3. DISCRETE VS. CONTINUOUS

Additionally, BNs can be either discrete, continuous or a hybrid of the two. In Discrete BNs, all random variables are discrete, which means that they can only take on a countable number of different values. For BNs this means that each node has a countable number of states, for example: Low, Moderate, and High. For a child node X_C with states $x_C^1, ..., x_C^s$, the random variable can be described by pmf:

$$p_{X_C|X_1,...,X_n}(x_C^i|x_1,...,x_n) = \mathbb{P}[X_C = x_C^i | X_1 = x_1, ..., X_n = x_n]$$
$$\text{s.t.} \quad \sum_{i=1}^{s} p_{X_C|X_1,...,X_n}(x_C^i|x_1,...,x_n) = 1,$$

for a combination of parent node states $(x_1,...,x_n)$. To fully describe the child node, the pmf needs to be given for all combinations of parent node states and child node states. The set of conditional probabilities for a child node can be given in the form of a table, known as a Conditional Probability Table (CPT).

Table 2.1 provides an example of such a CPT, which corresponds to the leaf node of the BN in Figure 2.6. Each row represents a combination of parent states, and each column corresponds to a child node state. The CPT has 27 rows and 3 columns in total, which should be filled with probabilities conditioned on the parent state combination of that row. The sum of the probabilities in one row should sum to one.

	Parent nodes			Child node: Y		
	X_1	X_2	X_3	Dull	Okay	Amazing
1	High	High	High	0.01	0.04	0.95
2	High	High	Moderate	0.02	0.08	0.90
3	High	High	Low	0.03	0.09	0.88
4	High	Moderate	High	0.02	0.08	0.90
5	High	Moderate	Moderate	0.05	0.10	0.85
6	High	Moderate	Low	0.06	0.14	0.8
7	High	Low	High	0.05	0.10	0.85
8	High	Low	Moderate	0.10	0.40	0.50
9	High	Low	Low	0.15	0.60	0.25
10	Moderate	High	High	0.05	0.10	0.85
⋮	...	...	...	...	...	...
26	Low	Low	Moderate	0.92	0.06	0.02
27	Low	Low	Low	0.95	0.04	0.01

Table 2.1: Example CPT corresponding to the example BN as in Figure 2.6.

For continuous BNs, the modeled variables are all continuous. In this case, random variable X_i can be described by a pdf $f_{X_i}(x)$, such that $\int_{-\infty}^{\infty} f_{X_i}(x)\,dx = 1$, given a combination of parent node states. As there are uncountably many states that the variable can take, the conditional probabilities for a child node can no longer directly be given by CPTs. Instead, a series of conditional probability distributions for continuous variables can be used.

Finally, BNs can also include both discrete nodes and continuous nodes, such BNs are also known as hybrid BNs. Continuous and hybrid BNs can also be made discrete by discretizing all variables.

In this thesis, continuous BNs will not be considered further. Hence, any mention of "BN" will refer exclusively to a static discrete Bayesian network.

2.3.4. BAYESIAN NETWORK STRUCTURE

The structure of a BN is defined by a graph that represents the model. This section will first discuss the influences of the BN structure on the number of CPTs and the CPT sizes that need to be specified. Additionally some ways to reduce the CPT size using structural changes are given. Finally, some references are given about how the BN structure can be determined.

The number of CPTs that need to be specified is determined by the structure of the BN, and so is the size of all individual CPTs. As the root nodes of a BN can be specified by marginal distributions or probabilities, the number of CPTs that exist in a BN is equal to the number of nodes that are not root nodes. The size of a CPT depends on the number of parent nodes and the number of states each parent node and child node has:

$$\text{Size(CPT)} = s_C \prod_{i}^{n} s_i, \tag{2.6}$$

where s_C, s_i are the number of states for the child node and parent node i respectively. In the next section an example of a small BN is given with three parent nodes that each have a directed arc that points towards a single child node, as is shown in Figure 2.6. For this BN, one CPT containing 81 probabilities needs to be specified for the child node, and three marginal probability distributions are needed for the parent nodes.

If the full BN would need to be parameterized by elicitation, it would require as many parameters to be elicited as the sum of all CPT sizes in the BN, which can quickly become too large for full elicitation. There are several ways to reduce the number of parameters to fully specify a BN. For example, the CPT sizes can be reduced by limiting the number of states a node can have, or by "divorcing" parent nodes. To divorce two parent nodes, an intermediate node is placed between the child node and the two parent nodes which summarizes the parent nodes (Olesen et al., 1989). In Figure 2.5, an example is shown how the structure changes when nodes B and C are divorced. If all nodes in the example network consist of s states, the total number of conditional probabilities needed for the CPTs in the BN would reduce from s^4 to $2s^3$. The process can be made

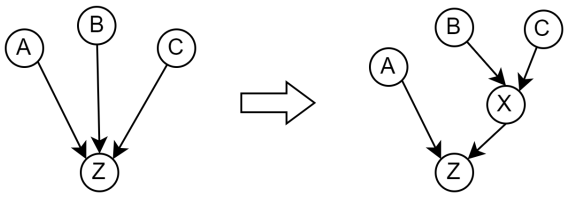


Figure 2.5: The result of divorcing parent nodes B and C.

"child-friendly" by choosing parents to divorce based on similarity, such that meaningful sub-BNs are created (Röhrbein et al., 2009). Divorcing nodes should be done with care, as it increases the distance between the root nodes and the leaf node, it may dilute the sensitivity and increase the uncertainty of the network (Cain, 2001; S. H. Chen & Pollino, 2012).

The complexity of a BN can be measured by the total number of nodes that are included, and by the depth of the model. The depth is determined by the number of layers of nodes. A guideline is given to keep the number of layers to four or fewer (Marcot et al., 2006). If this is not possible, it could be considered to divide the

BN into multiple shallow sub-models. The output of one BN could be used as input for the next BN.

Determining the structure of a BN is one of the main challenges when constructing a BN. As the number of variables in a BN grows, the number of possible structures grows even faster, see Table 2.2. For a BN with five nodes or more, it becomes unfeasible to try all possible structures to find an optimal structure. So instead of checking all possibilities, the structure can be determined algorithmically from data, by experts, or by a combination of the two. As the construction of BNs is not the main focus of this thesis, only a brief overview is given here.

Table 2.2: The number of possible Bayesian Network structures as a function of the number of variables n, given by the recursive formula $b(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} b(n-k)$ for $n > 0$ and $b(0) = 1$ (Robinson, 1976).

n	number of possible Bayesian network structures
1	1
2	3
3	25
4	543
5	29281
6	3871503
7	1138779265
8	783702329343

When data is available, many different algorithms can be used to find BN structures. A recent survey of BN construction methods by Kitson et al., 2023 contains reviews of over 60 structure learning algorithms. Otherwise, when data is not available, or existing data is not sufficient, the structure can be determined with the help of domain experts. This is generally not a linear process, instead, there are feedback loops between developing stages, to gradually define the structure of the network. Again, there is not one protocol for this, for example, Burgman et al., 2021, proposes guidelines to elicit structures from experts. It is also possible to combine expert input and structure learning algorithms (Kitson et al., 2023). For example, by first having experts provide priors on certain variables, which can be the presence or absence of an arc between two nodes for instance. Then an algorithm can be used to find the optimal structure given a set of restrictions based on the expert-elicited priors.

2.3.5. EXAMPLE OF A STATIC DISCRETE BAYESIAN NETWORK

To demonstrate the theory in this chapter, an example BN is given in Figure 2.6. This BN will be further used throughout this thesis to illustrate other examples. The example BN models the atmosphere of a party, depending on the quality of entertainment (X_1), the availability of food and drinks (X_2), and the number of people present (X_3). Each of the parent nodes has three states: High, Moderate, and Low. The child node Y also has three states, the party atmosphere can either be Amazing, Okay, or Dull. This means all node states are ordered, for this example it is assumed that the parent state High is most favorable for the child node to be in the state Amazing.

Note that, for this toy example, the descriptions of the states are left vague. In proper applications of BNs it is important to relate values to the states. For example, the state *High* of node X_3 could be defined as at least 100 people, *Moderate* could then be defined as 30-100 people, and *Low* as less than 30 people. These definitions of states can be made by a modeler, during the construction of the BN, or they can be determined by experts during the parameterization. Regarding the independence in the model, the parent nodes are independent: $X_1 \perp X_2 \perp X_3$, but not conditionally independent given the child node: $X_i \not\perp X_j | Y$ for all $i \neq j$. To specify the BN, four probability tables are necessary: $\mathbb{P}(X_i = x_i)$ for $i \in \{1,2,3\}$, to specify the marginal probabilities, and a CPT $\mathbb{P}(Y = y | X_1, X_2, X_3)$, as in Table 2.1. Using these, the joint pmf can be written as:

$$f_{X_1,X_2,X_3,Y}(x_1,x_2,x_3,y) = f_{Y|X_1,X_2,X_3}(y|x_1,x_2,x_3) \cdot f_{X_1}(x_1) \cdot f_{X_2}(x_2) \cdot f_{X_3}(x_3). \tag{2.7}$$

For the construction of the child node CPT a total of $3 \cdot 3 \cdot 3 \cdot 3 = 81$ values are needed, as there are 27 combinations of parent states, for each of which 3 child states need to be assessed.

Additionally, marginal distributions are to be specified for the parent nodes. As a default, uniform distributions can be appointed, such that each state receives a probability of occurrence of $1/s_C$.

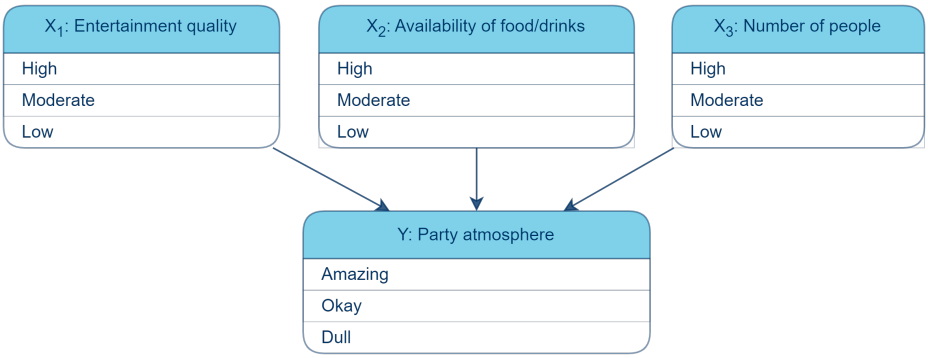


Figure 2.6: Example Bayesian Network that models the atmosphere of a party.

2.4. BAYESIAN NETWORKS IN LITERATURE

In theory, there is an unlimited number of possible applications for Bayesian networks. Possible applications range from environmental risk assessment (Kaikkonen et al., 2021), to applications in healthcare (Kyrimi et al., 2021) and cyber security (Chockalingam et al., 2017). To give a perspective on the broad spectrum of applications of Bayesian networks some of the main findings from three literature reviews are discussed.

The three reviews (Chockalingam et al., 2017; Kaikkonen et al., 2021; Kyrimi et al., 2021) cover a total of 212 papers that are in one of the following domains: Environmental Risk Assessment, healthcare, or cyber security. The papers included in the reviews were all published in the period from 2004 to 2019. Most of the papers included used static discrete BNs. For instance, 69 out of 72 papers in the Environmental Risk Assessment domain were about discrete BNs, and 70 out of 123 papers in the healthcare domain concerned static BNs.

More than 30% of the 212 papers included in the reviews involved experts for BN parameter specifications, this includes cases where expert input was used in addition to data. The use of experts varied over the domains, in a separate review of BNs in ecosystem service modeling, for 36 out of the 47 considered papers (published between 2001-2012) experts were used to fill CPTs (Landuyt et al., 2013).

Regarding the complexity of BNs, the number of nodes in the model can be compared. For the papers in the domain of ecosystem service modeling, there were 41 studies concerning one single BN. Of these 41 BNs, the average number of nodes included was 27 (std. 18) (Landuyt et al., 2013). The smallest BN consisted of 6 nodes and the biggest of 99 nodes, showing a large range of BN sizes in literature. For the smallest one, it may still be manageable to elicit the full BN parameterization from experts. For the larger BNs, this quickly becomes too much of a burden on experts, highlighting the need for ways to reduce the number of parameters that need to be elicited, in case expert judgment is needed for the full CPT specification.

3

EXPERT JUDGMENT

The second main theoretical basis that is relevant to this thesis is Structured Expert Judgment (SEJ). In most applications; such as nuclear safety, climate change, and public health; there is often insufficient data to support decisions. Either decision-making models can be simplified to be supported by the available data, or experts can be used instead. It may seem straightforward to ask experts for advice, but this is prone to mistakes due to biases when judgments are not gathered in a structured way. Therefore, SEJ can be used to limit the prevalent biases.

This chapter will start with a section on the heuristics and biases that exist for experts, which highlights the need for structured collection methods for expert judgments. Following this collection, the concept and implications of linguistic ambiguity are introduced. This chapter will further contain theory about SEJ, information on expert selection, an overview of methods that are designed to elicit probabilities, and concludes with a section about the burden on experts caused by parameter elicitation.

3.1. HEURISTICS AND BIASES

When experts are asked to estimate probabilities, they often resort to rules of thumb (heuristics) instead of mental calculations (Cooke, 1991). When these heuristics lead to estimates that are not explained by the expert's beliefs, this is called a bias. This section will contain an overview of some of the most prevalent biases in SEJ.

3.1.1. AVAILABILITY

The availability heuristic uses strength of association as a basis for judgment of frequency (Tversky & Kahneman, 1973), which can lead to biases in favor of the strongest availability. Although frequent events are often easier to imagine, other factors unrelated to frequency can influence the ease with which it is imaginable too, making the availability heuristic lead to biases. For example, when people were asked to estimate the probability of death from different causes, they often overestimated the probability for 'glamorous' cases such as a snake bite, or a tornado (Cooke, 1991). At the same time, they tend to underestimate the probability of more boring causes such as heart disease or diabetes.

To test the availability heuristic, multiple experiments were conducted. One such was the permutation experiment as shown in Figure 3.1. It was found that most subjects see more paths in A than in B, with a median of 40 paths in A and 18 in B. However, the true number of paths for both structures is equal to $8^3 = 2^9 = 512$.

This complies with the difference in availability for A and B. It may seem that there are more paths available in A due to several reasons. First, the most simple paths from the top to the bottom are straight down, A has eight of those and B only two. Furthermore, it is easier to visualize the different paths in A than in B. This is due to the paths in A being shorter, and due to the paths in A being more distinct than those in B.

3.1.2. ANCHORING

Often when people are asked for quantities or probabilities, they have a starting value in mind which they adjust slightly to give their answer. This becomes a problem when this starting value comes up unrelated to the question, for example, because it was mentioned in the question description, by an elicitor, or when other

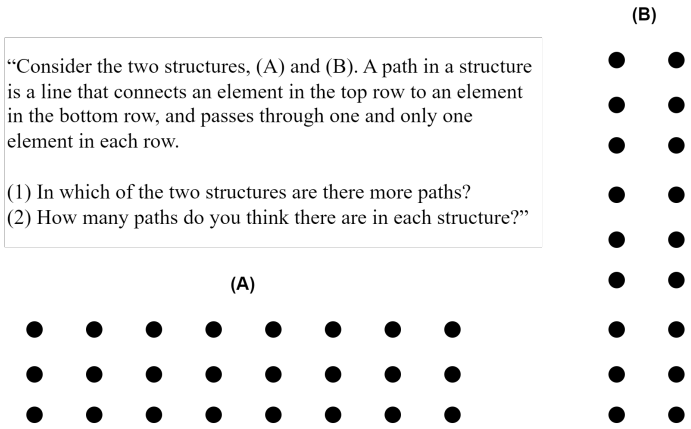


Figure 3.1: Experiment on availability based on permutations (Tversky & Kahneman, 1973).

group members give their opinion before you. This heuristic for finding answers is also called anchoring (Tversky & Kahneman, 1974), when different starting values lead to different answers, biased towards this starting value.

The starting value may be hidden in a question, such as in the following experiment. Students were split into two groups, in 5 seconds, the first group was asked to estimate product (a), and the second group estimated product (b):

$$\text{(a) } 8\cdot7\cdot6\cdot5\cdot4\cdot3\cdot2 \quad \text{and} \quad \text{(b) } 2\cdot3\cdot4\cdot5\cdot6\cdot7\cdot8.$$

It was found that the median answer for product (a) was 2250 and for (b) 512, whilst the correct answer was 40,320. Likely, the students first started calculating the first few multiplications and anchored to that value.

During expert elicitations, there are multiple ways that the anchoring heuristic can come up. When frequencies or probabilities are elicited in groups, the first response in the group can become an anchor for other members of the group to base their values on. In addition, when experts are asked for confidence regions as well as a best estimate. First asking for their best estimate can result in experts only adjusting this number slightly to find a confidence region.

3.1.3. REPRESENTATIVENESS

When probabilities are elicited, these are often about events relative to other events (Tversky & Kahneman, 1974). For instance, what is the probability that event A originates from event B? Or, what is the probability that A belongs to set B? When experts are asked to answer such questions, they often resort to using the representativeness heuristic, then the given probabilities are linked to the degree to which events resemble each other.

One of the consequences of using the representativeness heuristic can be that the effects of sample size are neglected. The following question was asked to a group of students:

"A certain town is served by two hospitals. In the larger hospital about 45 babies are born each day, and in the smaller hospital about 15 babies are born each day. As you know, about 50 percent of all babies are boys. However, the exact percentage varies from day to day. Sometimes it may be higher than 50 percent, sometimes lower.
For a period of 1 year, each hospital recorded the days on which more than 60 percent of the babies born were boys. Which hospital do you think recorded more such days?"

Students answered the question in the following way: 21 thought the larger hospital, 21 the smaller hospital, and 53 students thought both hospitals recorded about the same (less than 5 percent difference) number of such days. The correct answer is the smaller hospital, since a larger sample is less likely to deviate from the mean. The students, however, stuck to the idea that both situations are equally representative of the general population.

3.1.4. CONTROL

People tend to think that they have control over situations, even over situations that are completely determined by chance. In some situations, such as in casinos, people are given the illusion of control to give them the belief their winning chances are higher. For instance, when playing roulette players are allowed to choose a number, even though this does not influence winning chances, it may give players the feeling they have more control over the outcome of the game.

In one experiment, by Langer, 1975, the effect of choice on the illusion of control was tested using a lottery. Each subject was able to buy one ticket for the price of $1, one of the tickets was then randomly chosen to win $50. Two groups entered the lottery, and the first group (26 people) was able to choose their own ticket. The other group, consisting of 27 people, was given a ticket by the seller without choice. After the tickets were bought, each person was approached and asked for how much money they were willing to resell their ticket. The group that chose their ticket had a median reselling price of $8.67 and the other group had a median reselling price of $1.96. So apparently the group that got to choose their ticket had the illusion of control and thus attached more value to the ticket.

Although this bias might not be relevant for all expert judgment studies, there are cases in which it can become apparent. For instance, when it comes to risk assessment, man-made hazards may seem easier to control than natural hazards (Skjong & Wentworth, 2001). Which could result in people assigning a greater risk to natural disasters than man-made disasters.

3.1.5. OVERCONFIDENCE

There is a general tendency for people to be overconfident in their answers. Experiments have been undertaken to test the level of overconfidence when people are asked to provide confidence intervals. It was found that when the subjects were asked binary questions, where they had to choose between two answers and supply a percentage between 50-100% to represent their confidence level, overconfidence was modest (Klayman et al., 1999).

The overconfidence becomes more pronounced when intervals are considered. When 90% confidence intervals are elicited, the realization should fall within an expert's confidence interval 90% of the time. Another experiment found that only 43% of the time the realization fell within the confidence ranges (Klayman et al., 1999).

Another interesting result concerning overconfidence is that systematic differences in overconfidence were found between certain groups of people. For example, certain domains of knowledge are more likely to be overconfident than others, and it was found that men are generally more overconfident than women (Soll & Klayman, 2004).

3.1.6. CONFIRMATION BIAS

When looking for evidence to support one's view, people are often subject to confirmation bias, where they give more weight to evidence that supports their judgment and neglect the counter-evidence. This selectivity does not need to be deliberate, the unawareness is fundamental to the concept (Nickerson, 1998). This tendency can also refer to the failure to adequately consider counterfactual information.

In one experiment, subjects were shown a triplet of numbers and were asked to find the rule that generated them. The subjects were allowed to give other triplets to test their hypotheses, and received feedback on whether the triplets followed the rule. One example triplet was: 2 – 4 – 6. Subjects were likely to guess that the rule was *Successive even numbers*, and tested the hypothesis by giving other triplets that follow this rule. In most cases, subjects did not generate triplets which were inconsistent with their hypothesis, and thus failed to see counterfactual information. In the example, the rule could have also been *numbers increasing by 2, any three positive numbers,* or *three increasing numbers.*

3.1.7. GROUPTHINK

When people are to make decisions in a group, they are subject to social pressure. This pressure may lead group members to not voice their opinions when it is not in line with the rest of the group. Groupthink is a term that can be used when concurrence-seeking becomes so dominant in a group that it tends to override realistic appraisal of alternative courses of action (Janis, 1971). Group members fear criticizing others such that the atmosphere remains pleasant.

Apart from the softening of criticism, groupthink can also lead to ingroup bias, when there is a tendency to favor other people of the same group. This is paired with the group being hard-hearted towards outgroups.

Janis offers the main principle of groupthink in the following way: "The more amiability and esprit de corps there is among the members of a policy-making ingroup, the greater the danger that independent critical thinking will be replaced by groupthink, which is likely to result in irrational and dehumanizing actions directed against outgroups" (Janis, 1971).

3.2. LINGUISTIC AMBIGUITY

Apart from the previously described heuristics and biases, there exist other sources of mistakes. One of these is linguistic ambiguity. When expressing uncertainty, people often prefer to rely on words rather than on numbers, using terms like *possibly*, *likely*, and *almost certainly*. Words may be easier to understand than numbers and do not give the pretense that there is precision in the answer. When trying to elicit uncertainty, using words becomes very vague, because what exact probability does *likely* refer to? Although linguistic ambiguity may not exactly be a heuristic or lead to systematic biases, it is important to note its implications on elicitations.

In one experiment, graduate students were asked to relate terms of uncertainty to probabilities (Wallsten et al., 1986). They were asked to give a lower bound and an upper bound for the range of probabilities that the term could be associated with. The results are shown in Figure 3.2, for each term, the lower bars represent the range between the 25th percentile and the 75th percentile of the estimated lower and upper bounds. Notable

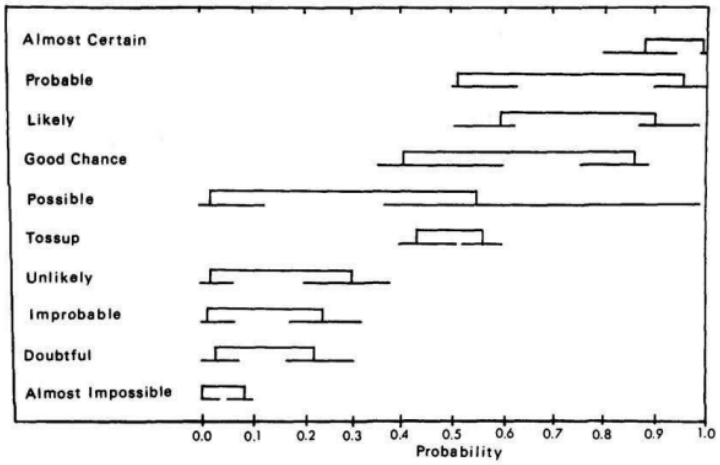


Figure 3.2: Related probabilities for words that describe uncertainty (Wallsten et al., 1986).

is that all of the ranges are fairly wide, even the range for a *tossup* ranges from 0.4 to 0.6. Even though this should be equivalent to the toss of a coin, which has a probability of 0.5. In addition, the range for the word *possible* spans almost the entire probabilistic interval $[0,1]$.

So during elicitations, the translation between terms of uncertainty and probabilities remains vague. Since the translations are personal, a term may refer to a wide range of probabilities. Therefore, such words should be used carefully.

FREQUENCIES VERSUS PROBABILITIES

Relating to people often finding it easier to understand words than probabilities, is the overall lack of understanding probabilities. In many cases, there are ways of representing probabilities in simpler terms. For example, a single event probability such as: "You have a 30% chance of a side effect from this drug" may be better represented as the following frequency: "Three out of every 10 patients have a side effect from this drug" (Gigerenzer & Edwards, 2003). Similar techniques can be used to represent conditional probabilities and relative risks.

3.3. STRUCTURED EXPERT JUDGMENT

Many different models can be used for expert judgment, these models can partly be distinguished by the level of interaction and aggregation between experts. Roughly speaking, there are two main approaches for aggregating experts' assessments. The first is based on mathematical aggregation, in that case, experts answer

questions individually and contact between experts is prevented (as far as that is possible). The second approach uses behavioral aggregation, where the experts have to come to one shared assessment. There exists a spectrum between these two extremes, on which most of the following SEJ models lie. In this section, some of the most well-known SEJ models are explained. The order in which the methods are presented is roughly following a chronological order, starting with the Delphi method and finishing with the IDEA protocol. In between, Bayesian methods, the Classical Model and SHELF are described. The section concludes with a small overview of some remaining methods.

3.3.1. THE DELPHI METHOD

One of the first methods developed for structured expert judgment is the Delphi method, which was developed by the RAND cooperation in the 1950s (Cooke, 1991). The method is based on the idea that "two heads are better than one", which can be extended to include that "n heads are better than one" (Dalkey et al., 1969). Three main features are the anonymity of response, iterative and controlled feedback, and statistical aggregation of the group response. There are many versions of the Delphi method, in this thesis the description of the method will be based on Brown, 1968.

Prior to the elicitation process, a suitable group of experts is to be found. Although there are no strict rules for the formation of a good expert panel, there is a list of guidelines for the panel, as suggested by Rowe and Wright, 2001:

1. Use experts with appropriate domain knowledge.

2. Use heterogeneous experts.

3. Use between 5 and 20 experts.

Appropriate domain knowledge is necessary for the experts to have sufficient confidence in their own judgments and those of the other experts. A heterogeneous group of experts should ensure that the full scope of the problem domain is reflected in the knowledge of the experts. Finally, there is no perfect panel size. The larger a panel becomes, the greater the administrative burden, but to adhere to the "n heads are better than one" idea, the panel also should not get too small.

After a group of experts is selected, the method starts with an initial round of questions. For each question, experts are asked to give their judgment, as well as a relative competency score, individually. The competency score is a relative score to the whole set of questions, for example, a number between one and four can be used. The individual judgments are collected and summarized by calculating the quartiles (or other percentiles, if wanted).

In the second question round, the experts receive the summarized judgments from the first round and may adjust their initial answers. If they choose to do so, they are also asked to justify their decision by stating factors that influenced their judgment.

In the following rounds, the experts will once again receive the summarized judgments, but now, also the supporting arguments from the other experts are given. With this new information, the experts will have the chance to adjust their answers another time. This should, once again, be supported by arguments about why their judgment has changed.

The process stops when a predefined stopping criterion is reached. This can be a previously agreed upon criterion, such as a fixed number of rounds, a reached consensus, or when the results stabilize (Grime & Wright, 2016). Reaching a consensus might not be very straightforward when feedback is given between rounds. One study found that the likelihood of a change of opinion is influenced by the feedback they receive (Barrios et al., 2021). When the feedback indicates that more than 75% of a group agrees, participants tend to shift their opinion towards the group opinion, otherwise, when less than 75% of he group agrees, this shift is away from the majority opinion. Making group consensus hard to reach when there is less agreement between experts to begin with.

3.3.2. BAYESIAN METHODS

After the Delphi method was developed, Bayesian methods for aggregating expert judgments were proposed. There are many different methods proposed with the same main idea: the decision maker proposes a distribution that will serve as a prior, and experts' judgments are used as observations that update the prior. A few of these methods will be introduced here, partly based on the descriptions in Cooke, 1991.

One of the Bayesian methods, by Mosleh and Apostolakis, 1986, directly uses Bayes' theorem to update the decision maker's prior, denoted by $\mathbb{P}(x)$. Let experts $1,...,E$ give estimates $X_1,...,X_E$, which form a vector of observations $\mathbf{X} = (X_1,...,X_E)$. Then, Bayes' theorem gives that:

$$\mathbb{P}(x|\mathbf{X}) = k \cdot \mathbb{P}(\mathbf{X}|x) \cdot \mathbb{P}(x), \tag{3.1}$$

where k is a normalization term. So, in addition to specifying the decision maker's prior, also the dependence between experts and the experts' errors need to be specified. In case the experts are assumed to be independent:

$$\mathbb{P}(\mathbf{X}|x) = \prod_{i=1}^{E} \mathbb{P}(X_i|x).$$

One error model that is proposed is the additive error model, in which the experts' assessments are treated as the sum of two terms: $X_i = x + e_i$. It is assumed that the error term is normally distributed $e_i \sim N(\mu_i, \sigma_i^2)$. The parameters of this error term are chosen by the decision maker, scoring the expert's bias and accuracy. If μ is zero, the expert is thought not to be biased. When $\mu_i > 0$ or $\mu_i < 0$, the expert is thought to be either negatively or positively biased, respectively. The other parameter σ_i is the perceived standard deviation of the expert's assessments. So this means that: $\mathbb{P}(X_i|x) \sim N(x + \mu_i, \sigma_i^2)$.

There is also the multiplicative error model, where the experts' assessments are treated as: $X_i = x \cdot e_i$. Note that, when taking the logarithm on both sides, this reduces to the additive error model for observations $\ln(X_i)$.

If the experts' assessments are not assumed to be independent, a joint normal distribution can be specified instead. In this case, also correlation coefficients need to be chosen between experts.

Another Bayesian method was proposed by Morris, 1977, where once again the experts' assessments are assumed to follow a normal distribution. Therefore, their probability distributions can be described by the mean and standard deviation $D = (\mu, \sigma)$. Like the previous method, the decision maker's prior density is to be updated, using Bayes' theorem:

$$\mathbb{P}(x|\mu,\sigma) = \frac{\mathbb{P}(\mu,\sigma|x)\mathbb{P}(x)}{\mathbb{P}(\mu,\sigma)}, \tag{3.2}$$

where $\mathbb{P}(\mu,\sigma)$ can be absorbed into a normalization term since it does not depend on x. The method relies heavily on the assumption of scale invariance:

$$\mathbb{P}(\mu,\sigma|x) = \mathbb{P}(\mu|\sigma,x)\mathbb{P}(\sigma|x) = \mathbb{P}(\mu|\sigma,x)\mathbb{P}(\sigma),$$

and the assumption of shift-invariance:

$$\mathbb{P}(\phi|x,\sigma) = \mathbb{P}(\phi|\sigma), \quad \text{where} \quad \phi(\mu, X) = F(X|\mu,\sigma),$$

where ϕ is a performance indicator function. This indicator is used to define a performance function $\Phi(r)$ as the decision maker's probability density that the true value realizes the rth quantile of the expert's distribution:

$$\Phi(r) := \mathbb{P}(\phi = r|\sigma).$$

Thus, the performance function takes into account the decision maker's view of the expert. Finally, the posterior for the decision maker, based on the assessment of one expert, is:

$$\mathbb{P}(x|\mu,\sigma) = k \cdot \Phi(F(x)) \cdot f(x|\mu,\sigma) \cdot \mathbb{P}(x),$$

where k is the normalization term. This can be extended to multiple experts' assessments $F_1,...,F_n$ as follows:

$$\mathbb{P}(x|\mu,\sigma) = k \cdot \Phi(F_1(x),...,F_n(x)) \cdot f_1(x|\mu_1,\sigma_1) \cdot ... \cdot f_n(x|\mu_n,\sigma_n) \cdot \mathbb{P}(x),$$

where $\Phi(F_1(x),...,F_n(x)) = \Phi(F_1(x)) \cdot ... \cdot \Phi(F_n(x))$ in case experts' assessments are assumed to be independent.

More recently, methods have been developed to deal with common issues regarding the previously introduced Bayesian methods. One such issue is regarding overconfidence in the posterior distribution, which can come from correlations between experts (Hartley & French, 2021). Finding such correlations, and constructing correlation matrices from this, is a challenge. One proposed method to overcome this challenge is to cluster groups of experts that share knowledge (Albert et al., 2012; Billari et al., 2014), where expert evaluations within the same cluster are assumed to be independently generated from the same distribution. This allows the modeler to account for dependencies between experts without direct implementation.

3.3.3. THE CLASSICAL MODEL

In this chapter, The Classical Model will be discussed based on the book *Experts in uncertainty* (Cooke, 1991). The model offers a structured way to aggregate individual experts' assessments into "decision makers" (i.e., aggregations of expert judgements which can be considered to belong to a hypotetical decision maker). The assessed variables can have both a continuous or a discrete range of possible values. When continuous variables are assessed, such as frequencies of events, quantiles are elicited from experts. Experts complete individual elicitations, where they answer both the questions of interest and calibration questions (i.e., questions whose answers are known with certainty by the analyst but not by the expert). Based on the calibration questions, experts are scored by a combination of a calibration score and an information score. These scores are then used to give each expert a weight that determines their influence in the final aggregation of the assessments.

The scores that are used in the Classical model are so-called proper scoring rules. A scoring rule is called proper when an expert receives a maximal score if and only if they state their true opinion. A scoring rule becomes improper when the scoring reward system encourages experts to state opinions that are different from their true opinions.

Let $e \in \{1, ..., E\}$ be an expert in the total set of experts, then the unnormalized weight of this expert is determined to be:

$$w_e = C(e) \cdot I(e) \cdot \mathbb{1}_\alpha(C(e)). \tag{3.3}$$

This weight can then be normalized by $W = \sum_{i=1}^{E} w_i$, to get the normalized global weight for the expert:

$$\frac{w_e}{W} = \frac{C(e) \cdot I(e) \cdot \mathbb{1}_\alpha(C(e))}{\sum_{i=1}^{E} C(i) \cdot I(i) \cdot \mathbb{1}_\alpha(C(i))}, \tag{3.4}$$

where:

- $C(e)$ is the calibration score,
- $I(e)$ is the information score,
- $\mathbb{1}_\alpha(C(e))$ is an indicator function which is one when the calibration score is larger than some threshold α (the significance level), otherwise it is zero.

In the next sections, the calibration and information score will be defined, and how the weights are used to construct decision makers is discussed in detail. But first, it is explained how the expert's probability distributions are determined.

EXPERTS' PROBABILITY DISTRIBUTIONS

During the elicitation process, experts are asked to give answers to questions by giving a 5th percentile, a 50th percentile, and a 95th percentile. For each question i these are gathered into a tuple $(q_5^e, q_{50}^e, q_{95}^e)$ for expert e. To form a probability distribution, first, a support is needed. In probability theory, the support is the set of possible values that a random variable can take. In this study, the intrinsic range will be used as the support, which is defined for an individual question as:

$$[L^*, U^*] = [L - k(U - L), U + k(U - L)],$$

where $L = \min\{q_5^1, ..., q_5^E, \text{realization}\}$ and $U = \min\{q_{95}^1, ..., q_{95}^E, \text{realization}\}$. The parameter k is the level of overshoot, which is generally taken to be $k = 0.1$. The intrinsic range can be described as the range between the smallest value and the largest value with a 10% overshoot on both sides. For the calibration questions, the realization is also considered in this calculation, but for the questions of interest, it does not play a role.

Finally, by interpolating between the points $(L^*, q_5^e, q_{50}^e, q_{95}^e, U^*)$ a cumulative probability distribution can be found. Thus for each question, each expert forms their own distribution. This will form four inter-quantile ranges: Q_1, Q_2, Q_3, Q_4, the first being between L^* and q_5^e, the second between q_5^e and q_{50}^e, and so on.

CALIBRATION SCORE

The first score to be discussed is the calibration score $C(e)$, this score measures the accuracy of the experts. The measure is based on how often an expert can capture the real value within their 90% confidence interval. For each expert, the empirical distribution vector $s(e) = (\frac{s_1}{m}, \frac{s_2}{m}, \frac{s_3}{m}, \frac{s_4}{m})$ is found, where each s_i is the number of realizations within the ith inter-quantile range Q_i, and m is the number of calibration questions. Each empirical distribution vector s gives rise to a distribution S. A perfectly calibrated expert would have an empirical distribution vector $s(e) = p = (0.05, 0.45, 0.45, 0.05)$. This vector gives rise to distribution P, corresponding to the realization falling within the 90% confidence interval exactly 90% of the time, while also being symmetrical around the 50th percentile. To determine how similar the empirical distribution vector $s(e)$ of an expert is to the vector p, the relative information score is used:

$$I(s,p) = \sum_{i=1}^{4} s_i \ln(\frac{s_i}{p_i}). \tag{3.5}$$

One may consider $I(s,p)$ as a measure of surprise if they believe p, but afterward learned s is the truth. The larger the value, the greater the surprise. This measure is used to define the calibration score, but first, it should be determined when an expert is calibrated well. The statement: *the expert is well-calibrated* is interpreted as the following statistical hypothesis:

$$Cal(P) := \textit{the uncertain quantiles are independent and identically distributed with distribution } P.$$

The calibration score is then defined as the probability under $Cal(P)$ of observing a discrepancy in a sample distribution S' at least as large as $I(S,P)$, on n observations:

$$\mathbb{P}(I(S',P) \leq I(S,P) | Cal(P), n \text{ observations}). \tag{3.6}$$

Thus, the calibration score measures the degree to which the data supports the statistical hypothesis $Cal(P)$.

For a finite number n, $2 \cdot m \cdot I(S,P)$ converges to a χ^2 distribution with $n-1$ degrees of freedom, when m gets large. Where the number of observations n is equal to the length of the empirical distribution vector s, and the parameter m is the number of calibration questions, so $n = 4$ and $m = 12$. Thus, the equation which is used to calculate the calibration score of an expert is determined to be:

$$C(e) = 1 - F_{\chi^2_{n-1}}(2 \cdot m \cdot I(s,p)). \tag{3.7}$$

INFORMATION SCORE

The second score that is important in expert performance analysis, is the information score. This score measures how informative each expert is, by defining a measure on the size of the confidence interval given by each expert on each question. Relating this to the probability mass function, this can be viewed as the degree to which the mass is concentrated.

The information score is relative to the background measure, which is chosen to be uniform on the interval $[L^*, U^*]$ for each question. Other measures could also have been considered, such as a log uniform distribution. This distribution concentrates the mass on the ends of the interval, instead of distributing the mass evenly over the interval. It is suggested to only use this measure when the expert's assessments span over more than four orders of magnitude for a single question.

Using the uniform distribution as a background measure, for $n = 4$, the relative information score is defined as:

$$I(e) = p_1 \cdot \ln\left(\frac{p_1}{q_5 - L^*}\right) + p_2 \cdot \ln\left(\frac{p_2}{q_{50} - q_5}\right) + p_3 \cdot \ln\left(\frac{p_3}{q_{95} - q_{50}}\right) + p_4 \cdot \ln\left(\frac{p_4}{U^* - q_{95}}\right) + \ln(U^* - L^*),$$

for each question. This means that the mean relative information score is given by:

$$I(e) = \frac{1}{n} \sum_{i=1}^{n} I(e), \tag{3.8}$$

for each expert. Note that the relative information score can be calculated for both the calibration questions and the questions of interest. The mean relative information score is averaged over only the calibration questions, as the calibration score is only calculated on those too.

The information score is a monotone function, the smaller the distance between the 5th percentile and the 95th percentile, the higher the information score. It is not necessarily a positive feature if the distance between the elicited percentiles is very small, thus it is important to note that performance should never be scored solely on informativeness.

DECISION MAKERS

After each expert's performance is assessed, the information and calibration scores can be used to determine the experts' weights as in Equation (3.4). These weights can then be used to create a so-called decision maker. A decision maker is a virtual expert created by aggregating the experts' assessments, using weights.

Two types of decision makers will be considered in this report. The most straightforward one is called the Equal Weights Decision Maker (EWDM), which gives each expert the same weight. Secondly, there is the Performance-based Weights Decision Maker (PWDM), which uses Equation (3.4) to determine the weight of each expert. When the experts' assessments are aggregated, a weight can also be calculated for the decision maker. The PWDM's unnormalized weight is a function of α and can be optimized by maximizing it over α. Thus the optimal α is then defined to be:

$$\alpha' = \text{argmax}_{\alpha \in (0,1)} \, w_{DM}(\alpha). \tag{3.9}$$

Finally, the aggregated assessments of the decision makers can be used to give answers to the questions of interest. Therefore, for each question, the distribution of the decision maker is determined to be:

$$P_{DM} = \frac{\sum_{e=1}^{E} w_e P_e}{W}.$$

This results in a probability distribution for each question, which can then be used to give answers to the questions of interest, by once again providing a 5th percentile, a 50th percentile, and a 95th percentile.

3.3.4. THE SHEFFIELD ELICITATION FRAMEWORK

The SHeffield ELicitation Framework (SHELF) is a method that uses behavioral aggregation, the method is based on elicitation practices described by O'Hagan et al., 2006. This section is based on the description of SHELF in the book: *'Elicitation: The Science and Art of Structuring Judgement'* (Gosling, 2018). Through the use of facilitated group discussions, the goal is for the experts to come to a consensus. The elicitor is recommended to have sufficient knowledge on probability and statistics and should have experience in managing meetings.

The SHELF method can be divided in eight stages. The first two are individual, then six group stages follow:

1. exercise specification,
2. expert selection,
3. training about elicitation process,
4. information sharing,
5. individual judgments,
6. distribution fitting,
7. discussion and aggregation,
8. feedback on distribution → loop through further discussion to end with a satisfactory consensus.

The first two steps, which are colored light blue, take place individually. The following six steps, dark blue colored, take place in a group meeting with an elicitor. The final step can be looped over until a satisfactory consensus is found.

In the first step, the problem owner must specify the quantities of interest to avoid ambiguities and further specify the knowledge that is to be elicited. The next step is to select experts to participate in the exercise. For both of these steps, there are no set instructions, however, from past applications, there are guidelines that can be followed. For instance, it was found that a group of five to ten experts is good (EFSA, 2014). Enough for a range of perspectives, but not too many such that the discussion stays manageable.

After the exercise is set, and the experts are gathered, the next step is to train the experts about the elicitation process. Experts do not necessarily have enough knowledge of probability theory to express their assessments by using probabilities. Therefore a practice exercise can be used to guide the experts through steps 5-8 in the process.

In the fourth step, a series of questions is used to gather information about the experts. This information is useful for both the decision maker and the other experts in the group, to have a functioning group and transparency. For the values of interest, it is asked whether the experts have any related interests, what their expertise is, what important factors are for making judgments, what evidence they have seen, and is they have structured the values of interest in terms of other quantities.

During the fifth step, whilst the quantity of interest is being discussed, including found evidence in the previous step, the experts make individual assessments, which are not immediately shared within the group. Although the SHELF method is flexible to support different types of assessments, one method is the quartile method. First, the upper and lower bounds are asked, this is chosen as an attempt to prevent anchoring and overconfidence. Then, the median is asked, and finally the lower and upper quartile. To guide the experts, visualizing elicitation methods can be used, an overview of elicitation methods is given in Section 3.5.

In step six, a distribution is fit to the experts' judgments. This is generally done by minimizing the least squares error. In case the quartile method is used for elicitation, the cumulative density function (cdf) F that minimizes:

$$F(Q_0)^2 + [F(Q_4) - 1]^2 + \sum_{i=1}^{4} \left([F(Q_i) - F(Q_{i-1})] - \frac{1}{4} \right)^2, \tag{3.10}$$

is to be found. There, Q_0 and Q_4 are the lower and upper bound respectively and the other Q_i are the remaining elicited quartiles. In this step, the elicitor must use their probability/statistics knowledge to choose a distribution that fits the experts' judgments, since many distributions will minimize Equation 3.10.

After distributions are fit to individual judgments, in the seventh step, the distributions are aggregated. This aggregation mainly functions as a guide to the elicitor and may or may not be shared with the experts in the group. The aggregated distribution can be used to identify experts with extreme judgments. One method for aggregation is the linear opinion pool, in which the N experts' distributions f_i are averaged to form f_A:

$$f_A(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x).$$

The final step of the process is the feedback stage. The experts' distributions are shared in the group which prompts a discussion, that should lead to a consensus. When judgments are gathered using the quartile method, the goal is to find three quartiles that correctly portray the experts' beliefs.

3.3.5. THE IDEA PROTOCOL

The IDEA protocol (Hanea et al., 2017), which is an acronym for the steps: Investigate, Discuss, Estimate, and Aggregate, is a protocol that uses components of previously described methods. The components are chosen carefully to optimize the advantages of previously existing methods and minimize their disadvantages. The elicitation process takes place in iterations and experts are supplied with anonymous feedback, similar to the Delphi method. However, unlike the Delphi method, there is also room for face-to-face discussions, and a consensus is not sought. The IDEA protocol allows for expert performance measures to be integrated as in the classical method.

Similar to other methods, the process starts with a pre-elicitation phase. During this phase, the elicitation is prepared and experts are selected. Again, there are no set rules for selecting experts, but the main selection criterion is that the experts must understand the questions that will be asked (Hemming et al., 2018). It is also suggested that a group size of 10-20 experts is aimed for.

- **Investigate:** A first elicitation round takes place, where experts provide their judgments individually. There are two constructions for eliciting quantities or probabilities, which will be elaborated on in Section 3.5. The received individual judgments will be analyzed and a graphical output will be created, which includes the individual judgments (anonymized) as well as the group aggregate.
- **Discuss:** During this phase, the results from the Investigation stage are shared among the experts and discussed. An elicitor is there to guide the experts in the discussion.
- **Estimate:** After the discussion has come to an end, the experts have the chance to revise their judgments. The judgments are kept anonymous, but linked to the previous judgments from the first elicitation round.
- **Aggregate:** Finally, the individual judgments are aggregated. The form of aggregation can be chosen specific to the application. Most often, although not recommended, quantile aggregation with the arithmetic mean is used.

3.3.6. OTHERS

There are still several methods left undiscussed, in this section a quick overview will be given about the wisdom of crowds, the Bayesian truth serum, and superforecasters. These methods are not discussed in detail, only the main concepts are discussed.

Perhaps the most simple method discussed in this thesis is the wisdom of crowds, described in the book *The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nation* (Surowiecki, 2004). According to the book, each person's answer consists of information and error. Averaging all individuals' answers will eliminate those errors. Thus, making the collective opinion better than any of the individual's opinions. An application of the Wisdom of Crowds method would be similar to performing one round of the Delphi method.

The focus of the book is therefore on the selection of an appropriate crowd, a wise crowd that also acts like one, as this part is deemed to be the most challenging. There are five aspects to a crowd that are sought after: diversity, independence, decentralization, aggregation, and trust.

The Bayesian truth serum is a method to elicit subjective data, when the objective truth is not known (Prelec, 2004). The method works by asking questions in pairs, one about their own opinion and one general question. For example, when the question of interest is "What percentage of the others rates Picasso as their favorite painter?". The paired question is "Is Picasso your favorite 20th-century painter?". The idea is that a person's opinion influences their view of other people's opinions, as their opinion serves as a data point in the total sample.

The method works by scoring the answers, based on how common they are in the sample against the collectively predicted opinion. A high score is rewarded for answers that are more common in the sample than was predicted. The method has seen a decrease in popularity, although it might have found a new application as "Machine Truth Serum" to classification problems (Luo & Liu, 2023).

The Good Judgment Project (GJP) was one of the contestants in a four-year-long prediction contest. GJP invited thousands of people to answer geopolitical and economic questions as volunteers (Tetlock & Gardner, 2016). The goal of the project was to determine whether there are people naturally better at forecasting than others, and if prediction performance can be enhanced. When the best performing "amateur" forecasters were selected and collected into a group of superforecasters, their predictions were nearly twice as accurate as those made by untrained forecasters (Schoemaker & Tetlock, 2016). Once again, the group composition was found to be an important factor. Contrary to other methods, where often only domain experts are selected, for a group of superforecasters not only a domain expert is needed, but also non-experts are important to challenge the domain experts.

3.4. EXPERT SELECTION

There are no strict rules for selecting experts to participate in an SEJ study, however, experts should not be chosen randomly either. This section will discuss some guidelines for expert selection, such as knowledge, diversity, and group size.

The descriptions of structured expert judgment methods also included some guidelines for selecting experts. The Delphi method called for experts with appropriate domain knowledge, whilst the Good Judgment Project also requires non-domain experts to be included. One of the most important requirements for an expert, with or without domain expertise, is that they understand the questions being asked. Since experts are not able to express their true beliefs if they cannot understand what is asked exactly. When CPTs are elicited for BNs, it is beneficial if experts have some understanding of probability theory and/or have seen BNs before.

Furthermore, a diverse set of experts, to ensure cognitive diversity, is generally wanted. Ensuring cognitive diversity is not a simple task, therefore diversity could be searched within proxies, such as cultural background, education, age, and gender. Identity characteristics do not matter in terms of cognitive ability, but do matter when finding a diverse set of minds. *People belonging to different identity groups pull from different wells of experience* (Page, 2008).

Recommendations for the number of experts to include in an SEJ study also differ between methods, but the recommendations reported in this thesis all supported that at least 5 experts are included, if attainable. The upper limit for the group size varies, from 10 for SHELF, to thousands of people as an initial pool for superforecasting. Especially when group discussions take place, care should be taken to not include too many experts, to keep discussions manageable. More often, the struggle lies in finding enough experts, than

in ensuring not too many experts are selected, as experts are often not available in great quantities.

Finally, experts should be willing to participate in the study and be available during the study. Especially when a large number of quantities needs to be elicited, experts must have sufficient time and motivation to finish the elicitations.

3.5. ELICITATION FORMATS

Simultaneously to the development of structured expert judgment methods, elicitation formats were being studied to best elicit quantities and probabilities. The goal of these formats is to minimize heuristics and biases, as introduced in Section 3.1, and confusion of the experts.

THREE-/FOUR-STEP INTERVAL ELICITATION FORMATS

During elicitations, experts are often not only asked about their best estimate for a quantity or probability, but also their uncertainty about it. There are multiple ways to ask experts about their confidence, either a confidence level is set and experts are asked to fit their uncertainty bounds to this, or the experts are first asked to give bounds to the answer and later determine their confidence level. The three-step and four-step interval elicitation format, as they are used for the IDEA protocol (Hanea et al., 2017), are shown in Figure 3.3. The three-step method can be used for eliciting probabilities, where the domain is bounded between zero and one, and it does not ask experts to specify their confidence level. The four-step method does ask experts to assess their uncertainty with a percentage.

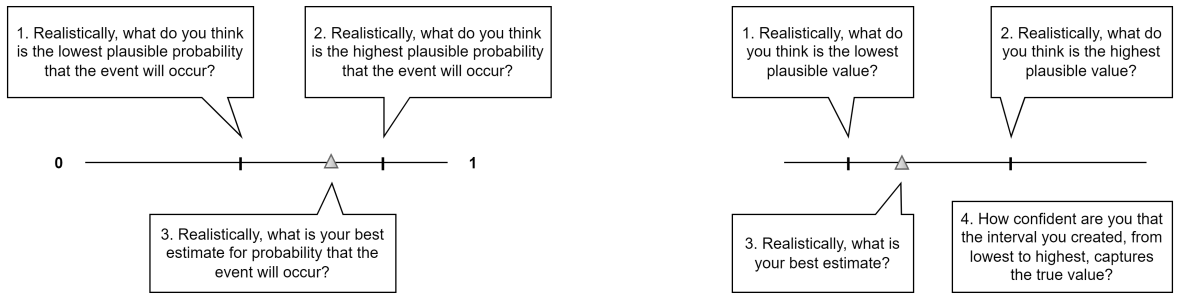


Figure 3.3: Three-step and four-step interval elicitation format for probabilities and quantities, respectively (Hanea et al., 2017).

In earlier studies, it was found that using the four-step format over the three-step format can reduce the overconfidence of experts (Speirs-Bridge et al., 2010). After this study, there was no more research done to prove that this initial reduction remained after aggregation. Since the four-step method gives experts the freedom to choose quantiles, the experts' distributions first need to be approximated before they can be aggregated, which means that more assumptions need to be made about the distribution for the four-step method than for the three-step method.

Another important aspect of the elicitation format is the order in which the questions are asked. In an attempt to reduce the effects of anchoring, the upper and lower bounds are asked before asking the best estimate. This way the expert is less likely to only alter the best estimate slightly to find bounds.

One disadvantage of eliciting intervals, and using such methods, is that they form a burden on the experts. Feedback on the three- and four-step methods included the repetitive nature of the formats, which can lead to fatigue after responding to a series of such questions. A rule of thumb is then given that it is optimal that between 8 and 12 estimates are to be elicited in one sitting (Speirs-Bridge et al., 2010). When many estimates are needed, as is often the case when CPTs are elicited, this rule of thumb is often neglected due to a lack of time. For a "small" BN as in the example in Figure 2.6, this would mean the elicitation of the full CPT would already take at least 7 days.

PARAPHRASING

When experts are less knowledgeable about probabilities, it may be helpful to rephrase probabilities in terms that the experts do understand, especially when conditional probabilities are at hand. One way of rephrasing is by asking for frequencies instead of probabilities. For example, when the question is: "Consider a service being hit with a speed of over 200 km/h, what is the probability that this service results in an ace?" can be rephrased using frequencies in the following way: "Imagine 100 services being hit with a speed over 200 km/h, how many of these will result in an ace?"

One drawback of using frequencies can be that these are hard to imagine by experts. In that case, the question could also be rephrased using likelihoods. The example would then become: "Consider a service being hit with a speed of over 200 km/h, how likely is it that this service results in an ace?" For this particular example, this may not be necessary, but in other domains, it is more applicable. For example, when low-incidence diseases are considered (Renooij, 2001).

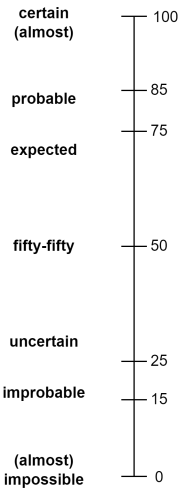


Figure 3.4: Scale with verbal anchors (Renooij & Witteman, 1999).

SCALES

In addition to rephrasing the question, there are also visual guides that can help experts to answer probability questions. One such visual tool is a scale, such as shown in Figure 3.4. This example does not only show a scale with numerical guide numbers but also verbal anchors are given. These are given to help experts understand the probabilities. These verbal anchors should be used carefully though, as previously mentioned in Section 3.2, such words have different associations for different people. For example, the related probabilities for *probable* are between 0.5 and 1 in Figure 3.2, but in the scale it is placed around 80-85%.

ROULETTE METHOD

The roulette method is a popular method within SHELF and has been found most applicable when experts have a solid background in statistics. The method needs to have a defined range for the plausible answers to start, which means that experts may need to agree on this first. When a range has been set, it is divided into a number of equally sized subintervals which will form "bins". The number of bins can be chosen to represent the accuracy as needed. The experts are then given a set number of "chips" per person, which they can place into the bins. Placing the chips in bins resembles placing bets on a roulette table, placing more chips in bins where they think the probability of the realization falling is high.

It is recommended that the experts are given no more than thirty chips and that the chips are physical, such that the experts have the chance to engage with the chips (Gosling, 2018). When elicitations take place in a group, it should be made sure that experts can distribute their chips privately, without seeing the chips of other experts.

GAMBLE-LIKE TOOL

Another type of elicitation tools for probabilities is using a gamble-like method. Instead of asking experts for their probability assessments directly, they are asked for their choice between lotteries. The expert may choose between two lotteries, in the first the expert has a probability to win a grand prize equal to the probability of interest. For the second lottery, the chance of winning the prize is set by the elicitor. The value in the second lottery is varied until the expert's preference is indifferent between the two lotteries.

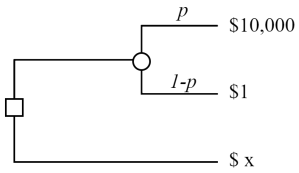
In one version of the tool, the expert can choose between entering a lottery where they have a probability p of winning \$10,000, or immediately receiving \$$x$. The corresponding decision tree is given in Figure 3.5a. The probability p is the probability that is assessed, and the value x is varied until the expert can't choose between entering the lottery or not. The probability is then calculated by:

$$\hat{x} = 10,000 \cdot p + 1 \cdot (1 - p),$$

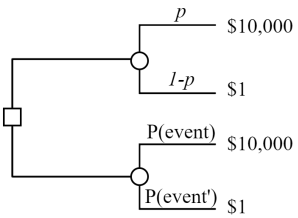
where $\hat{x}$ is the value of x for which the process stopped. The main drawback of this method is that the risk attitude of experts influences their choices greatly (Renooij, 2001). An expert who likes to take risks may stop at a value of x that is greater than the value of x of an expert who does not like to take risks, even though they would otherwise agree about the value of p.

For another version of the gamble-like tool, the expert has the choice between entering two different lotteries. In both lotteries, the expert can win these same prizes, one with a small value and one with a big value. For one of the lotteries, the elicitor varies the probability of winning the big prize p. For the other lottery, the probability of winning is equal to the to-be-elicited probability, denoted by $P(event)$. The probability of winning the small prize is equal to the probability of the event not happening $P(event')$. Again, the process stops when the expert is indifferent between the choices. The probability of interest is then found to be: $P(event) = p$.

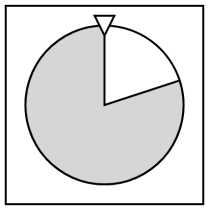
These gamble-like methods still come with a list of drawbacks, such as being time-consuming, complicated to learn, and it may be unethical in certain contexts. For example, in the medical field, when an expert could win \$10,000 if a patient dies (Gaag et al., 1999).



(a) Gamble-like.



(b) Gamble-like.



(c) Probability wheel.

Figure 3.5: Probability elicitation tools.

PROBABILITY WHEEL

The final visual tool discussed in this thesis is the probability wheel Renooij, 2001. This tool uses a wheel-of-fortune-like chart that is partly colored by the elicitor, like in Figure 3.5c. The expert is now asked to choose which of the following two options is more likely:

- the probability wheel stops on the colored part,
- the to-be-elicited event occurs.

So for the party atmosphere example in Figure 2.6, the second option could be the event that there is an amazing party atmosphere, given high entertainment quality, high availability of food and drinks, and a high number of people attending. The size of the colored part of the wheel is varied until the expert is indifferent between the two choices. The probability of interest is then equal to the fraction of the wheel which is colored.

This method is again very time-consuming, and experts may recognize that they can also give their assessment by coloring the wheel themselves. Finally, due to its purely visual nature, the wheel cannot be used for probabilities close to zero or one, as the wheel does not allow for small details to be captured.

3.5.1. APPLICATION TO CPT ELICITATIONS

Most of the methods introduced in this section are not developed for eliciting a large number of values. Especially the methods that lead to an estimate iteratively, such as gamble-like methods and the probability wheel, take a substantial amount of time for each assessment. When CPTs are elicited for BNs, often many conditional probabilities are needed, thus a method should be chosen that takes relatively little time for each assessment.

The method should also be applicable to elicit intervals, such that experts can express their uncertainty. It may be seen as not rigorous to elicit probabilities over probability assessments, as uncertainty is already captured in the probability assessment itself. To avoid eliciting second-order probabilities, relative frequencies can be elicited instead of probabilities.

From the discussed methods, it can be concluded that some form of visualization or other representation of probabilities can be helpful for experts to express their beliefs. However, after the elicitation of many probabilities, experts may not need the guidance anymore. Therefore, when experts have little understanding of probability theory, it is perhaps better to start the elicitation process with training which includes a theoretical part about probabilities, and practice questions. Afterward, experts should be able to give their assessment with only little guidelines.

The three- or four-step method can be combined with a scale with verbal anchors. These verbal anchors should be used carefully, and are likely not needed after a training on probability theory. Otherwise, the verbal anchors can also be generated with the help of the experts in the study. Finally, to ease the aggregation of experts' judgments, the three-step format is preferred over the four-step method, as density (construction and) aggregation can be applied without extra assumptions.

When other types of parameters need to be elicited, such as node influence weights, some ideas in this section can be used as well. Mainly, the idea of visualization can be applied in many shapes and forms. When an elicitation is set up, time should be spent on creating answering formats that include some form of visualization.

3.6. PARAMETER ELICITATION BURDEN

As briefly discussed in the previous section, the elicitation of quantities or parameters from experts does not come for free. The burden that the elicitation process imposes on experts can be measured by the number of parameters they need to assess. The larger the number of parameters that are needed, the more questions are asked, and thus the larger the burden on the experts. For instance, the example BN of Figure 2.6 needs one CPT to be parameterized which contains $3^4 = 81$ conditional probabilities. Adding an extra node, also with three states, would increase the number of conditional probabilities to $3^5 = 243$. Similarly, if the example BN would be made more detailed by adding an extra state to each of the nodes, the number of values in the CPT would increase to $4^4 = 256$. This shows how quickly the CPT grows when nodes or states are added to a BN.

However, it should be noted that not all parameters are equally complex to assess. For example, when a CPT is fully elicited with a child node with just two states, assessing the value for the first state is likely more burdensome than for the second state. As the probabilities should sum to 1, the second question serves more as a check than an actual elicitation. This could be extended to a child node with s states, then only the first $s-1$ probabilities are necessary to be elicited, the final probability can then be elicited for completeness and to normalize the probabilities such that they sum to 1.

Apart from eliciting probabilities, also other types of parameters can be elicited, such as weights, variance parameters, or correlation structures. How the burden differs between each type of elicited parameter is not defined clearly. It may be hypothesized that assessing probabilities is more straightforward than assessing weights for instance, as probabilities are a "natural" phenomenon and weights of parents are less intuitive.

In addition to the questions of interest, other sources of burden should be taken into account when a study is designed. For example, the method that is used for the elicitation, the number of training sessions included in the process, whether the elicitation takes place online or in person, and the amount of supplementary material the experts receive. Elaborating on the elicitation method, when surveys are used for elicitations, the way questions are posed influences the burden on the experts. Different psycholinguistic text features such as acronyms, low-frequency terms, vague quantification terms, and quantitative mental calculations should be avoided to relieve the burden (Lenzner et al., 2010). Although it may be hard to avoid using some of these terms, they should be used with care.

In this thesis, the burden of elicitation will be measured in terms of the number of parameters that are to be elicited. Although there is likely a different level of burden for different types of parameters, there is no measure that accurately describes the amount of burden for each type of parameter. Therefore, the burden of assessing each different type of parameter is assumed to be equal.

4

Construction Methods for Conditional Probability Tables using Expert Judgment

The construction of Conditional Probability Tables (CPTs) for Bayesian Networks (BNs) can be a laborious task, which depends on the complexity of the model, on the amount of data and the quality of the available data. When constructing a CPT from data, there should be sufficient data, but there are no clear rules about when data are sufficient. The minimal amount of data necessary should cover all values in a CPT. So, for each combination of parent states a set of events should be included in the data set, such that each entry of the CPT table can be determined. Each entry is a conditional probability that can be calculated for all combinations of child and parent states. Calculation methods include Gibbs sampling, Expectation Maximization, and Gradient Descent (S. H. Chen & Pollino, 2012).

Since each CPT entry could be represented by a multidimensional discrete vector, and the number of possible vectors grows exponentially with the addition of variables, the data requirements also grow exponentially as the BN grows. Not always there is sufficient data available to fully specify the probabilities, for example in volcanology there exists limited data due to the infrequent occurrence of eruptions (Christophersen et al., 2018). In that case, experts can be involved in the process, the construction then relies either solely on experts or a mix of data and experts.

When only experts are involved, the most straightforward way to construct a CPT is for experts to go over all possible combinations of parent states and give the probability of each child state occurring. In this way the full CPT is elicited. This method is only viable when the CPT is relatively small, as the number of nodes, or the number of node states grows, the number of probabilities to be elicited grows even faster, making the elicitation a large burden on the experts. For a child node X_C with s_C states and n parent nodes $X_1, ..., X_n$, where each parent node X_i has s_i states, the number of probabilities that are to be elicited is $s_C \prod_{i=1}^{n} s_i$. For the example BN given in Section 2.3.5, this means that a total of 81 values would need to be elicited.

To lighten the burden on experts, the number of values that an expert has to assess needs to decrease. One option is to change the structure of the BN, by divorcing nodes or reducing the number of node states, as described in Section 2.3.4.

Furthermore, methods have been developed to construct CPTs using a limited number of elicited parameters without changing the structure. An overview of a selection of those methods, which are introduced in this section, is given in Table 4.1. The table also states the number of parameters that need to be elicited for each method to construct a single CPT. Most methods need significantly fewer parameters to be elicited than the full CPT elicitation but come with other limitations, such as a loss of accuracy. For some of the methods an interval is given, in that case, there are multiple versions of the method, and each needs a different number of assessments. Additionally, the number of assessments that would be needed for the example BN in Section 2.3.5 is given.

This chapter will continue by discussing general assumptions that all of the CPT construction methods

Table 4.1: Overview of CPT construction methods, including the number of parameters that need to be elicited to construct a single CPT for a general BN, in addition, the number of parameters necessary for the example specified in Section 2.3.5. In case a method has multiple versions, a range is given for the number of parameters.

Construction method	Number of parameters	Example
Noisy-OR/MAX (Kim & Pearl, 1983; Pearl, 1988)	$(\sum_{i=1}^{n} s_i + 1)s_C$	30
Ranked Nodes Method* (Fenton et al., 2007)	$[3n+3, 3n+s_C+3]$	[12, 15]
InterBeta (Mascaro & Woodberry, 2022)	$[2s_C, 2m_C + \prod_{i=1}^{n} m_i]$	[6,33]
Functional interpolation method (Podofillini et al., 2014)	$2^n \cdot s_C$	24
EBBN (Wisse et al., 2008)	$4n + s_C + s_C^2$	24
Weighted Sum Algorithm (Das, 2004)	$[\max_i s_i, \sum_{i=1}^{n} s_i]$	[3,9]
Cain's method (Cain, 2001)	$s_C(2 + \sum_{i=1}^{n}(s_i - 1))$	24
Røed's method (Røed et al., 2009)	$[n+1, n+1+\sum_{i=1}^{n} s_i \cdot s_C]$	[4,31]
ACE (Hassall et al., 2019)	$2n$	6
Likelihood method (Kemp-Benedict, 2008)	$s_C + \sum_{i=1}^{n} s_i$	12
Full CPT elicitation	$s_C \cdot \prod_{i=1}^{n} s_i$	81

*It is assumed here that fitting parameters were found in one iteration.

impose on the structure of the BNs. After which extensive overviews of some of the existing methods for constructing CPTs are given. These include the full elicitation of CPTs, Noisy-OR/MAX, the Ranked Nodes Method (RNM), InterBeta, and Functional interpolation, the other methods included in Table 4.1 are introduced in Appendix A. The methods are explained, linked to SEJ, and some of the limitations of the methods are highlighted. Next, a quick overview is given of some applications of the previously mentioned methods. The chapter concludes with a section on measures for comparing the accuracy of constructed CPTs.

4.1. GENERAL ASSUMPTIONS

Each CPT construction method comes with its own set of assumptions. Some of these assumptions are method-specific, and some assumptions are shared by all methods that are presented in this thesis. The largest shared assumption is on the structure of the BN. The methods apply to BNs, or parts of BNs, that consist of one child node denoted by X_C and n (independent) parent nodes $X_1, ..., X_n$, as depicted in Figure 4.1.

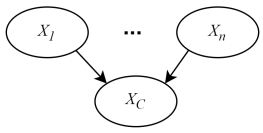


Figure 4.1: BN structure suitable for construction methods.

In addition to the assumed graphical structure of the BN, several of the methods are only applicable to BNs with **ranked nodes**. A ranked parent node can be defined as a node whose states can be ordered in terms of their influence on the child node. The highest-ranked state of the parent node shifts the mean of the child node distribution to its highest-ranked state the most, and the lowest-ranked parent node state pushes the child node distribution towards its lowest-ranked state. Thus, there should exist a positive correlation between parent and child.

Note that, when there originally exists a negative correlation between the parent and child node, the state ordering of the parent node can be reversed such that the correlation becomes positive.

4.2. FULL CPT ELICITATION

The first method for parameterizing BNs using experts is to elicit the full CPT. In that case, experts are asked to evaluate all of the probabilities included in the CPT. One by one, the experts are asked questions of the following type:

- *Given that parent node X_1 is in state x_1 and ... and parent node X_n is in state x_n, what is the probability that the child node X_c is in state y?*

Each time the question is asked, the scenario changes, such that, by the end of the elicitation process all possible combinations of parent node states and child node states have been covered. The exact formulation of the questions can be altered to make the questions easier to interpret, but for all of the scenarios, the structure may remain the same. For the example BN of Section 2.3.5, one question could be formulated as: Given that the entertainment quality is *high*, the availability of food and drinks is *moderate* and the number of people is *moderate*, what is the probability that the party atmosphere is *amazing*? By replacing the italicized words with the other possible states, all scenarios can be elicited. Note that, instead of asking for probabilities,

also relative frequencies can be used, which allows for uncertainty intervals to be elicited as well.

If the experts were to answer the question for all possible combinations of parent node states and child node states, this would amount to a total of $s_C \cdot \prod_{i=1}^{n} s_i$ parameters. This number can be reduced by using the fact that the probabilities in one CPT row, that specifies the child node distribution for a combination of parent states, should sum to one. Thus, only $s_C - 1$ probabilities are necessary to be elicited for each row, which would bring the total number of to-be-elicited parameters down to $(s_C - 1) \cdot \prod_{i=1}^{n} s_i$.

4.3. NOISY OR/MAX

The first expert burden-reducing method for constructing CPTs that is introduced is called the Noisy-OR method. Extensions of the method, which introduce leaks, and the Noisy-MAX method are introduced in this section as well. For each of the methods, calculation methods are mentioned and what types of questions may be used for elicitation purposes.

The Noisy-OR model is perhaps the most well-known method for constructing CPTs (Kim & Pearl, 1983; Pearl, 1988). This model interprets the relation between parent and child nodes as a causal relationship, where it is assumed that all parental influences are independent of each other. The model applies to BNs with binary nodes, so nodes with two states. This means that the method does not apply to the example of Section 2.3.5 unless the nodes are altered such that each has two states.

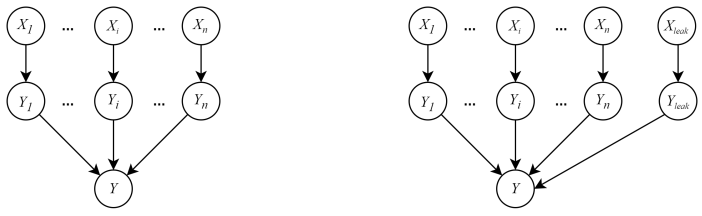
In this section, parent nodes will be referred to as causes that can be present or not. The child node is referred to as the effect, which can also be present or absent. If this method were to be applied to the example BN of Figure 2.6, the states of each node would have to be combined such that only two states are left. This can be a complex task and may mean that less detail can be captured in the model, possibly making it a large concession.

As the name suggests, the relationship between the influence of the parent nodes is modeled by a logical OR gate with added noise. In contrast to a deterministic OR gate, for which the presence of at least one of the causes will guarantee the presence of the effect, the Noisy-OR model does not guarantee this output. Instead, the presence of the effect is dependent on probability. Noisy-OR can be modeled using a deterministic OR gate by introducing inhibitor nodes. In Figure 4.2a the general model is shown, in which $X_1, ..., X_n$ are the causes, Y is the effect, and $Y_1, ..., Y_n$ are the inhibitor nodes, which represent the noise. The CPT of Y has the following form:

$$\mathbb{P}(y_i|x_i) = p_i,$$
$$\mathbb{P}(y_i|\bar{x}_i) = 0.$$

Thus, p_i is the probability that the effect is there, given that cause X_i is present. The second line ensures that the absence of a cause never influences the presence of the effect. So, the complete CPT can be constructed by just n parameters, one parameter for each parent node. When experts are involved in determining the parameters, n questions need to be asked:

- *What is the probability that the effect Y is present, given that the cause X_i is present, and all other causes in the model are absent?*



(a) Noisy-OR. (b) Leaky Noisy-OR.

Figure 4.2: Local structure of a BN for the Noisy-OR model and the Leaky Noisy-OR model.

LEAKY NOISY-OR

The Noisy-OR model can be extended to include the possibility of there being more causes for the child node to have a positive state than just the influence of the parents. This model is called Leaky Noisy-OR. A leak is

introduced that models the probability that the child node is in a positive state given that all parental causes in the model are absent. This can be modeled by adding an extra variable X_{leak} which has a probability of being present of $\mathbb{P}(x_{leak}) = p_{leak}$. This variable is once again independent of the other causes and can be modeled by adding an extra two nodes to the network as shown in Figure 4.2b.

To give the formal definition, as in Fenton et al., 2019, let $y, x_1, ..., x_n$ be boolean variables that take the value 1 if present and 0 if absent, that have weights $p_i \in [0,1]$. Let $p_{leak} \in [0,1]$ be the leak factor, then:

$$\mathbb{P}(y = 1 \,|\, x_1, \ldots, x_n) = 1 - (1 - p_{leak}) \prod_{i=1}^{n} (1 - p_i)^{x_i}.$$

If $p_{leak} = 0$ the original Noisy-OR model is recovered. Thus, the probability of the effect Y being present, when all causes $X_i, i = 1, ..., n$ are absent is $\mathbb{P}(y = 1 \,|\, x_1 = 0, ..., x_n = 0) = p_{leak}$.

The model can either be parameterized by directly eliciting the p_i, or by eliciting $q_i = \mathbb{P}(y|\bar{x_1}, .., x_i, ..., \bar{x_n})$. The second option might be easier to understand as it relates closer to a possible real-life situation, where only one cause is present and all other causes are absent. The two parameters can be linked by the following relation:

$$q_i = 1 - \frac{1 - p_i}{1 - p_{leak}}. \tag{4.1}$$

To elicit the $n+1$ parameters needed, one for each parent node and one for the leak, the following two types of questions can be used in elicitations:

(i) *p_i: What is the probability that Y is present when X_i is present and all other causes of Y that we are considering in the model are absent?*

(ii) *q_i: What is the probability that X_i produces Y? or What is the probability that Y is present when X_i is present and all other causes of Y (including those not modeled explicitly) are absent?*

The first type of question can be used to elicit the p_i, and the second for q_i. Only one of these two types of questions is necessary to be elicited. The difference between the two questions may be hard to see. For the second type of question, the leak is separated from the other causes, which can be hard to understand. In one study about 50% of subjects seemed to answer the second question as if it were the first (Zagorecki & Druzdzel, 2004).

NOISY-MAX

A further extension of the (Leaky) Noisy-OR model is to include nodes with more than two states (Henrion, 1988). For the parent nodes, this is relatively simple, each state of the parent node now has its own probability of causing the effect. Also, the child node can be extended to include more than two states (Diez, 1993). In this case, the max function is needed, and the method is then better known as the Noisy-MAX model. The child node will take the state that corresponds to the maximum state of the parent nodes. So for each combination of a parent node state and a child node state, a probability needs to be known. This means that a total of $\sum_{i=1}^{n} s_C \cdot s_i$ parameters need to be elicited. If also a leak is present, another s_C parameters are added.

The questions for the Noisy-OR model have to be slightly adjusted to account for the extra states included. For example, the question could become:

- *What is the probability that Y is in state y when X_i is in state x and all other causes of Y that we are considering in the model are absent?*

4.3.1. LIMITATIONS

The model reduces the number of parameters that are to be elicited greatly, but this comes with some limitations. To start, the parent nodes are assumed to have independent effects on the child node, so the model is not able to model the joint effects. For the example BN of Section 2.3.5, all parent nodes can be considered to have a monotone influence on the child node separately. However, when one considers the combination of low food availability with a small number of people equivalent to a high availability of food in combination with a high number of people attending, this cannot be modeled by Noisy-OR/MAX. This equivalence can be made since the relative amount of food available for each person would remain the same. The Noisy-OR model is not flexible enough to capture such joint influences of parent nodes.

Another limitation of the method is the complexity of the elicitation. It might be difficult for experts to separate the leak from the other causes, and they might not be able to differentiate between the two types of parameters. If there exists confusion between the experts, this may make the elicited assessment unreliable.

4.4. RANKED NODES METHOD

The Ranked Nodes Method (RNM) was first developed by Fenton et al., 2007. The method is made for discrete BNs, but especially when nodes are to represent discretized continuous variables, RNM is a popular choice. Since the node states are linked to sub-intervals on $[0,1]$, this is intuitive for discretized continuous scales. As the name suggests, the method requires the nodes in the BN to be ranked. In addition, for most versions discussed here, the method needs all nodes in the BN to have an equal number of states: $s_C = s_i = s$.

The nodes in the example BN of Figure 2.6 do not necessarily represent continuous variables, but each node has an equal number of states and we can assume that the nodes are ranked. For the example BN it is assumed that, for all parent nodes, the *high* state is most favorable for the *amazing* child state.

Over the years, multiple versions of the method have been created, this section will cover some of these iterations by first giving a detailed overview of the original version and an overview of the first updated version. This updated version will be referred to as the improved RNM model, and the further improved models as the Static/Dynamic RNM. For an overview of Static/Dynamic RNM, see Appendix A.

4.4.1. ORIGINAL VERSION

The first version of RNM was introduced by Fenton et al., 2007 and implemented in a software called AgenaRisk (Agena, 2018). A more detailed description of the method can be found in (Laitila & Virtanen, 2016) and (Fenton et al., 2007), this section will be based on those descriptions. It is assumed that any node has a discrete set of states that can be ranked and that the distribution of the child node X_C can be approximated by a truncated normal distribution, which is a unimodal distribution.

Each state of a node, x_i, is to be associated with a subinterval z_i of $[0,1]$, which is called a state interval. Each state interval is of equal width and is disjunct from the others. The union of all state intervals covers $[0,1]$. So, in case a node has s states, the state intervals are: $\left[0,\frac{1}{s}\right),\left[\frac{1}{s},\frac{2}{s}\right),...,\left[\frac{s-1}{s},1\right]$.

Overview of method The first step in the method is to link the node states to the state intervals. Here it is important to set the direction of influence equal for all nodes. For the example of Section 2.3.5, for the availability of food and drinks during a party, it is generally the case that more is better for the atmosphere. Also for the other parent nodes in the example, the direction of influence is already the same. that means that for all parent nodes the following node states and state intervals are linked:

- Low: $[0,1/3)$
- Moderate: $[1/3,2/3)$
- High: $[2/3,1]$

The main idea of the method is based on the following proposition, which is only attained when all nodes have an equal amount of states:

Proposition 4.4.1. *Let the nodes $X_1,...,X_n,X_C$ each have the same amount of states. When each parent node X_i has state x_i^j, which is associated with the same state interval z_i^j for each parent, the child will be associated with the state x_C^j, that is, the mode of the child node distribution is in state interval z_i^j.*

The above proposition ensures that, when all of the parent nodes are in the *moderate* state, the mode of the child node distribution will also be in the *moderate* state. The degree to which the distribution is peaked at the mode is determined by the variance parameter.

The second step is to select a weight function, with possible functions being: $WMEAN$ (4.2), $WMIN$ (4.3), $WMAX$ (4.4), or $MIXMINMAX$ (4.5). This weight function can map the combination of parent node state intervals to the child node state interval. For each different weight function, the mode of the child node distribution has a different tendency. Either the child state interval has a tendency towards the mean of the parent state intervals, slightly below or above the mean, or towards the weighted average of the minimum and maximum parent state interval. Experts can support the selection of a weight function by providing estimates for all combinations of the extreme states for the parent nodes. So, for the party atmosphere example, the mode assessments in Table 4.2 could be used.

The next step is for the expert to choose parent weights $\mathbf{w}$ and a variance parameter σ^2. For $WMEAN$, $WMIN$, and $WMAX$ a weight is needed for each parent node: $\mathbf{w} = (w_1, w_2, ..., w_n)$. For $MIXMINMAX$ only two weights are needed: $\mathbf{w} = (w_{MIN}, w_{MAX})$, for the best and worst state included in a scenario. These weights are used to calculate mean row parameters that are used to determine the CPT values.

	Parent nodes			Child node: Y		
	X_1	X_2	X_3	Dull	Okay	Amazing
1	**High**	**High**	**High**			x
2	**High**	**High**	Low			x
3	**High**	Low	**High**			x
4	Low	**High**	**High**		x	
5	**High**	Low	Low		x	
6	Low	**High**	Low	x		
7	Low	Low	**High**	x		
8	Low	Low	Low	x		

Table 4.2: Example of an expert's assessments for the child node mode for combinations of extreme states of parent nodes, corresponding to the example in Figure 2.6. The bold-faced states are associated with the state interval $(2/3, 1]$.

Elicitation Thus, given a BN with ordered nodes and defined states with state intervals, the following steps can be used in an elicitation:

(i) *Given that parent node X_1 is in state $x_1^{\uparrow}$ (or $x_1^{\downarrow}$) and ... and parent node X_n is in state $x_n^{\uparrow}$ (or $x_n^{\downarrow}$), in what state is the mode of the child node X_C?*

(ii) Determine weight and variance parameters by trial and error:

(a) *What is a suitable weight for parent node i?*

(b) *What is a suitable variance parameter for the CPT?*

For the first step, the most positive state of a parent node is denoted by $x_i^{\uparrow}$, and the most negative state by $x_i^{\downarrow}$. The questions of step (i) can then be used to make a table such as Table 4.2, which can be used by the modeler to determine a suitable weight function and by the experts to align their weight and variance parameters to.

For step (ii), there are no guidelines given in the literature on how to exactly elicit the weights and variance parameters other than using trial and error to match the previously elicited best estimates in Table 4.2. The trial and error process can be guided by an elicitor who takes the expert's input for every trial and calculates the resulting CPT. The expert may change the weight and variance parameters until the resulting CPT represents their view well enough. It is possible to first individually elicit parameters and then aggregate the experts' CPTs, or the experts can go through the trial and error process as a group. Using the Agenarisk (Agena, 2018) software, experts would also be able to go through the process individually if they have sufficient probability knowledge. The variance parameter is that of the truncated normal describing the child node's distribution for each row, which is also chosen by trial and error to match the experts' beliefs.

The first step needs a total of $2n+2$ assessments, and the second step needs $n+1$ assessments for each trial. Depending on the number of trials the experts need to determine suitable weight and variance parameters, the total number of input parameters ranges from $3n+3$ to $2n+2+n_{iter}(n+1)$. If the constructed CPT does not represent the experts' beliefs well, the CPT can also be constructed in parts by repeating the steps above for each part, or the elicitation can be redone from scratch.

Calculation When all parameters are set, each value of the CPT is calculated separately, so for each combination of parent node states the following process will need to be repeated. For each parent node state x_i, p equidistant points are sampled from the corresponding state interval z_i. Two of these points are the lower and upper bounds of z_i. This results in a total of $n \cdot p$ sample points. Each combination of sample points k, with one sample point from each parent node, can form the following set: $\{(z_{1,k}, ..., z_{n,k})\}_{k=1}^{p^n}$. For each k one of the following weight functions is used to calculate a mean parameter μ_k:

$$WMEAN: \quad \mu_k = \frac{\sum_{i=1}^{n} w_i \cdot z_{i,k}}{\sum_{i=1}^{n} w_i}, \tag{4.2}$$

$$WMIN: \quad \mu_k = \min_{i=1,\dots,n} \left\{ \frac{w_i \cdot z_{i,k} + \sum_{j \neq i} z_{j,k}}{w_i + n - 1} \right\}, \tag{4.3}$$

$$WMAX: \quad \mu_k = \max_{i=1,\dots,n} \left\{ \frac{w_i \cdot z_{i,k} + \sum_{j \neq i} z_{j,k}}{w_i + n - 1} \right\}, \tag{4.4}$$

$$MIXMINMAX: \quad \mu_k = \frac{w_{MIN} \cdot \min_{i=1,\dots,n}\{z_{i,k}\} + w_{MAX} \cdot \max_{i=1,\dots,n}\{z_{i,k}\}}{w_{MIN} + W_{MAX}}. \tag{4.5}$$

The CPT value that corresponds to the combination of parent state nodes can then be calculated. This is equal to the average value of the pdf of the truncated normal distribution on $[0,1]$, with the previously calculated mean μ_k and chosen variance σ^2, integrated over the child state interval z_C:

$$\mathbb{P}(X_C = x_C | X_1 = x_1, \dots, X_n = x_n) = \frac{1}{p^n} \sum_{k=1}^{p^n} \int_{z_C} TNormpdf(u, \mu_k, \sigma^2, 0, 1)\, du. \tag{4.6}$$

Finally, after the CPT is fully calculated, it can be verified. Once again, the expert can go back to a previous step and change one or more of the parameters. It is also possible to construct the CPT in parts. It may be that only for a certain number of rows, the distribution does not fit appropriately. In that case, for those rows, a different set of parameters and a separate weight function can be used to construct the CPT.

4.4.2. IMPROVED

The enhanced version of the RNM (Laitila & Virtanen, 2016), proposes a method that improves the flexibility of RNM. This new method applies to nodes that represent continuous variables, by having states based on discretizations of the continuous values. The scale intervals that are used to discretize the nodes are no longer fixed during the creation of the CPT. Instead, experts are involved in setting the boundaries for state discretizations. In addition, guidance is provided for the experts to determine the weights.

Overview of method For a parent node X_i each state x_j is linked to a scale interval $Z_i^j = [A_i^j, B_i^j]$, which is linked to the interval $z_i^j = [a_i^j, b_i^j]$ on the normalized scale. All of the A_i^j and B_i^j are then associated with a_i^j and b_i^j respectively. Note that $a_i^1 = 0$ and $b_i^n = 1$ for all $i = 1, \dots, n$, and $a_i^j = b_i^{j-1}$ for $j = 2, \dots, s_i$, $i = 1, \dots, n$ of the previous interval, since all intervals are adjacent to each other.

The state intervals z_i remain the same as for the original version of RNM, which means that the interval $[0,1]$ is once again split in s disjoint intervals of equal width. The first step is to let the experts decide how these state intervals match the interval scales. For all of the state interval boundary points $a_C^1, \dots, a_C^s, b_C^s$ of the child node it is checked whether this corresponds to the parent state boundary points. So, for example, if all parent nodes are in the states $a_1^2, \dots, a_n^2$ the parent nodes would have values $A_1^2, \dots, A_n^2$, it is then checked if A_C^2 is a good fit. This is done for all $n+1$ boundary points.

Based on the determination of the intervals that correspond to the node states, a piece-wise linear mapping can be defined. The linear map is between the state interval and the scale intervals, for a node X_i with s states, the map $h_i(x) : \mathbb{R} \to [0,1]$ is defined as:

$$h_i(x) = \begin{cases} h_i(A_i^j) = a_i^j, \quad \text{for } j = 1, \dots, s, \\ h_i(B_i^s) = b_i^s, \\ h_i(x) = h_i(A_i^j) + \left(\frac{x - A_i^j}{B_i^j - A_i^j} \right) \left(h_i(B_i^j) - h_i(A_i^j) \right), \quad \text{for all } x \in [A_i^j, B_i^j],\ j = 1, \dots, s. \end{cases} \tag{4.7}$$

In the definition, A_i^j is the jth left boundary point of node X_i, which corresponds to a_i^j.

The method can then use expert elicitation to find weights. Experts are asked to give assessments for $2n$ cases, where each case consists of the best or worst-case scenarios of the expert. Let y_i denote the value of

Table 4.3: Weights obtained from expert elicitation for the mode of the child node, for *WMEANS* (1), *WMIN* (2), *WMAX* (3), and *MIXMINMAX* (4).

	Weight a	Weight b	Feasibility conditions
(1)	$w_k^{N,a} = 1 - h_C(\hat{y}_C^{a,k})$	$w_k^{N,b} = h_C(\hat{y}_C^{b,k})$	$w_k^{N,a} = w_k^{N,b} = w_k^N \in [0,1] \forall k$ $\sum_{k=1}^n w_k^N = 1$
(2)	$w_k = \frac{(n-1)(1-h_C(\hat{y}_C^{a,k}))}{h_C(\hat{y}_C^{a,k})}$	$v^k = \frac{1}{h_C(\hat{y}_C^{b,k})} - n + 1$	$w_1, ..., w_n \geq 1$ $v^k = \max_{i \neq k}\{w_i\} \forall k$
(3)	$v^k = \frac{1}{1-h_C(\hat{y}_C^{a,k})} - n + 1$	$w_k = \frac{(n-1)h_C(\hat{y}_C^{b,k})}{1-h_C(\hat{y}_C^{b,k})}$	$w_1, ..., w_n \geq 1$ $v^k = \max_{i \neq k}\{w_i\} \forall k$
(4)	$w_{MIN}^{N,a,k} = 1 - h_C(\hat{y}_C^{a,k})$ $w_{MAX}^{N,a,k} = h_C(\hat{y}_C^{a,k})$	$w_{MIN}^{N,b,k} = 1 - h_C(\hat{y}_C^{b,k})$ $w_{MAX}^{N,b,k} = h_C(\hat{y}_C^{b,k})$	$w_{MIN}^{N,a,k} = w_{MIN}^{N,b,k} =: \lambda \in [0,1] \forall k$

parent node i on the scale interval. Then the following two sets of scenarios will be elicited:

$$\begin{cases} S_a = \{y^{a,k} = (y_1^{a,k}, ..., y_n^{a,k}) \mid h_i(y_i^{a,k}) = 1 \forall i \neq k, \text{and } h_k(y_k^{a,k}) = 0\}_{k=1}^n, \\ S_b = \{y^{b,k} = (y_1^{b,k}, ..., y_n^{b,k}) \mid h_i(y_i^{b,k}) = 0 \forall i \neq k, \text{and } h_k(y_k^{b,k}) = 1\}_{k=1}^n. \end{cases} \tag{4.8}$$

Thus, $y^{a,k}$ represents the combination of parent node values where all nodes have value $y_i^{a,k} = B_i^n$, but only node k is in the worst state, and $y^{b,k}$ represents the opposite. Let $\hat{y}^{a,k}$, $\hat{y}^{b,k}$ be the expert assessment for the mean of the child node in these situations. This assessment can be given as an interval where the expert thinks the mode will fall between: $[\underline{\hat{y}}^{a,k}, \bar{\hat{y}}^{a,k}]$. Following the weight feasibility conditions and weight formulas as stated in Table 4.3, a weight function can be chosen, and intervals can be found for the weights. The expert will still need to decide which weights are to be used exactly.

Finding weights that accurately represent the experts' beliefs remains an iterative process. Thus, the final step of RNM also stays the same, for a selection of CPT rows it is checked with the experts whether the distribution for the child node makes sense to them. If not, the CPT constructing process can be repeated, where it is also possible to redetermine the CPT in parts.

Elicitation Given a BN with ranked nodes, and an initial discretization, the following steps are part of the elicitation process to construct a CPT in one part:

(i) *Given that parent node X_1 has value A_1^j and ... and parent node X_n has value A_n^j, should the mode of the child node X_C have value A_C^j?*

  (a) If not: *Which of the values $A_1^j, ..., A_n^j, A_C^j$ would need to be changed, such that the values are compatible with each other? And what should the new value(s) be?*

(ii) *Given that the parent nodes have values as in scenario S_a (or S_b), what is the value of the mode of the child node $\hat{y}^{a,k}$ (or $\hat{y}^{b,k}$)? Or, in what interval is the mode of the child node: $[\underline{\hat{y}}^{a,k}, \bar{\hat{y}}^{a,k}]$ (or $[\underline{\hat{y}}^{b,k}, \bar{\hat{y}}^{b,k}]$)?*

(iii) Determine weight and variance parameters by trial and error, given the feasibility constraints in Table 4.3:

  (a) *What is a suitable weight for parent node i?*

  (b) *What is a suitable variance parameter for the CPT?*

Before the elicitation process begins, an initial discretization is made freely. This discretization is revised by the experts by using questions of step (i). If intervals are elicited in step (ii), feasible regions can be computed

which helps the trial and error process of step (iii). If the constructed CPT, using the elicited parameters, does not portray the experts' beliefs properly, the CPT can also be determined in parts, by repeating the steps above for different parts of the CPT.

So step (i) amounts to $s+1$ parameters to be elicited, step (ii) requires $2n+2$ or $2*(2n+2)$ assessments, and in step (iii) an additional $n_{iter}(n+1)$ values are elicited.

4.4.3. LIMITATIONS

The original version was praised for needing only $n+1$ parameters, n for the parent weights, and one for the variance. However, this number does not take everything into account, the original RNM additionally asks experts to give their best guess for the mode of the child node distribution for all combinations of best- and worst-case states of all parent nodes. Furthermore, since the weights are determined using a trial and error process, this can quickly increase the expert burden. As previously mentioned, the original version requires $(n+1)\cdot n_{iter}+2n+2$ parameters, and the first improved version needs $(n+1)\cdot n_{iter}+2n+s$.

The CPT may also be constructed with multiple weight expressions and variance parameters, which means the CPT is constructed using partitions. Note that the number of parameters that need to be elicited quickly rises each time the CPT is partitioned. After some partitions, the number of values needed comes close to the number of values in the full CPT specification.

As the name suggests, the method calls for ranked nodes. In experiments, it was found that, while the method can be used on other types of nodes, the original method and first improved method perform best on so-called elementary RNM-compatible nodes (Laitila & Virtanen, 2020). A child node and its parent nodes are called elementary RNM-compatible if they all have the same number of states and if the states can be sorted such that Proposition 4.4.1 is attained. This makes the method less flexible, and not applicable to all BNs, as in some cases no 'perfect' BN can be constructed.

Proposition 4.4.1 also forces the influences of parents to be independent. The method cannot model cases in which certain parent nodes may cancel each other out, or magnify each other's influence on the child node.

Next, consider the types of parameters that are to be elicited. These include weight parameters that might be unnatural to understand for experts. What those weights exactly are, especially for the MIXMINMAX mean function, is not intuitive. The fact that trial and error is used to determine these weights serves as a testament to this.

Apart from the weight parameters, the variance parameter being a single parameter is a limiting factor to the flexibility. This forces all of the rows of the CPT to be generated with an equal variance. The only way to vary the variance is to construct the CPT in parts. The option to have a varying variance parameter is not included in the introduced versions of RNM.

The method now also calls for trial and error to choose the weight function and refine the weights and variance. For this iterative process to run smoothly, the generation of a CPT based on new parameters must take as little time as possible. In an experiment, it was found that a sample size of $p = 5$ is enough to generate CPTs that represent the experts' view well enough. Additionally, it was found that, when the BN has 5 parents or less, and each node has at most 7 states, the generation of the CPT is still in the order of seconds, if it is constructed at once (Laitila & Virtanen, 2020). Thus, for the elicitation process to be time-efficient there is a limit on the number of parents and states of the BN that is suitable for RNM.

4.4.4. PROPOSED IMPROVEMENTS

Some of the previously stated limitations may be improved upon. In this section, I propose one main enhancements for the RNM method. In Chapter 6.2 this possible improvement will be revisited.

The number of parameters needed as input for RNM can become relatively large when the trial and error phase is considered. So eliminating this phase should result in a method that requires fewer parameters. The search for weights that ensure the resulting CPT matches the experts' views could be (partly) automated. I propose that instead of eliciting only the mode of the child node, full multinomials should be elicited for those scenarios instead. For example, the mode assessments of Figure 4.2 could be extended to give a probability for each child node state for each combination of parent node states in the table. That would increase the initial number of probabilities to be assessed from $2n+2$ to $s_C(2n+2)$ but would remove the need for variance and weight parameters to be elicited. Thus, the additional burden depends largely on the number of states the child node can take.

Using optimization techniques to find weights also removes the problem of the weights being less straight-

forward to understand for experts. At the same time, it also allows experts to express their uncertainty regarding the mode of the child node in certain scenarios. In the current version, experts were only able to show the uncertainty by choosing a variance parameter and adjusting weights to match their views.

The feasibility conditions for the weights, as stated in Table 4.3, give guides for the optimization process. These boundaries reduce the domain in which an optimal answer needs to be found. A simple grid search algorithm may already be sufficient. More information on the application of this possible improvement can be found in Section 6.2

4.5. INTERBETA

InterBeta (Mascaro & Woodberry, 2022) is a method that uses interpolation to construct CPTs when only best- and worst-case scenarios are given. In that way, the method is similar to Cain's method, which will be introduced later in this section. Like RNM, InterBeta also assumes that the nodes are ranked and that parental influences can be weighted, before combining them using an independent combination function. The main assumption of the method is that the child node can be approximated by a beta distribution. This distribution is bounded on $[0,1]$ and offers flexibility as it can approximate uniform, unimodal, and bimodal (at each boundary) distributions.

Using interpolation to construct the CPT works by interpolating the parameters of the beta distribution. The best and worst row of the CPT are elicited from experts, by either specifying the beta distribution parameters, giving a mean and standard deviation, or by giving a multinomial distribution. If the latter is the case, InterBeta fits the beta distributions for the best and worst rows by using the method of moments and an optimization strategy. First, the mean μ and variance σ^2 are calculated, which can then be used to calculate α and β using the method of moments:

$$\alpha = \mu\left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right), \quad \beta = (1-\mu)\left(\frac{\mu(1-\mu)}{\sigma^2} - 1\right) \tag{4.9}$$

The α and β are mutated iteratively using Gaussian noise, where only improvements of the Kullback-Leibler (KL) divergence, which wll be defined in Section 4.8, between the discretized fitted beta distribution and the originally elicited multinomial are accepted. The process is stopped after a chosen number of iterations, 1000 was found to be an adequate number (Mascaro & Woodberry, 2022). The beta distributions are discretized by assuming that the s_C child states equally divide the interval $[0,1]$, into $[0, \frac{1}{s_C}), [\frac{1}{s_C}, \frac{2}{s_C}), ..., [\frac{s_C-1}{s_C}, 1]$. The corresponding multinomial is then found by calculating the probability mass within each interval.

Once the beta distributions have been fit to the best and worst rows, the interpolation can commence. For a combination of parent states $\mathbf{X}_k = \mathbf{x}_k = (X_1 = x_1, ..., X_n = x_n)$ the child node distribution obeys the following conditional probability:

$$\mathbb{P}(X_C \mid \mathbf{X}_k) \sim Beta\big(g_\alpha(\mathbf{x}_k), g_\beta(\mathbf{x}_k)\big), \tag{4.10}$$

where the functions g_α and g_β are the interpolated distribution parameters. To calculate these parameters, each CPT row is given a weight $w_k \in [0,1]$, where the best and worst rows have weights 1 and 0 respectively. These weights are determined based on the input parameters given, which will be set out later in this section. If parent state weights are known, these are aggregated independently using some type of mean function, such as the arithmetic mean, geometric mean, or harmonic mean. If the worst case row is approximated by $Beta(\alpha^\downarrow, \beta^\downarrow)$, and the best case row by $Beta(\alpha^\uparrow, \beta^\uparrow)$, the parameters can be calculated using the following equation:

$$\begin{aligned} \hat{\alpha}_k &= g_\alpha(\mathbf{x}_k) = w_{y_i} \cdot \alpha^\uparrow + (1 - w_{y_i}) \cdot \alpha^\downarrow, \\ \hat{\beta}_k &= g_\beta(\mathbf{x}_k) = w_{y_i} \cdot \beta^\uparrow + (1 - w_{y_i}) \cdot \beta^\downarrow. \end{aligned} \tag{4.11}$$

This ensures that all $Beta$ parameters lie on a line between the best and worst-case parameters.

The InterBeta method has a list of variations, such as a variant only using the best and worst CPT row and one that also includes parent weights. For each variation, a different number of parameters are needed, but also a different level of detail can be captured. All but two of these methods are described here. The "Defaults" method requires no input and may be useful during exploratory modeling, for qualitative analysis. On the other end of the spectrum, there is also the option to specify beta distributions for each row, which removes the need for interpolation entirely. The variations that fall in between in terms of complexity are presented here in greater detail.

How the computations are carried out exactly, and how it is implemented in Python is described in Section 6.1.2. The variations are introduced in order of complexity. As for each variation, a new type of parameter needs to be elicited, the corresponding elicitation questions are also introduced per version.

BEST AND WORST

For this variation, only the best row and worst row of the CPT are elicited. The structure of the questions will be similar to that of eliciting the full CPT, but only for two scenarios, resulting in two types of questions:

- *Given that parent node X_1 is in state $x_1^{\uparrow}$ and ... and parent node X_n is in state $x_n^{\uparrow}$, what is the child node distribution?*
- *Given that parent node X_1 is in state $x_1^{\downarrow}$ and ... and parent node X_n is in state $x_n^{\downarrow}$, what is the child node distribution?*

Where $x_i^{\uparrow}$, and $x_i^{\downarrow}$ represent the most positive and most negative state of parent node i respectively.

In case they are elicited as multinomials, following the questions above, the total number of parameters to be elicited is $2s_C$, where s_C is the number of child node states. Since there are no weights elicited, these are computed from scratch. The parent node states are mapped to equispaced points in the interval $[0,1]$. For a node X_i with $|X_i|$ states, the states are mapped to 0, $\frac{1}{|X_i|-1}$, $\frac{2}{|X_i|-1}$, ..., 1, according to their rank. The effect of moving from one state to the next is equal within one node, or between nodes if they have an equal number of states.

The row weights are then calculated by combining the mappings of the inputs, which are finally normalized to the unit interval $[0,1]$. The method for combining the weights can be chosen, for example, the arithmetic mean or the geometric mean can be used. The latter indicates the central tendency of the input values and showed to be promising in applications by Barons et al., 2022.

PARENT WEIGHTS

For this variation, not only the best and worst row of the CPT are elicited, but also a weight parameter for each parent node. Thus, totaling $2s_C + n$ parameters to be elicited. There exists no protocol for eliciting these weights, but the weights could be elicited in two steps:

(i) *Order each of the parent nodes in terms of the strength of their influence on the child node.*

(ii) *Relative to the parent node with the weakest influence of the child node distribution, what weight do you give to parent node i?*

After completing step (i), the parent node with the weakest influence is given a weight of one, the other nodes are given weights by the experts in step (ii).

The parent states are then mapped to the interval $[0, w_{X_i}]$ in the same way as for the Best and Worst variation, resulting in the mapping: 0, $\frac{w_i}{|X_i|-1}$, $\frac{2w_i}{|X_i|-1}$, ..., w_i . Also, the calculation of the row weights remains the same as for Best and Worst. The additional parameters enable the method to alter the relative influences of the parents on the child node.

PARENT STATE WEIGHTS

This further variation does not only allow for weights to be given to parent nodes but also to each intermediate parent state between the best and worst. So for each parent node, an extra $s_i - 2$ parameters are needed, resulting in a total of $2s_C + \sum_{i=1}^{n}(s_i - 1)$. Since the parent weights are included in the $\sum(s_i - 1)$ extra parameters, n does not need to be added separately. As for the parent weights, there exist no guidelines for eliciting these parent state weights. Given ranked nodes, they could be elicited in the following steps:

- *For parent node i: given that the most negative state has weight zero and the most positive state has a weight equal to the parent weight, what weight do you give to state j?*

The parent states are again mapped to the interval $[0, w_{X_i}]$. But in this case, for each node, the parent states are not mapped with equidistant points, but according to the weight they have received. Using the supplied parent state weights, the row weights will be determined once again by using some independent combination function of the mappings.

ROW WEIGHTS

The final variation of InterBeta which uses interpolation, asks experts to assess the best row and worst row of the CPT, as well as weights for each row. So, that requires a total of $2s_C + \prod_{i=1}^{n} s_i$ values to be elicited. This is a significantly higher number than needed for the previous variations. To lighten the elicitation burden, one of the other variations can be used to generate initial weights, which can then be altered by the experts as desired.

Also for the row weights there does not exist a protocol for elicitation. Potentially the row weights could be given in two steps:

(i) *Make an ordering of all combinations of parent node states (rows), where the row that pushes the child node distribution towards its most positive state the most is ranked best. The row for which the child node is most likely to be in its most negative state should be ranked the worst.*

(ii) *For each row k: what is the weight of this row?*

Note that, in step (i) it is allowed to rank a row that is not the "best row" the highest.

Row weights also allow for special cases to be distinguished. For example, it can be set that anytime a certain node is in the lowest state, the child node will also be in the lowest state. Instead of manually changing the weights of these rows to zero, a decision tree could be used to determine the weights. That way, Parent Weights or Parent State Weights can generate initial weights, and the decision tree can be added to adjust the weights for some cases.

The main advantage of this variation over the others is that arbitrary dependencies between the parent nodes and the child nodes can be modeled. The parental node effects no longer need to be independent. The method is still restricted to beta distributions with parameters that are linearly interpolated between the best and worst rows.

4.5.1. LIMITATIONS AND PROPOSED IMPROVEMENTS

Like RNM, the InterBeta versions that don't use row weights are only applicable to ranked nodes. This, in combination with the independent effects of the parent nodes, means that the influences of certain combinations of parent states can not always be modeled correctly. Having experts give row weights may be challenging, like parent weights and state weights, this is less intuitive than assessing conditional probabilities. In Section 8.1 it is shown how the parent weights relate closely to the correlations found between the child and parent nodes.

Another limitation of the method is that the interpolated beta distribution parameters are all on a line between the best and worst beta or mean/variance parameters. This does not allow for much flexibility. The method does not allow for other rows to be elicited and given as input to InterBeta. This could possibly be improved upon by adding intermediate rows to the elicitation. The beta parameters can then be interpolated between each of the elicited rows, or possibly a quadratic interpolation could be used instead.

Related to this limitation is the type of rows that need to be assessed by the expert. InterBeta heavily relies on the best and worst row of the CPT. If experts are comfortable giving probabilities for these scenarios, there is no problem. But often, the best and worst case scenario are ones that do not occur frequently. It could lead to experts anchoring to a scenario they are familiar with, then adjusting this to find the best and worst case. Instead, it is also possible to allow experts to choose the elicited rows themselves, potentially with guidelines, then using both interpolation and extrapolation to construct the rest of the CPT.

Another factor that may limit flexibility is the set discretization intervals of the child node distribution. When the beta distributions are fit and later discretized, the discretization intervals are each of equal width. If CPTs contain rows where close to full probability is given to a single state, the beta distributions for these rows would need α, β parameters which are large or close to zero. If this is the case for the best or worst row, this would also affect the beta distribution parameters for the intermediate rows. In Section 8.6 the influence of the size of the discretization intervals on the accuracy is tested.

Finally, note that the parent state weights version of InterBeta gives weight to all parent states, except for the worst states. This means that state weights cannot model a potentially negatively dominant parent node, which forces the child node distribution to the worst state when this parent node is in its worst state. A different version that uses full state weights could be used instead, where the states are given default values between zero and one and a separate weight.

4.6. THE FUNCTIONAL INTERPOLATION METHOD

The final CPT construction method that is tested in this thesis is the Functional Interpolation method. Similar to the InterBeta method, the functional interpolation method relies on the interpolation of parameters for filling CPTs (Podofillini et al., 2014). Instead of the beta distribution, the normal distribution is used, and instead of only the best and worst row being fully elicited, all combinations of the extreme states of the parent nodes are elicited. That means, if $x_i^{\uparrow}, x_i^{\downarrow}$ are the most positive and most negative state of parent node i respectively, the child node distribution is to be elicited for each combination of parent node states $(x_1, ..., x_n)$ where $x_i \in \{x_i^{\uparrow}, x_i^{\downarrow}\}$. So, in total 2^n rows of the CPT need to be elicited, for a child node with s_C states, this results in $2^n s_C$ values that need to be assessed.

As for the other CPT construction methods that are based on interpolation, the following question structure can be used:

- *Given that parent node X_1 is in state $x_1^{\uparrow}$ (or $x_1^{\downarrow}$) and ... and parent node X_n is in state $x_n^{\uparrow}$ (or $x_n^{\downarrow}$), what is the child node distribution?*

The remaining unelicited values of the CPT are then determined in the following way. First, if multinomials are elicited, normal distributions are fit on the elicited rows, it is also possible to directly elicit the mean and standard deviation. The mean and standard deviation are then linearly interpolated for all rows that fall between the elicited rows. For each row, the normal pdf with these parameters can then be discretized to find the CPT values. Note that the normal distribution is used, which has support $(-\infty, \infty)$, so the calculated CPT values will still need to be normalized such that they sum to one. The exact calculation methods are set out in Section 6.3.

4.6.1. LIMITATIONS AND PROPOSED IMPROVEMENTS

The Functional Interpolation method relies on the normal distribution for the child node distribution, but this is less flexible than the beta distribution, as it is not able to replicate bimodal discrete distributions. However, the normal distribution can easily replaced by other distributions. In Section 6.3 it is tested what distribution (normal, truncated normal or beta) performs the best to construct accurate CPTs.

Another factor that lacks in the Functional Interpolation method, is the fact that no weights are included for the parent nodes. Although this is not a complex addition to the calculation method, it would increase the burden of the already burdensome method. This would only be viable for BNs with parent nodes that have a relatively high number of states, as the number of parent states does not influence the expert burden.

4.7. APPLICATIONS OF CPT CONSTRUCTION METHODS IN LITERATURE

This section contains an overview of some applications of the previously discussed CPT construction methods. The section consists of an overview of some studies that have been performed to compare the performance of several CPT construction methods. In addition, for each of the previously introduced methods, some examples of real-life applications are highlighted.

COMPARISON STUDIES

Two studies compared the performance of the EBBN method, the likelihood method, and the weighted sum algorithm. The first study found that the likelihood method outperformed EBBN and the weighted sum algorithm (Knochenhauer et al., 2013). The three methods were used to construct CPTs for three nodes of the *Car diagnosis 2* BN in Netica (Corp., 2007). As an accuracy measure, the mean of the absolute difference between all probabilities in the constructed BN and the original BN in Netica was used. The likelihood method can better handle situations for which the child's state probabilities switch from low to high, or vice versa.

The other study found that the likelihood method produced the most realistic results, although it did take more time to calibrate the weights (Zio et al., 2022). The weighted sum algorithm and EBBN did not give high enough probabilities to the positive child node state in certain cases. For both methods, this was explained by the fact that the relative weights depend on the number of parents, where more parents means smaller relative weights.

Another study compared the Functional Interpolation method, EBBN, RNM, Cain's method, and Røed's method (Mkrtchyan et al., 2016). The methods were tested on two BNs, one with two parent nodes and one with three parent nodes, that concerned human reliability analysis. Two aspects of CPT construction methods were found to be important, the ability to address strong factor influences (single and multi-factor) and

proper uncertainty characterization. The functional interpolation method addressed both of these aspects the best, however, the downside of the method is that the elicitation burden increases exponentially with the number of parent nodes included in the BN.

NOISY-OR/MAX

The applicability of the method has been studied previously on existing BNs. Three BNs were included in the study, which totaled 67 CPTs of child ndoes with at least two parent nodes. It was found that the Noisy-MAX gate provides a good fit for as many as 50% of CPTs in two of these networks (Zagorecki & Druzdzel, 2013).

Another study investigated the influence of using a leaky Noisy-OR version of a BN, which is used for the early detection of classical swine fever in pigs, on the performance in comparison to a fully elicited version (Bolt & Gaag, 2010). One-third of the variables in the BN were replaced by values constructed by the leaky Noisy-OR model, reducing the total number of parameters needed from 470 to 348. The maximal Kullback-Leibler (KL)-distance was used to compare the original and constructed CPTs, which ranged from $< 10^{-4}$ to 0.2206. In addition, the performance of the BNs was measured by the number of correct suspicions of swine fever in pigs, when applied to a real-life data set containing 466 individual pigs. A maximum of four fever cases were predicted differently by the two models, when applied with different threshold levels. It was cautiously concluded that the method can indeed be applied for diagnostic applications.

More recently, applications of the Noisy-OR/MAX method include the development of BNs for the investment strategies of farmers (Yanore et al., 2023), the occurrence of hydrogen leakage in proton exchange membrane fuel cells (G. Chen et al., 2024), and construction project risks (Ji et al., 2022).

RNM

In one application of RNM, a BN is built to assess and improve the teamwork quality of agile teams (Freire et al., 2018). The BN was constructed with the help of a literature review and one expert. The WMIN function was chosen based on some of the assessments of the expert. The weights and variance parameters were chosen empirically to match the expert's configuration.

Another study proposed a method for constructing BNs based on Decision Making Trial and Evaluation Laboratory (DEMATEL) for eliciting the structure of the BN and RNM to define the CPTs (Kaya & Yet, 2019). DEMATEL is a technique that aims to represent causal relationships through matrices (Si et al., 2018). The matrices found by DEMATEL were also used to find weights and variance parameters for RNM. WMEAN was the function chosen for the mean calculations, but no support was given for this decision. The proposed method was tested in a case study with 14 experts from an automobile manufacturer in Turkey. In expert reviews, it was found that the reasoning mechanism was consistent with domain knowledge.

The fact that the construction of CPTs for a BN does not have to rely on a single construction method is demonstrated by a study on the disaster assessment of the oil and gas supply chain (Sakib et al., 2021). Part of the CPTs in the model are constructed by the Noisy-OR model to add uncertainty and a possible leak, additionally, part of the CPTs are constructed using RNM. No details are given on the used weight and variance parameters of RNM.

INTERBETA

Since the InterBeta method was proposed, no paper has been published that applies the method to construct CPTs from scratch. The method has only been tested on existing BNs that have been fully elicited previously. When InterBeta was tested on two examples, one about food security and one about bees (which will be introduced in the next section), it was found that the parent state weights version constructed CPTs close to those that were fully elicited (Barons et al., 2022). The study used optimized weights to best fit the known CPTs, which means that it is not investigated yet how well the method works when all parameters are elicited from scratch.

FUNCTIONAL INTERPOLATION

One application of the Functional Interpolation method was for dynamic BN modeling of the residual life assessment of corroded sub sea pipelines. The paper does not use static BNs, but a time element is added. The paper proposes improvements to Functional Interpolation to minimize expert burden, and to combine it with the use of actual data (Aulia et al., 2021).

4.8. CPT COMPARISON MEASURES

Once a CPT has been constructed using one of the methods described in this chapter, the next step is to determine the accuracy. Assuming that there is a true CPT, there are many measures that could be used, methods measuring distances between CPTs such as absolute differences, the Shannon-Jensen divergence, the Kullback-Leibler divergence, the Root Mean Squared Deviation (RMSD), the total variation distance, and the Hellinger distance. In addition, the percentage of agreement between CPTs can be measured when different scenarios are entered into the BN, for instance by calculating how often the two CPTs agree on what state the child node is most likely to be in.

This section will describe each of these measures, given that a true CPT is known. In practical cases, the true CPT is rarely known. Therefore, when available, fully elicited BNs are assumed to represent the truth.

ABSOLUTE DIFFERENCES

Perhaps the most straightforward measure is to calculate the absolute difference between the probabilities of the two CPTs. For each pair of probabilities, one from each CPT at the same location, the absolute difference can be calculated. The average absolute difference, over all CPT values, can then be used to measure the similarity between the CPTs.

D_{KL}: KULLBACK-LEIBLER DIVERGENCE

The Kullback-Leibler (KL) divergence of P from Q can be interpreted as the expected surprise from using Q as a model when the actual distribution is P. If both P and Q are pmfs that have s possible states, the metric is defined as:

$$D_{\mathrm{KL}}(P\|Q) = -\sum_{i=1}^{s} P(i)\log\frac{Q(i)}{P(i)}. \tag{4.12}$$

Since $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$, the KL-divergence cannot be seen as a distance. A KL-divergence close to zero shows that the two distributions are similar.

Applied to CPTs, the KL-divergence will be measured between each CPT row. It would also be possible to compare the KL-divergence between the joint distributions, but this also depends on the KL-divergence of the conditional distributions, as shown in Appendix B.1. The average of the KL-divergence over all rows is taken, as well as the maximal value. For a combination of parent node states $k : (x_1^k, ..., x_n^k)$, the KL-divergence is calculated as:

$$D_{KL}^k(P(X_C\,|\,X_1,...,X_n)\|Q(X_C\,|\,X_1,...,X_n)) = \sum_{x_C\in X_C} P(x_C\,|\,x_1^k,...,x_n^k)\ln\frac{P(x_C\,|\,x_1^k,...,x_n^k)}{Q(x_C\,|\,x_1^k,...,x_n^k)}.$$

In this thesis, the mean KL-divergence between two CPTs is used as the primary performance measure, and is defined as:

$$D_{KL}(\mathrm{CPT}_P\,\|\,\mathrm{CPT}_Q) = \frac{1}{\#\text{ CPT rows}}\sum_{k=1}^{\#\text{ CPT rows}} D_{KL}^k(P(X_C\,|\,X_1,...,X_n)\|Q(X_C\,|\,X_1,...,X_n)).$$

D_{JS}: JENSEN-SHANNON DIVERGENCE

The Jensen-Shannon divergence is a smooth and symmetric version of the Kullback–Leibler divergence. It is the average KL-distance between the distributions P, Q, and the equally weighted mixture distribution between P, Q, defined as:

$$D_{\mathrm{JS}}(P\|Q) = \frac{1}{2}D_{\mathrm{KL}}(P\|M) + \frac{1}{2}D_{\mathrm{KL}}(Q\|M), \quad \text{where } M = \frac{1}{2}(P+Q).$$

The square root of the Jensen-Shannon divergence is often referred to as the Jensen-Shannon distance. Once again, the closer this divergence is to zero, the more similar the two distributions are. Similar to the KL-divergence, the JS-divergence for CPTs can be calculated on a row-by-row basis.

RMSD: ROOT MEAN SQUARED DEVIATION

The Root Mean Squared Deviation (RMSD) is the root of the average of the squared errors. Let P be a vector of n observations of the actual distribution, and Q a vector of length n of observations from the model.

$$\text{RMSD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - Q_i)^2}.$$

A score close to zero shows that the distributions are in good agreement.

TVD: TOTAL VARIATION DISTANCE

The Total Variation Distance can be interpreted as the total difference between probabilities that the two probability distributions can assign to the same event. Figure 4.3 shows what the TVD graphically looks like, as it corresponds to half of the area between the probability distributions in the continuous case. For the discrete case, a sum is taken over all possible events. If pmfs P and Q each have s possible states, this gives the following measure:

$$\text{TVD} = \frac{1}{2}\sum_{i=1}^{s}(|P_i - Q_i|).$$

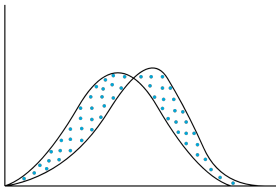


Figure 4.3: The Total variation distance is half of the marked area between probability distributions.

This distance can be calculated for each row of the CPT, where P_i and Q_i represent the distributions of the child node conditional on the parent nodes:

$$\text{TVD}_k = \frac{1}{2}\sum_{x_C \in X_C}(|P(x_C \,|\, x_1,...,x_n) - Q(x_C \,|\, x_1,...,x_n)|).$$

A TVD of zero would mean that all events are given the same probability for both distributions, the distance can take value in $[0,1]$, so a score close to zero is wanted. Pinsker's inequality bounds TVD by the KL-divergence in the following way:

$$\text{TVD}(P,Q) \leq \sqrt{\frac{1}{2}D_{\text{KL}}(P\|Q)}.$$

H: HELLINGER DISTANCE

The Hellinger distance for two discrete probability distributions $P = (p_1,...,p_n)$ and $Q = (q_1,...,q_n)$ is defined as:

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{n}(\sqrt{p_i} - \sqrt{q_i})^2}.$$

The distance can be calculated by pair-wise comparing CPTs on each row. The TVD and the Hellinger distance are related by the following inequality:

$$H^2(P,Q) \leq \text{TVD}(P,Q) \leq \sqrt{2}H(P,Q).$$

The Hellinger distance takes value in $[0,1]$, where a value close to zero signifies a close similarity between the two probability distributions.

PERCENTAGE OF AGREEMENT

Another performance measure, which is not as formally defined in terms of probability distributions, is the percentage of agreement on the most likely state for the child node in different scenarios. As used for the performance testing of EBBN (Wisse et al., 2008). Going over all possible combinations of parent node states, the percentage is that of the cases when the true CPT and the constructed CPT give the highest probability to the same child node state.

For each combination of parent node states k in the total set of combinations $\{(x_1,...,x_n)\}_{k=1}^{S}$, where $S = \prod_{i=1}^{n} s_i$ is the total number of combinations that exist, it is checked if the child states are equal for the constructed CPT and the true CPT:

$$\text{Agreement \%} = \frac{100}{S}\sum_{k=1}^{S}\delta\{x_C^k = \hat{x}_C^k\},$$

where $\delta\{x_C^k = \hat{x}_C^k\} = 1$ when the child state of the true CPT x_C^k is equal to the child state of the constructed CPT $\hat{x}_C^k$, otherwise it is zero. If the CPTs agree on the child state for all combinations of parent states, the agreement percentage is 100%. Thus, for this performance measure a value close to 100% is desired.

5

OVERVIEW OF PREVIOUSLY ELICITED CONDITIONAL PROBABILITY TABLES

The availability of fully elicited Bayesian Networks is limited. There are not many cases of fully expert-elicited CPTs. From those, often data are not made publicly available. In this thesis, a small selection of fully elicited CPTs are used to test the CPT construction methods' performance in Chapter 7. All of the CPTs are part of three main BNs from different application areas: the abundance of pollinators (bees) (Barons et al., 2018), the household food security in Victoria, Australia (Kleve & Barons, 2021), and the persistence of polar bears (Atwood et al., 2016).

An overview of all of the CPTs included in the comparison is given in Table 5.1. For each CPT, information is given on the structure of the BN, the total number of probabilities in the CPT, and how often these CPTs are elicited separately by individual experts (and decision makers). For the Honey Bee Abundance CPT, ten experts' elicitations are included, individually referred to as experts A-J, and the Equal Weights Decision Maker (EWDM). Similarly for the Household Food Security CPT, experts A-E are included, as well as the EWDM and Performance-based Weights Decision Maker (PWDM). Each of the CPTs included in the table parameterizes a BN with a ranked child node and ranked parent nodes. Four of the CPTs in the Polar Bears BN are not included in the table, as either the parent nodes are not ranked or they are equal to an already included CPT.

Table 5.1: Overview of CPTs included in the comparison, the "nodes" column contains the number of states for each of the parent nodes in brackets and the number of child nodes after the arrow.

Bayesian network	CPT	nodes	# values	# elicitations
Pollinator abundance	Honey Bee Abundance	(2,2,2) -> 2	16	10 (+1)
Food security	Household Food Security	(3,2,2) -> 4	36	5 (+2)
Polar bears	PrimPrey: Primary Prey Abundance	(3,3) -> 3	27	1
Polar bears	MrnPry: Marine Prey Base Quality	(3,3) -> 3	27	1
Polar bears	Mrn: Overall Marine Conditions	(4,3) -> 3	36	1
Polar bears	Ice: Overall Sea Ice Conditions	(6,4,3) -> 4	288	1
Polar bears	TerrPry: Overall Terrestrial Prey/Food Availability	(4,4) -> 4	64	1
Polar bears	Terr: Overall Terrestrial Conditions	(4,3) -> 3	36	1
Polar bears	Hab: Overall Habitat Suitability	(3,3) -> 4	36	1
Polar bears	OthMor: Other Mortality or Removal Events	(3,3,3) -> 3	81	1
Polar bears	EvMort: Event-driven Mortality	(3,3,3) -> 3	81	1
Polar bears	ADSur: Adult Survival	(4,3,3) -> 3	108	1
Polar bears	SASur: Sub-adult Survival	(4,3,3) -> 3	108	1
Polar bears	AFBod: Adult Female Body Condition	(4,3,3) -> 4	144	1
Polar bears	Recr: Recruitment	(4,3) -> 4	48	1
Polar bears	Disturb: Sub-Lethal Human Disturbance	(3,3,3,3) -> 3	243	1
Polar bears	BioStr: Other Biotic Stressors	(3,3) -> 3	27	1
Polar bears	CumPop: Cumulative Potential for Persistence	(4,3,3) -> 5	180	1

The *nodes* column contains the BN structure for each CPT. For the Household Food Security CPT, the structure is given by *(3,2,2) -> 4*. This means that there are three parent nodes, of which one parent node has three states and the other two have two states each. The child node has four states. Most of the CPTs that are included in the table have two or three parent nodes, except for one, which has four parent nodes. Most of the nodes included in the CPTs of the Polar Bears network contain three or four states, the nodes included in the Food Security and Bee Abundance CPTs have fewer states.

The selection of fully elicited CPTs covers a broad range of CPT sizes. The smallest CPT included is made up of 16 probabilities, and the largest contains 288 probabilities. In this chapter, each of the included BNs is described in more detail individually. An overview is given on the structure of the BN, the application, and the elicitation process.

5.1. POLLINATOR ABUNDANCE

The first BN application that is considered concerns the abundance of pollinators in the UK (Barons et al., 2018). Figure 5.1 shows the complete BN of the paper. The influence of varroa control, the weather, and the environment on the abundance of different types of pollinators is modeled. Varroa is a parasitic mite, determined to be the key pest affecting honey bees by the experts in the study, and is used as a proxy for the overall disease pressure on honey bees.

A group of eleven experts agreed to participate in the elicitations. Before the elicitation of the probabilities, the experts defined the states of each of the nodes. After the states were clarified, the probabilities to fill the CPTs of the network were elicited using the IDEA protocol, as introduced in Section 3.3.5. All CPTs in the network were fully elicited, summing to a total of 32 values, 16 for the honey bee abundance, and 8 values each for the other child nodes. For each CPT value, a confidence interval and best estimate were asked. Originally, the parent nodes (Varroa control, Weather and Environment) were supposed to have more than two states, but this number was reduced to be able to finish the elicitation in time.

In addition to the probability assessments, the experts were asked to take part in a calibration exercise. It was found that there was no significant difference in calibration scores between the experts. Finally, the EWDM was chosen to parameterize the BN. One of the main conclusions of the paper was that "the quantities provided by the experts show that varroa control has an enormous effect on the abundance of honey bees" (Barons et al., 2018). The weather and environment have a less large effect.

In this thesis, only the CPT of the 'Honey bee abundance' node is considered, as the other CPTs are small enough for full elicitation. For those CPTs, using one of the CPT construction methods of Chapter 4 would not reduce the expert burden significantly.

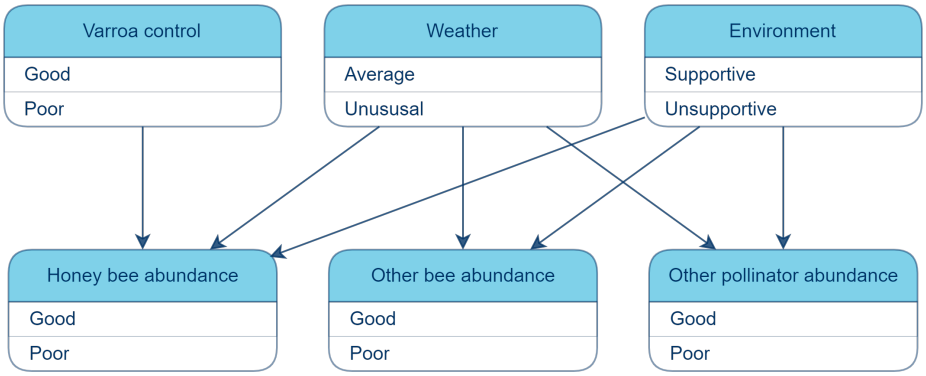


Figure 5.1: Bayesian network about pollinator abundance in the UK (Barons et al., 2018).

5.2. HOUSEHOLD FOOD SECURITY

The next application of BNs has to do with household food security in Victoria, Australia (Kleve & Barons, 2021). The influence of physical access to food, the availability of food, and equivalised income on the level of food security is modeled by the BN in Figure 5.2. This is part of a larger BN to develop a food security inte-

grated decision support system (IDSS). For the nodes of this sub-BN, no data was available to parameterize, thus expert judgment was used.

Similar to the pollinator abundance application, this application also uses the IDEA protocol for eliciting CPTs. A total of five experts participated in the study, where 48 values were elicited to fill the CPT, and additional calibration questions were elicited. The probabilities needed to fill the CPT were elicited using relative frequencies, for example by *Q1: Out of 100 people with high equivalised disposable income and good physical access when food availability is good, how many will be food-secure?* Again, for each CPT value, a best estimate and a 95 % confidence interval were elicited.

The experts' assessments of the second elicitation round were finally aggregated using performance-based weights. It was found that there was one parent node with a dominating effect on the child node over the other parent nodes. "Whilst the effects of physical access and food availability on food security status are significant, household disposable income is by far the strongest determinant." (Kleve & Barons, 2021).

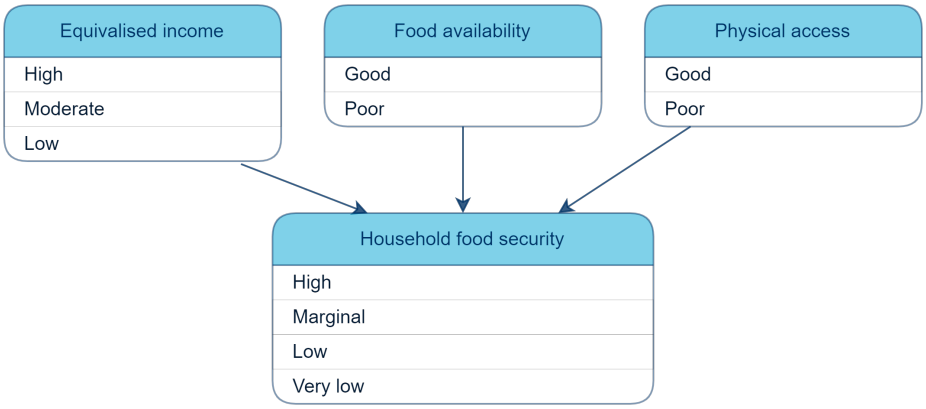


Figure 5.2: Bayesian network about household food security in Victoria, Australia (Kleve & Barons, 2021).

5.3. POLAR BEARS

The final fully elicited BN considered in this thesis studies the relative influence of environmental and anthropogenic stressors on the persistence of polar bears (Atwood et al., 2016). The BN, as shown in Figure 5.3 is significantly larger than the previous examples, with a total of 48 nodes.

In total, 21 CPTs were to be specified by the experts, the other 27 nodes are root nodes specified by uniform marginal distributions. The CPTs were filled by having an Excel file go around a team of 8 experts until a consensus was reached. The elicitation process was not performed using one of the SEJ methods as introduced in Section 3.3, which means that no extra effort was put into preventing biases, such as anchoring, from being present in the data. This also means that no calibration exercise was performed, thus for each CPT, there is only one final assessment.

Most of the included nodes in the BN can be ranked, with the *elevated* or *improved* state being ranked the highest, and the *(greatly) reduced* or *(greatly) decreased* receiving the lowest rank. In some cases, this ranking had to be reversed to have positive influences between the child node and all parent nodes. The *Ecoregion* node cannot be ranked, which means that the three child nodes, *Foraging Sea Ice Distribution*, *Bears on shore*, and *Terrestrial Maternal Den Access*, do not have only ranked parent nodes. Although fully elicited CPTs exist for these nodes, these are not included in the comparison. Additionally, it was found that the CPTs for *BioStr*, *Pol*, and *AntStr* are exactly the same. Therefore, only the *BioStr* CPT is included in Table 5.1.

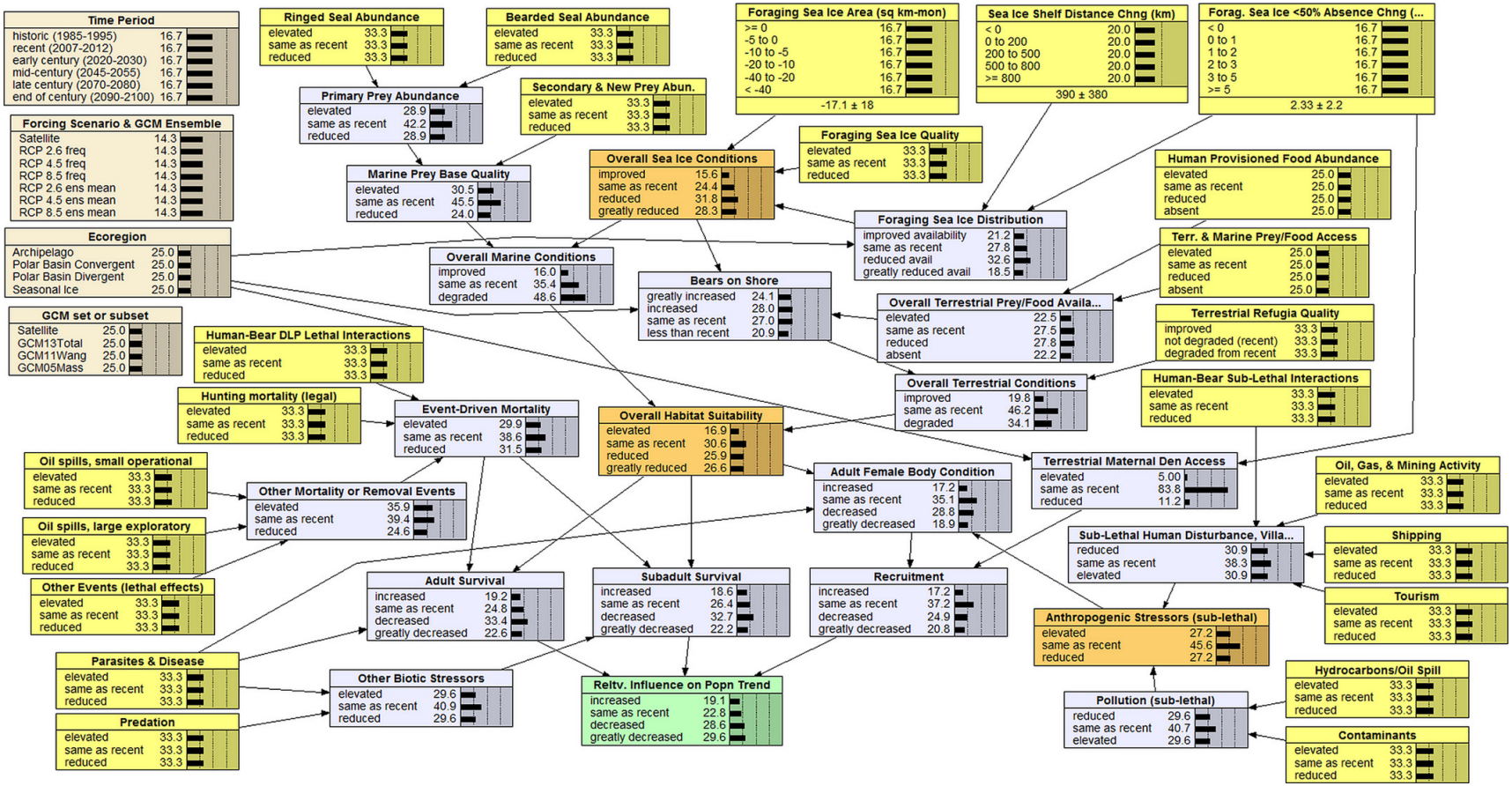


Figure 5.3: Bayesian Network about the relative influence of environmental and anthropogenic stressors on the persistence of polar bears (Atwood et al., 2016).

6

IMPLEMENTATION AND EXTENSIONS OF EXISTING METHODS

Three of the CPT construction methods that were presented in Chapter 4 - InterBeta, RNM, and Functional Interpolation - are used to reconstruct the fully elicited CPTs introduced in Chapter 5. These three methods were chosen because they are applicable to most of the CPTs included in the data, and provide flexibility. The Noisy-OR/MAX models are not included as these can be confusing to elicit and provide less flexibility.

First, for each of these methods, new extensions are presented in this thesis. These extensions were developed as a response to some of the identified limitations of the CPT construction methods as reported in Chapter 4. Then, the implementation of the original methods and the extensions are discussed in detail, and so are the algorithms that were used to find optimal parameters if necessary[1].

6.1. INTERBETA

The first CPT construction method that is tested is InterBeta. Based on some of the limitations and proposed improvements to the method, as mentioned in Section 4.5.1, extensions have been developed in this thesis. These include additional types of mean functions, additionally elicited middle rows, and the "ExtraBeta" version, which not only uses interpolation but also extrapolation. Each of these extensions is introduced in more detail in the following subsection.

Next, the implementation of the method in Python is outlined. This outline is based on the description of InterBeta in Section 4.5. This is followed by a description of the algorithm for finding optimized parameters. The weight parameters are chosen such that they minimize the difference between the previously elicited CPTs and the constructed CPTs.

6.1.1. EXTENSIONS

One of the main limitations of InterBeta is the inflexibility of what rows are to be elicited from the experts, the method calls for the best and worst row of the CPT to be assessed. Two of the extensions that are introduced here address this limitation. First by adding an extra elicited row, an intermediate row between the best and worst row, and secondly by giving freedom to experts to assess a "good" and "bad" row. But first, an additional mean function is introduced as elaborated below.

ADDITIONAL MEAN FUNCTIONS

The arithmetic and geometric mean were previously tested on the Honey Bee Abundance CPT and Household Food Security CPT (Barons et al., 2022). In addition, the paper states that the harmonic mean is also an option.

However, both the harmonic mean and the geometric mean have a large downside, as they equal zero if one of the entries equals zero. Since the weight of the worst state of a node equals zero, there are often zeros involved in calculating row weights. This means that both the harmonic mean and the geometric mean give weight zero to all rows where at least one of the parent nodes is in its worst state.

To solve this problem, the "shifted geometric mean" is introduced. This mean function is based on the geometric mean, but to omit the effect of zero entries, a constant is added to each entry before taking the

[1] The Python implementations of each of the discussed methods and the algorithms to find suitable parameters are available on request.

mean, which is finally subtracted from the calculated mean. The exact formulation of the mean function is given in the next section.

INTERBETA WITH ELICITED MIDDLE ROWS

As suggested by the title, the second extension of InterBeta not only requires the best and worst row to be elicited but also one or more middle rows. This should add more freedom to the beta parameters. In this thesis it is investigated if also eliciting a middle row, which is a row where all of the parents are in an intermediate state between the best and worst, increases the performance of InterBeta.

This particular type of intermediate row is chosen because it is in the middle between the best and worst-case situations. It is also possible to use other rows, for example, all rows where the parent nodes are in one of their extreme states, as the Functional Interpolation method uses. However, this would quickly add to the expert burden, so only adding the middle row is considered in this thesis.

EXTRABETA

The final extension to InterBeta is called ExtraBeta, because it no longer solely relies on interpolation. The extension allows experts to assess scenarios that are within their frames of knowledge. This is in line with the ideas behind the Weighted Sum Algorithm as in Appendix A.3, where experts need to group compatible parent configurations.

Instead of the best and worst row, a good and bad row is elicited. These rows will be in the top half of the CPT and the bottom half of the CPT respectively. The exact details of which rows can be chosen by experts are given in the next section. The rest of the CPT is then constructed by interpolating between the good and bad rows and extrapolating outside of them. The method will have the same weight options as InterBeta: parent weights, parent state weights, and row weights.

One potential limitation of this method is that, if experts do not assess the same CPT rows as input, it is not possible to aggregate their assessments before constructing the full CPT. However, in a previous application of InterBeta, it was found that there does not exist a significant difference between interpolating the aggregated expert assessment or aggregating the interpolated assessments (Barons et al., 2022).

6.1.2. IMPLEMENTATION

Based on the description of InterBeta in Section 4.5, the function `InterBeta` is implemented in Python. The function includes all different additional weight versions of InterBeta and uses the appropriate version based on the given input data. The implementations of the extensions are also introduced in this section. The implementation of these extensions is largely the same as the original InterBeta implementation, thus only the differences will be highlighted.

InterBeta The function first sets the parameters depending on the input given. If no input is given for a type of parameter, they are set to the default values:

- best row: $\alpha = 4,\ \beta = 1,\ \mu = 0.8,\ \sigma^2 = \frac{2}{75}$,
- worst row: $\alpha = 1,\ \beta = 4,\ \mu = 0.2,\ \sigma^2 = \frac{2}{75}$,
- parent weights: $w_i = 1$ for all $i = 1, ..., n$,
- parent state weights: $\omega_{i,1} = w_i,\ \omega_{i,2} = w_i - \frac{w_i}{s_C - 1},\ \omega_{i,3} = w_i - \frac{2w_i}{s_C - 1},\ ...,\ \omega_{i,s_C} = 0$.

In case elicited rows are given as multinomials, the Beta distribution parameters α, β are chosen by the method of moments, which are then further optimized by a greedy search algorithm as described in Section 4.5.

In Figure 6.1 the improvement of the KL-divergence per iteration of the beta fitting function is shown. The multinomials that are used in this example are based on CPT rows that are simulated using correlation structures as given in Section 8.1, with three parent nodes that each have three states. The number of states for the child node is varied from two to six. The figure shows the results of fitting beta distributions to the rows of 200 CPTs, each containing 27 rows. After about 900 iterations, the KL-divergence stopped improving, therefore 1000 iterations were chosen to fit the beta distributions.

Once the beta distribution parameters have been found, the next step is to set the weights. If parent weights $w_i > 0$ are specified, they are used for the determination of the parent state weights, which are in turn used to determine the row weights. These row weights are in $[0, 1]$, where the worst row receives weight 0

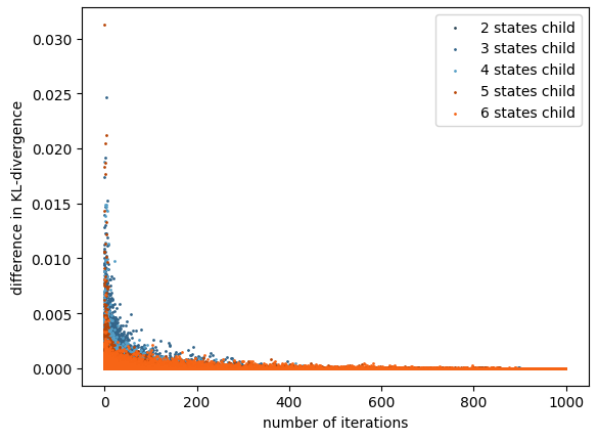


Figure 6.1: The improvement of the KL-divergence for each iteration of the beta fitting function, when applied to simulated multinomials with 2 to 6 states.

and the best row receives weight 1. Only when the row weights are given as input by the experts, they can be outside of $[0,1]$, but this should be done with care to make sure that the beta distribution parameters do not become negative values. If input is given for the parent state weights, it is made sure that the state with the largest weight, which is generally the best state, receives a weight equal to the parent node weight. The worst state receives weight zero.
In case the row weights are not given as input they can be calculated by different mean functions. Apart from the previously tested arithmetic and geometric means, the harmonic mean and a shifted geometric mean can be used as well:

$$\begin{aligned}
\text{arithmetic mean:} &\quad \frac{1}{n}\sum_{i=1}^{n}\omega_{i,j},\\
\text{geometric mean:} &\quad \left(\prod_{i=1}^{n}\omega_{i,j}\right)^{\frac{1}{n}},\\
\text{shifted geometric mean:} &\quad \left(\prod_{i=1}^{n}(\omega_{i,j}+\delta)\right)^{\frac{1}{n}}-\delta,\\
\text{harmonic mean:} &\quad \frac{n}{\sum_{i=1}^{n}\frac{1}{\omega_{i,j}}}.
\end{aligned}$$

For both the geometric mean and the harmonic mean it holds that, if one of the state weights equals zero, the resulting row weight will also be zero. This would result in all scenarios where at least one of the parent nodes is in its worst state to receive weight zero. To omit the influence of the zero values for the geometric mean, the shifted version can be used, where δ is added to all state weights, and finally subtracted from the result. Other methods exist, which either ignore the zeros completely or optimize the value of δ to each data set (de la Cruz & Kreft, 2018), but these are not applicable for such small sets of entries. In this thesis, the value of $\delta = 1$ is chosen based on trial and error. Figure 6.2 shows the row weights calculated with each of the mean functions, for example, a BN with three parent nodes that each have three states, where the parent and state weights are the default values. The figure emphasizes that the weights are zero for a large chunk of the CPT when either the geometric mean or harmonic mean is used. The shifted geometric mean looks like a slightly smoothed version of the arithmetic mean.

Once the row weights have been calculated, beta distribution parameters are determined for each row using Equation (4.11), or by interpolating the mean and variance in the same way. The weights are then used to determine α and β by the method of moments (4.9). For each row, let α_k, β_k be the corresponding parameters. Then, the cdf of beta distribution, denoted by $F(x;\alpha_k,\beta_k)$, is discretized to calculate the CPT row:

$$\mathbb{P}(X_C = x_C^i \,|\, (X_1 = x_1, \ldots, X_n = x_n)_k) = F\left(\frac{s_c+1-i}{s_C};\alpha_k,\beta_k\right) - F\left(\frac{s_c-i}{s_C};\alpha_k,\beta_k\right), \tag{6.1}$$

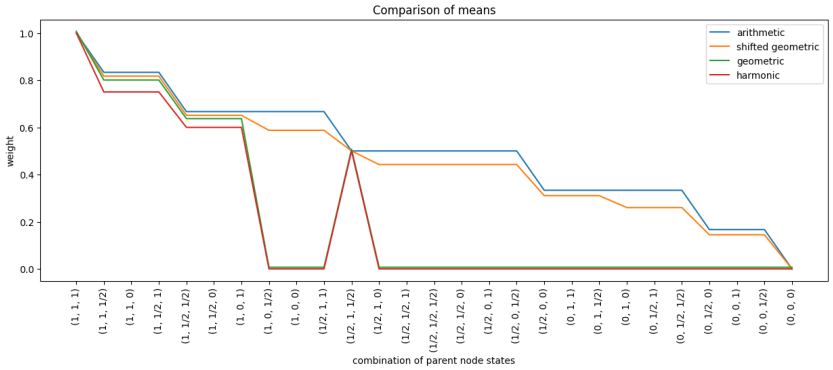


Figure 6.2: Example: calculated weights when the arithmetic, geometric, shifted geometric, or harmonic mean is used.

where $(X_1 = x_1, ..., X_n = x_n)_k$ is the combination of parent node states corresponding to the kth row, and $i = 1, ..., s_C$, with s_C being the number of child states. Figure 6.3 shows how the domain of the child node distribution $[0,1]$ is divided in s_C sub-intervals of equal width. Then the discretization is done by calculating the probability density within each sub-interval.

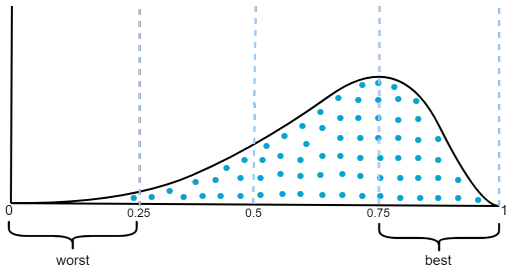


Figure 6.3: Example of the discretizing method for a beta distribution for a child node with four states.

InterBeta with middle rows elicited In addition to the best and worst row, a middle row is needed as input. If all parent nodes have at least three states, the middle row is the combination of all parent nodes being in their second state. If all parent nodes have more than three states, more middle rows can be given as input. The number of middle rows that can be given as input is equal to $\min_{i=1,..,n}(s_i) - 2$.

The weight assignment then proceeds in the same way as for the original InterBeta, which means that also the weights for the middle rows are not set. The only difference comes when the parameters are interpolated, a piecewise linear interpolation is used, where the middle rows are set points.

ExtraBeta For the ExtraBeta method, two rows are required as input, a "good" and a "bad" row. A good row is defined to be a row with a default weight larger than 0.5. Similarly, a bad row is defined to be a row with a default weight of less than 0.5. Additionally, the good and bad row should be specified with different multinomials or parameters, and the mean of the good row should be larger than the mean of the bad row.

Using the following equations, α parameters are found for the best and worst row of the CPT:

$$\alpha_{\uparrow} = \alpha_{good} + (\alpha_{good} - \alpha_{bad}) \cdot \frac{w_{\uparrow} - w_{good}}{w_{good} - w_{bad}}, \tag{6.2}$$

$$\alpha_{\downarrow} = \alpha_{bad} - (\alpha_{good} - \alpha_{bad}) \cdot \frac{w_{bad} - w_{\downarrow}}{w_{good} - w_{bad}}, \tag{6.3}$$

where initially $w_{\uparrow} = 1$ and $w_{\downarrow} = 0$. If these values result in negative α, the $w_{\uparrow}, w_{\downarrow}$ are gradually decreased, and increased, until positive $\alpha_{\uparrow}, \alpha_{\downarrow}$ are found. The same method is used to find best values for $\beta_{\uparrow}, \beta_{\downarrow}$.

Once the beta distribution parameters are found for the best and worst row, the rest of the CPT is calculated in the same way as for InterBeta.

6.1.3. ALGORITHM

The method that is used to optimize weights for the InterBeta method, returns the optimized weights for each different version. For each type of weight, first, initial weights are found and then optimized to fit the true CPT. The algorithm starts with the row weights, which are then used as a basis to also find initial values for the parent and parent state weights.

Row weights The initial row weights are determined based on how much the mean of the child node distribution shifts between rows. Thus for each CPT row, the means of the child node distribution will be compared. The child node distribution for each row k of a CPT is given by a multinomial $(p_{1,k},...,p_{s_c,k})$. Where each probability $p_{i,k}$ is assigned to an interval $[a_i,b_i)$ that divides the unit interval $[0,1]$ in s_C equally sized intervals. The mean μ of the child node distribution for each row k is then determined in the following way:

$$\mu_k = \sum_{i=1}^{s_C} p_{i,k} \cdot \frac{b_i - a_i}{2}. \tag{6.4}$$

For the initial row weights, the difference between the mean of the child node distribution for row k and the worst row is used. This difference is then divided by the difference between the means of the best and worst row to normalize the weights between zero and one:

$$w_k = \frac{|\mu_\downarrow - \mu_k|}{|\mu_\downarrow - \mu_\uparrow|}.$$

In the Python implementation, the weights are rounded to two decimals.

The row weights are then optimized one at a time. To ensure the optimal row weight is not missed, the row weights that are tested range between $w_k - 2$ to $w_k + 2$. This means that also negative row weights and row weights larger than one are tested, as long as they result in positive α, β parameters.

Using these possible row weights, beta distribution parameters are calculated, which are then used to calculate the CPT row using Equation (6.1). By calculating the KL-divergence defined in Equation (4.12), the constructed row is compared against the elicited (true) CPT row. The row weight that produces the KL-divergence closest to zero is chosen to be optimal.

Parent weights The initial parent weights are determined based on the optimized row weights. The intuition behind these weights is that they are based on the relative shift of the mean when the parent node switches between the best and worst state, while the other nodes remain in the same state. Thus, the relative weight of parent node i, PW_i, is the optimized row weight of the row $(X_1 = x_1^\downarrow,...,X_i = x_i^\uparrow,...,X_n = x_n^\downarrow)$. Only positive weights are accepted, if a negative or zero weight is found, this is replaced by 0.01. The weights are normalized such that the smallest weight equals one and are rounded to the nearest half, such that they conform to possible elicited weights, as it is unrealistic for experts to give more precise weights than that.

For the optimization of the weights, a grid search approach is chosen. The smallest initial parent weight remains fixed, whilst the others are varied over a grid with range $[PW_i - 5, PW_i + 5]_{i=1}^{n-1}$ and integer steps. The parent weights that lead to the smallest KL-divergence between the constructed CPT and the true CPT are chosen. If these newly found weights are on the edge of the grid search range, the grid search is repeated using these newly found weights. After this first grid search process is finished, a refined grid search is performed around the optimally found weights $P\hat{W}_i$, with a grid step of 0.5 and range $[P\hat{W}_i - 0.5, P\hat{W}_i + 0.5]_{i=1}^{n-1}$.

Parent state weights The same principle as for the parent weights can be applied to find the initial parent state weights. In this case, the shift of the mean of the child node distribution is measured for all CPT rows for which one parent node moves from its worst state to another state, whilst the other nodes are in the worst state. In practice, the initial parent state weight $PSW_{i,j}$ for node i and state j is equal to the optimized row weight of the row $(X_1 = x_1^\downarrow,...,X_i = x_i^j,...,X_n = x_n^\downarrow)$, where $j = 1,...,s_i$. The found weights are then transformed such that the weight of the worst state is zero, and the weight of the best state is equal to the previously found optimized parent weight.

Finally, for the parent state weights, a greedy evolutionary search algorithm is used. During the search, the weights of the best and worst state remain fixed, while the other state weights are changed iteratively in the following way:

$$P\tilde{S}W_{i,j} = w + 0.05 \cdot Z, \quad \text{where } Z \sim \mathcal{N}(0,1),$$

where each $P\tilde{S}W_{i,j}$ is rounded to two decimals. Only improvements in the KL-divergence between the constructed CPT and the true CPT are accepted, otherwise the adjusted weight is discarded.

Extensions For the extensions where middle rows are elicited, the algorithm for finding optimized weights does not change. For the ExtraBeta method, it does change slightly. As before, first, the optimal row weights are determined on a row-by-row basis, starting with the best and worst row parameters using Equation (6.3). The weights for these are then set to one and zero, after which the remaining row weights are determined in the same fashion as was done for InterBeta.

The algorithm to find the parent weights and parent state weights are the same as for InterBeta.

6.2. RANKED NODES METHOD

The original version of the Ranked Nodes Method (RNM) uses trial and error to find fitting weights and a variance parameter. The implementation of this method will be outlined in this section, alongside with the implementation of an extended version. The extended version will be introduced in this section, and the difference in implementation with the original version is discussed.

6.2.1. AUTORNM

The extended version of RNM proposed in this thesis does not depend on experts finding suitable weights by trial and error, instead, these weights are optimized to fit a set of elicited rows. This explains the name AutoRNM as RNM with automated weight determination. This version requires a set of elicited CPT rows as input instead of weight and variance parameters. The suitable weight and variance parameters are then fit to these elicited rows.

The algorithm that is used by AutoRNM is the same as is used for RNM to reconstruct previously elicited (true) CPTs. The algorithm is discussed after the implementation section.

6.2.2. IMPLEMENTATION

For RNM, two methods are implemented in Python, the original version as described in Section 4.4.1, and AutoRNM. AutoRNM makes use of the original RNM implementation after finding optimal weights and variance parameters.

Original RNM As input, the method requires a set of weights for the parent nodes, a variance parameter for the entire child node distribution, and a weight function to be specified. In addition, a parameter p can be specified which is the number of sample points that are used to calculate the CPT values.

The method starts by determining the state intervals of the parent nodes and the child node. As previously described, a node with s states will be given state intervals $[0, \frac{1}{s}), [\frac{1}{s}, \frac{2}{s}), ..., [\frac{s-1}{s}, 1]$, from the worst to the best state. The intervals have an equal width with their union covering the unit interval.

Once the state intervals have been assigned, the method goes through all possible combinations of parent node states. For each combination, p equidistant sample points $z_{i,k}$ are taken from each parent node state interval, so in total $n \cdot p$ sample points are necessary for each combination. The method then iterates through the set of sample points $\{(z_{1,k}, ..., z_{n,k})\}_{k=1}^{p^n}$ and calculates the mean values μ_k using one of the Equations (4.2), (4.3), (4.4), or (4.5). The values of the CPT are then calculated using Equation (4.6), which uses the truncated normal distribution with the calculated means.

Once the CPT values have all been calculated, all zero values are replaced by 0.001 and the rows are normalized such that they sum to one.

AutoRNM AutoRNM is implemented as an extension of the original RNM. For AutoRNM, no weights or variance parameters are used as input, but instead, the method requires $n+1$ or $2(n+1)$ CPT rows to be elicited from experts. If the weight function is not known beforehand, the function requires $2(n+1)$ CPT rows as input, with which all weight functions can be tested. If the weight function is known beforehand, $n+1$ rows are enough as input. The method then uses an optimization method to find fitting weights, a variance parameter, and a fitting weight function that can be used for the original RNM.

The scenarios S_a and S_b are defined as:

$$S_a = \{(X_1 = x_1, ..., X_n = x_n) \,|\, x_i = x_i^{\uparrow} \forall\, i \neq k, x_k = x_k^{\downarrow}\}_{k=1}^{n}, \tag{6.5}$$

$$S_b = \{(X_1 = x_1, ..., X_n = x_n) \,|\, x_i = x_i^{\downarrow} \forall\, i \neq k, x_k = x_k^{\uparrow}\}_{k=1}^{n}. \tag{6.6}$$

S_a is the set of scenarios where all parent nodes are in their best state except for parent node k, which is in its worst state. S_b is the set of scenarios where all parent nodes are in their worst state, but only node k is in its best state. Additionally, define S_a^+ to be the set S_a plus the scenario where all parent nodes are in their best state, and S_b^+ the set containing S_b and the scenario where all parent nodes are in their worst state.

Then the input for the different weight functions is the following:

- **WMEAN**: S_a^+, S_b^+, or $S_a^+ \cup S_b^+$,
- **WMIN**: S_a^+,
- **WMAX**: S_b^+,
- **MIXMINMAX**: S_a^+, S_b^+, or $S_a^+ \cup S_b^+$.

The method then uses these elicited rows to find fitting weight and variance parameters. This optimization algorithm is similar to the algorithm that is used to find optimal weights for the original RNM, which will be described in the next section. The main difference is that, when the parameters are optimized, the performance is measured only on the elicited rows instead of the full CPT.

Finally the extended RNM function uses the implementation of the original RNM to calculate the CPT.

6.2.3. ALGORITHM

The algorithm to find suitable weight and variance parameters starts by finding an initial value for the variance based on the Method of Moments. For each row of the true CPT, which is described by a multinomial with s_C values: $(p_{1,k}, ..., p_{s_C,k})$, the mean ($\mu$) and the variance ($\sigma^2$) are calculated as follows:

$$\mu = \sum_{i=1}^{s_C} p_{i,k} \cdot \frac{b_i - a_i}{2}, \tag{6.7}$$

$$\sigma^2 = \sum_{i=1}^{s_C} \left(\frac{b_i - a_i}{2} - \mu\right)^2 \cdot p_{i,k}, \tag{6.8}$$

where $[a_i, b_i]$ are the state intervals of the child node.

The next step is to find weight parameters. As described by Laitila and Virtanen, 2016, there are certain interpretations of the weights linked to the state intervals. Let $z = \{z_1, ..., z_n\}$ be a combination of sample points of the parent node's state intervals, and let $z' = \{z_1, ..., z_k + \Delta z_k, ..., z_n\}$ be the combination where only the kth sample point is changed. Then for **WMEAN**, it is found that:

$$w_i^N = \frac{w_i}{\sum_{j=1}^{n} w_j} = \frac{\Delta\mu}{\Delta z_k},$$

which means that the normalized weight w_k^N can be approximated by using the difference in the mean between two combinations of parent states where only one state has changed. In the algorithm, the scenarios S_a and S_b in Equations (6.5) and (6.6) are compared to the scenario where all parent nodes are in their most positive state (S_a^+, S_b^+) and where all parent nodes are in their most negative state (S_a^-, S_b^-).

Let S_a^i be the scenario where all parent nodes are in their most positive state, except for node i, which is in the most negative state. Similarly, S_b^i is defined such that only parent node i is in the most positive state. The initial weight parameters can then be calculated in the following way:

$$\hat{w}_i = \frac{|\mu_{S_+} - \mu_{S_a^i}| + |\mu_{S_-} - \mu_{S_b^i}|}{2(\tilde{z}_{max} - \tilde{z}_{min})},$$

where $\tilde{z}_{max} - \tilde{z}_{min}$ is the distance between the most positive and most negative state for parent node X_i. The state intervals for the most positive and most negative states are known to be $[0, \frac{1}{s_C})$ and $[\frac{s_C - 1}{s_C}, 1]$, the

midpoints of each of these intervals are taken for $\tilde{z}_{min}$ and $\tilde{z}_{max}$ respectively. Thus, the following values are taken:

$$\begin{cases} \tilde{z}_{min} &= \frac{1}{2s_C} \\ \tilde{z}_{max} &= 1 - \frac{1}{2s_C} \end{cases} \quad \Rightarrow \quad \hat{w}_i = \frac{|\mu_{S_+} - \mu_{S_a^i}| + |\mu_{S_-} - \mu_{S_b^i}|}{2 - \frac{2}{s_C}}. \tag{6.9}$$

Next, for weight function **WMIN**, let z_i be such that:

$$z_i \leq \frac{1}{n} \sum_{j=1}^{n} z_j \leq z_l, \quad \forall l = 1, ..., n, l \neq j. \tag{6.10}$$

So, the sample point z_i must be smaller or equal to the average of the combination of sample points that describes the scenario, which in turn needs to be smaller or equal to all other sample points. If this condition is attained, the following relationship exists:

$$w_i = 1 + \frac{\sum_{j=1}^{n} z_j - n\mu}{\mu - z_i}.$$

To apply this relationship to find suitable initial weights for $WMIN$, the scenarios S_a can be used, as these match the condition in Equation (6.10). Then, the initial weights can then be calculated by:

$$\hat{w}_i = 1 + \frac{(s_C - 1)\tilde{z}_{max} + \tilde{z}_{min} - (s_C - 1)\mu_{S_a^i}}{\mu_{S_a^i} - \tilde{z}_{min}},$$

where $\tilde{z}_{min}$ and $\tilde{z}_{max}$ are as in Equation (6.9).

Closely related to $WMIN$ is the weight function **WMAX**, which needs z to be such that:

$$z_l \leq \frac{1}{n} \sum_{j=1}^{n} z_j \leq z_i, \quad \forall l = 1, ..., n, l \neq j. \tag{6.11}$$

In this case, the sample point z_i needs to be the largest instead of the smallest. If this condition is attained, the following relationship exists:

$$w_i = 1 + \frac{n\mu - \sum_{j=1}^{n} z_j}{z_i - \mu}.$$

The scenarios S_b are used, as these fulfill the condition of Equation (6.11). This leads to the following equation to find initial weights for $WMAX$:

$$\hat{w}_i = 1 + \frac{(s_C - 1)\mu_{S_a^i} - (s_C - 1)\tilde{z}_{max} + \tilde{z}_{min}}{\tilde{z}_{min} - \mu_{S_b^i}},$$

where $\tilde{z}_{min}$ and $\tilde{z}_{max}$ are as in Equation (6.9).

Finally, for weight function **MIXMINMAX**, the relative weights can be found by using the following:

$$\begin{cases} w_{MIN} = \frac{\max_{i=1,...,n}\{z_i\} - \mu}{\max_{i=1,...,n}\{z_i\} - \min_{i=1,...,n}\{z_i\}}, \\ w_{MAX} = 1 - w_{MIN}. \end{cases}$$

In this case, only one scenario would be enough to find weights, but to filter out any possible oddities (e.g., the effect of a dominant parent node), both sets of scenarios S_a and S_b can be used. So, the set of scenarios that is used is $S_{ab} = S_a \cup S_b$, which means that for all scenarios the $\tilde{z}_{max}$ and $\tilde{z}_{min}$ remain the same. This results in the following equation:

$$\begin{cases} w_{MIN} = \frac{1}{2s_C} \sum_{i=1}^{2s_C} \frac{\tilde{z}_{max} - \mu_{S_{ab}^i}}{\tilde{z}_{max} - \tilde{z}_{min}}, \\ w_{MAX} = 1 - w_{MIN}. \end{cases} \tag{6.12}$$

Once both initial values are found for the variance and the weights, the feasibility is checked. If one of the found initial weights for a weight function is negative, the weight function is declared to be infeasible. If the weights are feasible, they are optimized using a grid search algorithm, in the same way as the parent weights were optimized for InterBeta. After the optimized weights are found, the variance is optimized using a greedy search algorithm. Iteratively, the variance is varied slightly, and only improvements in the KL-divergence are accepted. The algorithm stops after a set number of iterations.

When weights are needed for the original RNM, the KL-divergence is measured between the constructed CPT by RNM and the complete true CPT. For the extended version, the same process is used, but the KL-divergence is only measured between the elicited rows of scenarios S_a, S_b. Another main difference in the optimization process for the two methods is that for the original method the weights are rounded to halves, whereas for the extended version, this is not the case. This is done to emulate the real situation, where experts are likely to not give more precise weights than halves.

6.3. FUNCTIONAL INTERPOLATION

The final CPT construction method that is included in this comparative study is Functional Interpolation. Unlike the other two methods, this method only requires elicited CPT rows as input, which means that there is no algorithm needed to find optimal parameters. In addition to the normal distribution which was the distribution used in the paper by Podofillini et al., 2014, two other distributions are implemented, which are introduced in the next section. The section goes on to discuss the implementation of the method in Python based on the description in Section 4.6.

6.3.1. EXTENSIONS

In addition to the normal distribution, also the truncated normal distribution and the beta distribution will be considered for the Functional Interpolation method. These two distributions are chosen because they are used for RNM and InterBeta. The beta distribution should give more flexibility than either of the normal distributions, as it can also represent bimodality. As for InterBeta, it is possible to interpolate the α, β or the mean and variance of the beta distribution.

The implementation of the other distributions does not change much from the original method, other than using different fitting methods to the elicited multinomials. These differences are highlighted in the next section.

In addition to using different distributions, it is also possible to use parent weights, or even parent state weights, like for InterBeta. This is not investigated in this thesis as Functional Interpolation is a method that poses a relatively large burden on the experts.

6.3.2. IMPLEMENTATION

As previously mentioned, the Functional Interpolation method only requires elicited CPT rows as input. The specific rows that need to be elicited are those where either all parent nodes are in their most positive state, all are in their most negative state, or one of the scenarios in S_a or S_b as described for the extended RNM in Equations (6.5) and (6.6). The remaining, unelicited, rows are then constructed in three steps: fitting a distribution to the elicited rows, interpolating the found distribution parameters for the other rows, and then discretizing the distributions with the interpolated parameters again.

The process for fitting the normal distribution is based on using the pdf instead of the cdf. For each elicited CPT row, the assessments are given as a multinomial with each probability assigned to the rank of the child state, so, the multinomial $(p_{1,k}, ..., p_{s_C,k})$ would be corresponding to the values $(1, ..., s_C)$. Using the method of moments a mean and variance can be calculated. The normal distribution with the found mean and variance can then be discretized by calculating the probability density at each value, which is then normalized such that it sums to one. The mean and variance parameters are then optimized by minimizing the MSE between the true CPT row and the discretized CPT row. In Figure 6.4 an example is shown for a child node with three states. If the normal distribution was fit exactly, the pdf would go through each of the points perfectly.

As opposed to the normal distribution, the truncated normal and beta distribution are discretized based on the cdf. For the truncated normal and beta distribution, it is assumed that the multinomial corresponds to the intervals $([0, \frac{1}{s_C}), ..., [1 - \frac{1}{s_C}, 1])$. The distributions are then discretized by calculating the probabilities $\mathbb{P}(a_i < X < b_i)$ for each interval $[a_i, b_i)$, like in Figure 6.3. The parameters are then optimized by minimizing

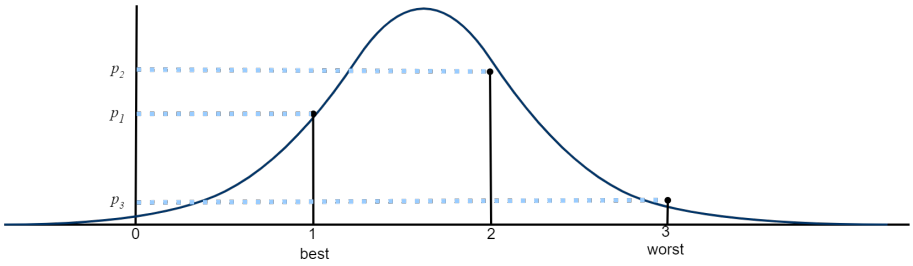


Figure 6.4: Example of a probability density function of a normal distribution fit to a multinomial (p_1, p_2, p_3).

the KL-divergence between the discretized CPT row and the true CPT row.

After the parameters are found for the elicited rows, these are interpolated to find the parameters for the remaining rows. Since the elicited rows are all combinations of the parent nodes being in one of their extreme states, a hypercube can be used where the parameter values are located at each of the corners. Let the worst state be zero and the best state be 1, then denote a distribution parameter by $\rho_{\tilde{x}_1,\ldots,\tilde{x}_n}$. Then $\rho_{0,\ldots,0}$ is the parameter of the row where all parent nodes are in their most negative state.

For example, for $n = 3$ it would become:

$$\rho_{\tilde{x}_1,\tilde{x}_2,\tilde{x}_3} = \rho_{000}\cdot(1-\tilde{x}_1)\cdot(1-\tilde{x}_2)\cdot(1-\tilde{x}_3) + \rho_{100}\cdot\tilde{x}_1\cdot(1-\tilde{x}_2)\cdot(1-\tilde{x}_3) + \ldots + \rho_{111}\cdot\tilde{x}_1\cdot\tilde{x}_2\cdot\tilde{x}_3,$$

where $\tilde{x}_i = \frac{s_i - 1 - \text{rank}(x_i)}{s_i} \in [0,1]$ is the value for the state x_i of parent node i which has s_i states in total. So, everywhere the data point has a 0 in the subscript, it corresponds to a term $(1-\tilde{x}_i)$. Everywhere the data point has a 1 in the subscript it pairs with the term $\tilde{x}_i$. For the three-dimensional case, a visual representation of the interpolation is shown in Figure 6.5. As for InterBeta, it is possible to interpolate the α, β or the mean and variance of the beta distribution.

When the distribution parameters are determined for each row, the CPT values are calculated by discretizing the distributions as described in the first step. Thus, when the normal distribution is used we find that for row k:

$$\mathbb{P}\left(X_C = x_C^i \,|\, (X_1 = x_1, \ldots, X_n = x_n)_k\right) = f\left(\text{Rank}(x_c^i); \mu_k, \sigma_k^2\right).$$

For the truncated normal and beta distribution, Equation (6.1) or a similar version is used. Finally, any probabilities equal to zero are replaced by 0.0001, then the found probabilities are normalized such that they sum to one.

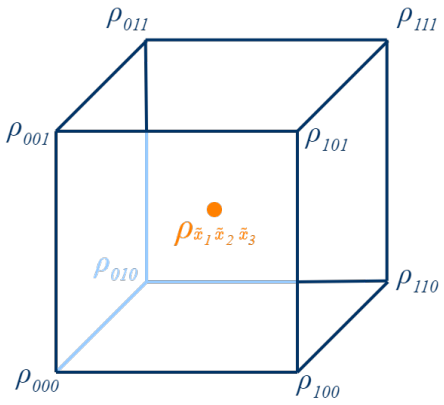


Figure 6.5: Visual guide for interpolation on a hypercube, example for three dimensional case.

7

Performance of CPT construction methods on expert-elicited CPTs

Each of the introduced methods and their extensions from Chapter 6 are used to reconstruct the fully elicited CPTs of Chapter 5. This chapter will first treat each CPT construction method separately, where the results of all extensions of each method are compared. The InterBeta section also contains an analysis of interpolated beta distribution parameters on a row-by-row basis.

The section continues with a comparison of the results of all three CPT construction methods applied to the fully elicited CPTs. This comparison will not only assess the performance in terms of accuracy but also in terms of the burden each method places on experts.

7.1. InterBeta

In this section, for each version of InterBeta, the different choices for the mean function and parameters that are interpolated are compared. Either the α/β or the mean/variance of the Beta distribution are interpolated using one of the following mean functions: arithmetic, geometric, shifted geometric, or harmonic. First, the original InterBeta method will be discussed, after which the variation where also the middle rows are elicited is analyzed and finally results of ExtraBeta are presented.

Original InterBeta

To start, Figures 7.1 to 7.4 show the mean KL-divergence between the constructed CPT and the fully elicited (true) CPT for the best and worst, parent weights, parent state weights and row weights versions of InterBeta. In Table C.2, the KL-divergence and percentage of agreement are given for each best-performing mean function and set of interpolation parameters.

It is noteworthy that for each version of InterBeta that requires more specific weights as input, the performance improves. For each version with more input parameters, these can be chosen such that the resulting row weights are equal to the row weights determined by a version requiring fewer weight parameters. Thus, the following holds:

$$D_{KL}(\text{CPT}_{bw} \,\|\, \text{CPT}_{true}) \geq D_{KL}(\text{CPT}_{pw} \,\|\, \text{CPT}_{true}) \geq D_{KL}(\text{CPT}_{sw} \,\|\, \text{CPT}_{true}) \geq D_{KL}(\text{CPT}_{rw} \,\|\, \text{CPT}_{true}),$$

where bw is best and worst, pw is parent weights, sw is parent state weights, and rw denotes the row weights version of InterBeta. Since this relationship is known beforehand, the question is not which version of InterBeta is the best. Instead, the relative improvement between each version with respect to the expert burden is of interest. This question will be revisited in Section 7.4. The other remaining topic of interest is which mean function performs best, in combination with which set of interpolation parameters. This topic has been addressed previously, by comparing the arithmetic mean and geometric mean (Barons et al., 2022). In this section, the topic will be revisited, where the shifted geometric mean and harmonic mean are taken into consideration as well.

First, consider the best and worst version, as shown in Figure 7.1. For all three BNs: Pollinator Abundance, Food Security, and Polar Bears; a clear difference can be seen between using the different mean functions.

In most cases, the arithmetic or shifted geometric mean performs the best, which could have been expected when Figure 6.2 is considered. For the Polar Bears BN, it also becomes clear that it is better to interpolate the α/β than the mean/variance. This difference is less significant for the Pollinator Abundance CPT and Food Security CPT.

The results of the parent weights version of InterBeta are presented in Figure 7.2. In this case, n extra parameters are added as input, which is the number of parent nodes. The performance results have not changed much from the best and worst version. For the Pollinator Abundance BN, it remains clear that the geometric mean and harmonic mean perform the worst. But the difference between the shifted geometric mean and arithmetic mean, and interpolating the α/β versus the mean/variance remains very small. A little more has changed for the Food Security CPTs, where the arithmetic mean started outperforming the shifted geometric mean. For the Polar Bears BN, the preference for interpolating the α/β remains. There is a slightly larger improvement for the arithmetic mean than for the shifted geometric mean. In general, the arithmetic mean and shifted geometric mean, where the α, β are interpolated, remain the best-performing combinations.

Now focusing on the performance of the state weights, see Figure 7.3. Since the parent nodes in the Pollinator Abundance CPT do not contain more than two states, the results are equal to the results of the parent weights version. For the Food security BN there is one parent node with three states, which means that there is one extra parameter as input for the parent state weights in comparison to the parent weights. This has had the most significant effect on expert E, where arithmetic mean with α/β interpolation started performing significantly better. For the Polar Bears BN, the performance change is case dependent. There is no clear pattern to what characteristics of a CPT constitutes to the largest improvement in performance. Still, for most cases, the arithmetic or shifted geometric mean with α/β interpolation performs the best. The performance of these two combinations is very similar for most CPTs. Only for three CPTs, the arithmetic mean using the mean/variance as interpolation parameters comes out the best.

Finally, the performance of the row weights version is analyzed, see Figure 7.4. First note that the order of magnitude of the y-axis has decreased significantly for this method. The type of mean function no longer has an influence on the performance, as the row weights are pre-determined. The performance differences that are visible are likely due to fitting errors, resulting from fitting the beta distribution to the best and worst CPT rows. As the other CPT rows are calculated, these fitting errors are propagated. The choice of interpolation parameters remains an influence on the performance results. There is no clear "winner". However, for the larger CPTs included in the Polar Bears BN, which have at least 81 values, interpolating the α, β shows better results than interpolating the mean and variance. For the smaller CPTs, the best-performing set of interpolation parameters alternates between being α/β and mean/variance.

In general, there seems to be a preference for using either the arithmetic mean or shifted geometric mean. When the best and worst version is used, there is a clear preference to interpolating the α, β, but as the InterBeta version becomes more flexible with more parameters, this slightly better performance fades. So, overall, it can be noted that the performance of the mean function and the interpolated parameters depends on the CPT. So this leads to the questions:

- When is it better to use the arithmetic mean or the shifted geometric mean?
- When is it better to interpolate the α, β or the mean and variance?

In Section 7.1.1 a first attempt to answer these questions is made. A closer look is taken at the parameters of the beta distribution for each separate CPT row and what the effect is of interpolation between the parameters of the best and worst row. This will also help to showcase why the variations of InterBeta perform so differently.

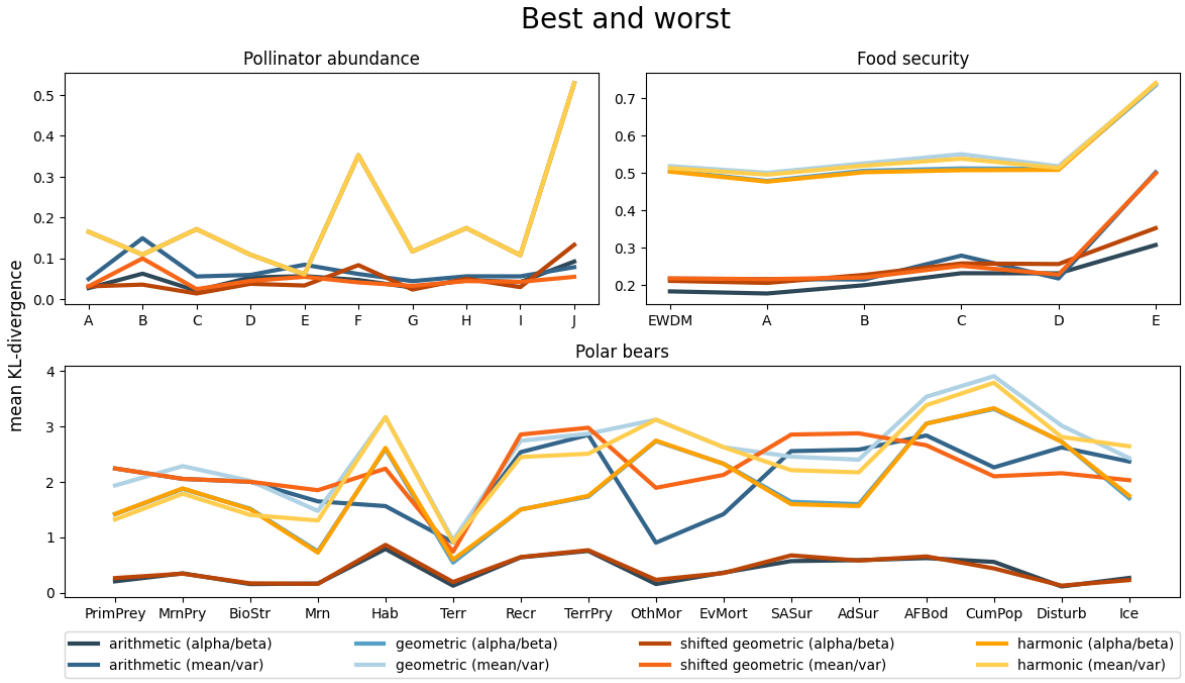


Figure 7.1: Mean KL-divergences of constructed CPTs using different mean functions for the 'best and worst' version of InterBeta.

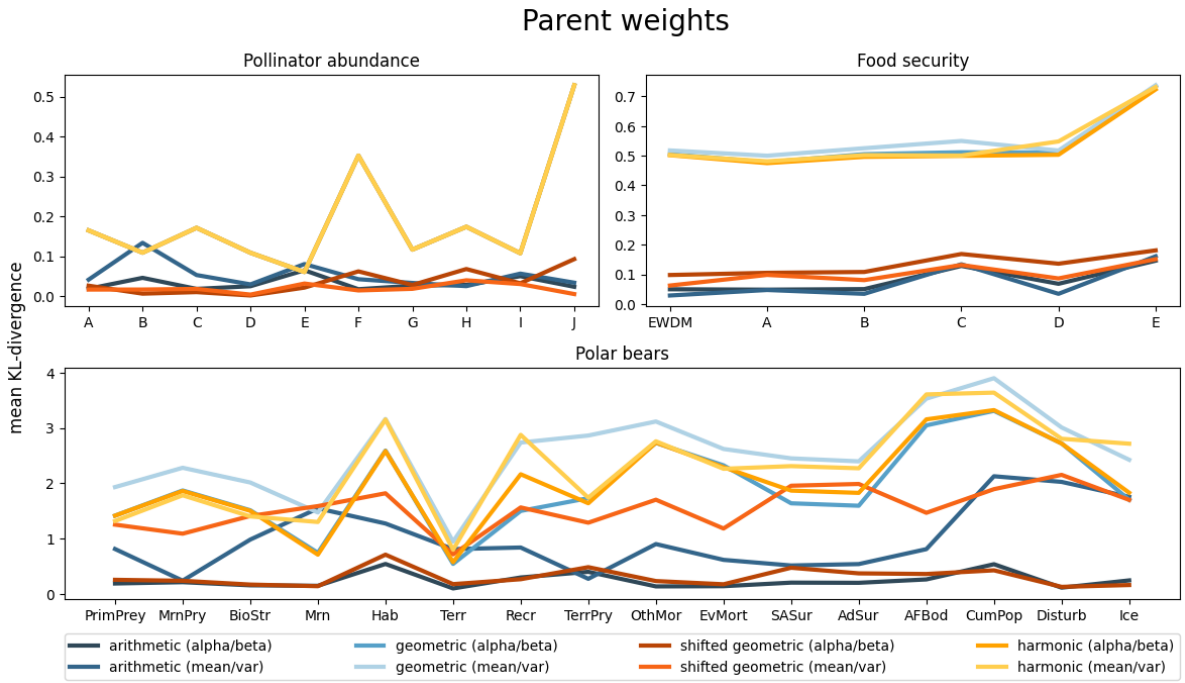


Figure 7.2: Mean KL-divergences of constructed CPTs using different mean functions for the 'parent weights' version of InterBeta.

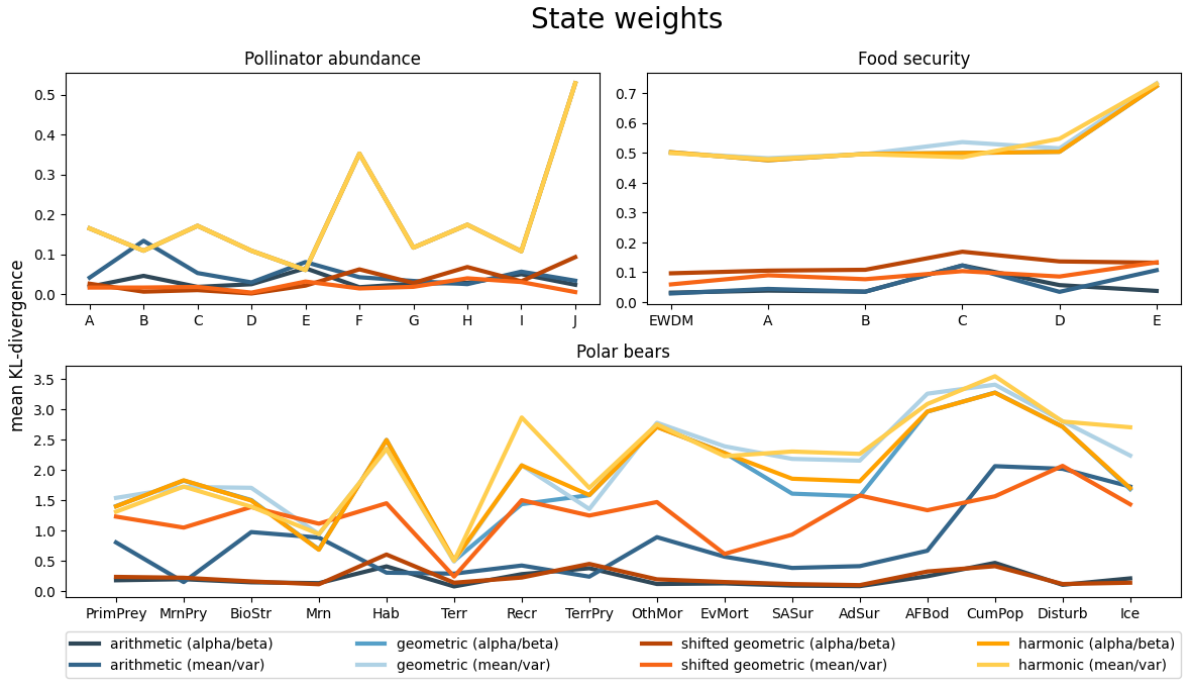


Figure 7.3: Mean KL-divergences of constructed CPTs using different mean functions for the 'parent state weights' version of InterBeta.

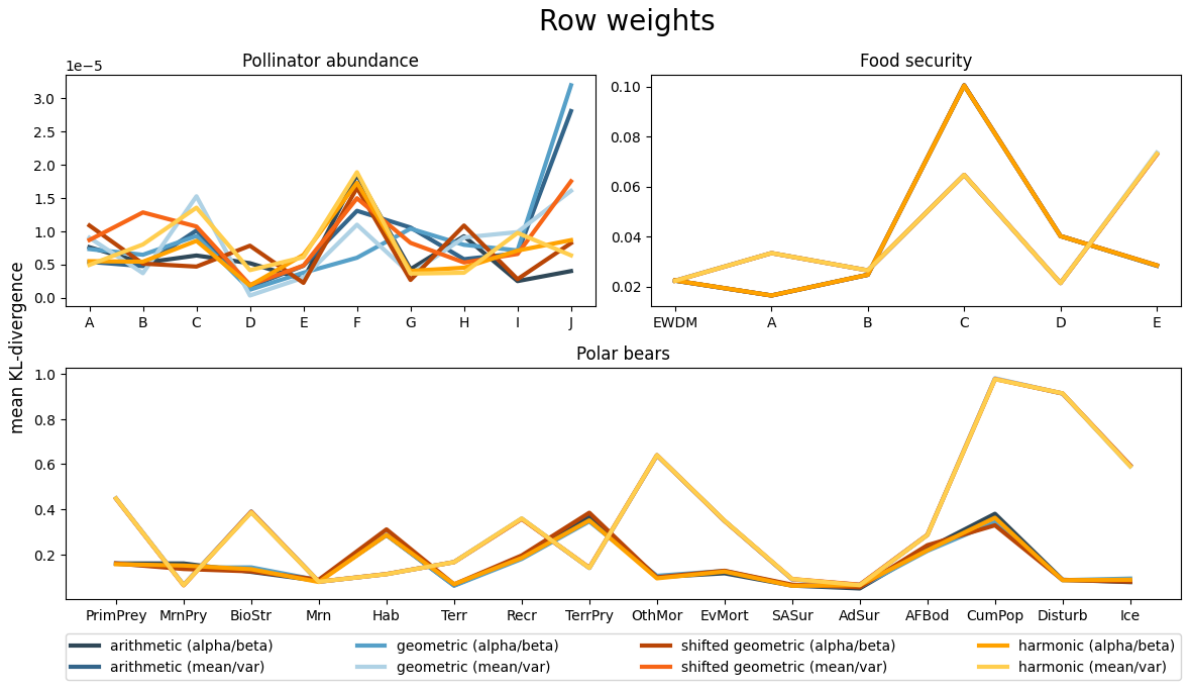


Figure 7.4: Mean KL-divergences of constructed CPTs using different mean functions for the 'row weights' version of InterBeta.

INTERBETA WITH MIDDLE ROWS ELICITED

When comparing the different mean functions and interpolation parameters for InterBeta with elicited middle rows, the differences become less clear. Figure 7.6 contains the performance in terms of the mean KL-divergence. In this case, the performance is only given for the Polar Bears BN, as both the Food Security and Bee Abundance CPTs do not contain middle rows as defined in Section 6.1.2. In Table C.3 also the scores of the best-performing mean function in combination with interpolation parameters are given in terms of the mean KL-divergence and the percentage of agreement.

For the best, worst and mid version in Figure 7.6a, the arithmetic and shifted geometric mean, where the mean/variance are interpolated, perform best for the CPTs having more than 81 values. For the smaller CPTs, there is not one type that performs significantly better than the others. What is remarkable, is that using the arithmetic mean or shifted geometric mean with α/β interpolation performs significantly worse, even though it performed the best for the original InterBeta.

For the parent weights version and parent state weights version in Figures 7.6b and 7.6c, a division has grown between using the different types of mean functions. This is mostly clear for the larger CPTs, where the arithmetic mean and the shifted geometric mean are once again the favorites. As before, there is no clear better choice between what parameters to interpolate, which holds for all CPTs.

However, for the row weights version in Figure 7.6d, it does become clear what parameters are best to be interpolated. For all CPTs, interpolating the α/β performs better than interpolating the mean/variance. As was the case for the original InterBeta with row weights, there is no significant difference between the mean functions, which is because these do not play a role in determining the row weights in this case.

COMPARISON OF INTERBETA WITH OR WITHOUT ELICITED MIDDLE ROWS

After discussing the original InterBeta and InterBeta with elicited middle rows separately, their performance is further compared. For each application of an InterBeta version, the mean function and interpolation parameters are chosen that have the best performance in terms of the mean KL-divergence. For a full comparison of the performances of all Polar Bear CPTs, see Figure C.1 in the Appendices. In this section, three CPTs are focused on: Ice, Recr, and MrnPry; of which the results are shown in Figure 7.5.

For each of the three CPTs, the best and worst (and mid) version is better for the original InterBeta than the extended version with elicited middle rows. For Ice and Recr, the same can be seen for the parent weights, state weights, and row weights versions. However, this is not the case for TerrPry, for which the elicited middle rows do improve the performance of these versions.

In general, when all CPTs are taken into account of the Polar Bears BN it becomes clear that in most cases it is better to use the original InterBeta method and to not increase the complexity with extra elicited rows. Especially when considering the difference in performance between the best and worst (and mid) versions, eliciting middle rows is not beneficial. For two of the CPTs, the elicited middle rows do improve performance when the parent weights or state weights versions are used. Only the row weights version of InterBeta with elicited middle rows outperforms the original InterBeta regularly. This is the case for more than half of the CPTs but at a cost of a significantly increased expert burden.

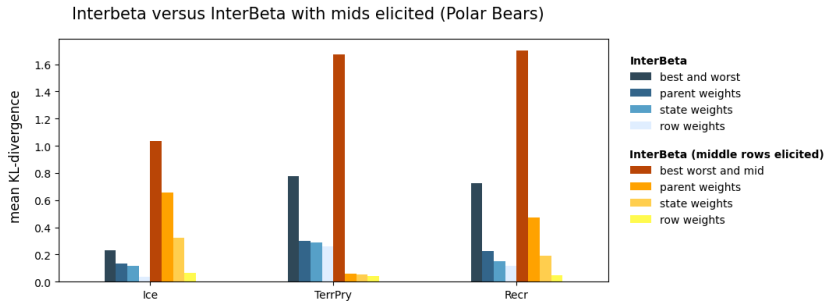
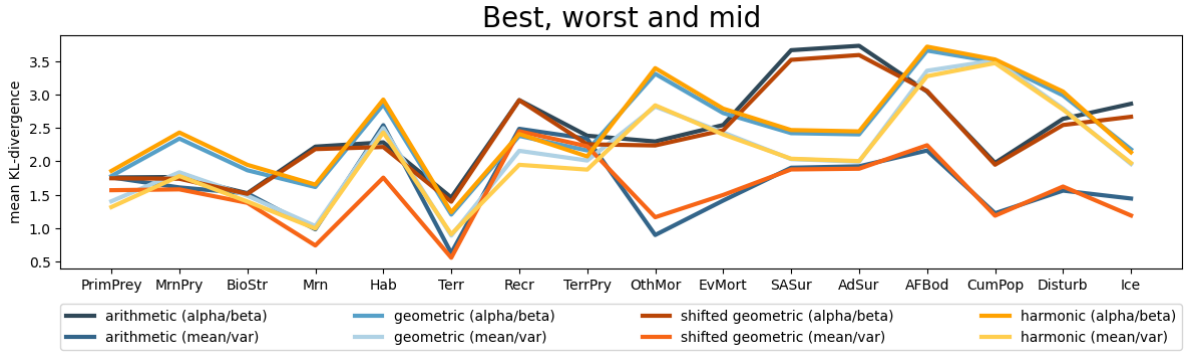
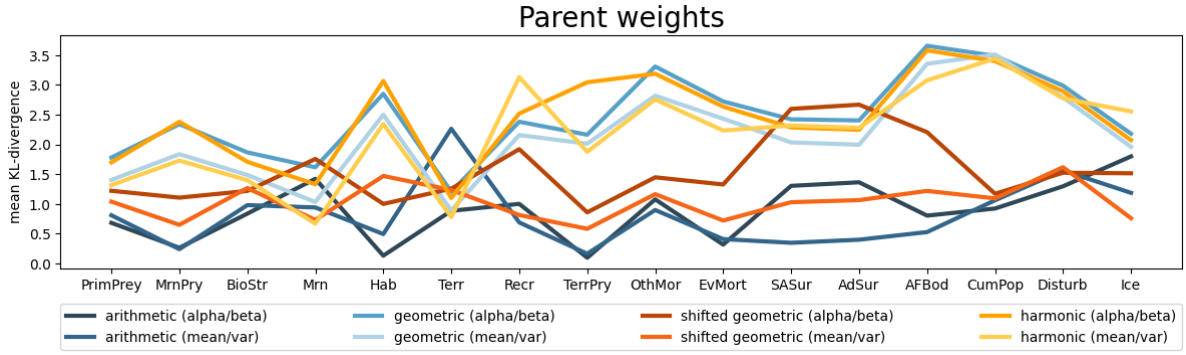


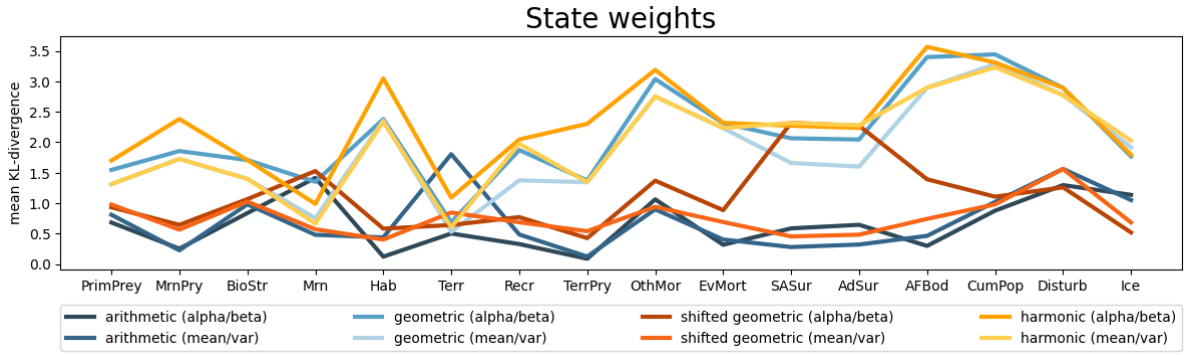
Figure 7.5: Comparison of the KL-divergence of all original InterBeta versions versus all InterBeta versions where the middle rows are elicited as well, for the three CPTs of the Polar Bears BN: Ice, Recr, and TerrPry. With optimally chosen mean function and interpolation parameters.



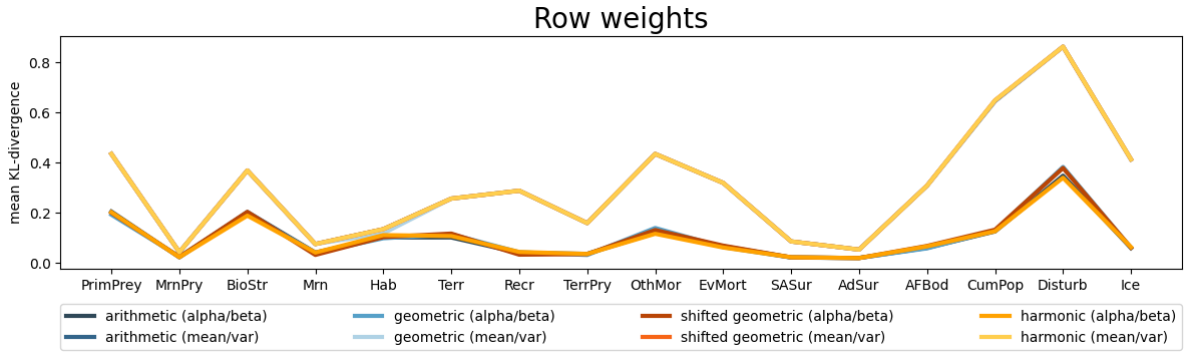
(a) Best, worst and mid.



(b) Parent weights.



(c) Parent state weights.



(d) Row weights.

Figure 7.6: Mean KL-divergences of constructed CPTs using different mean functions for the different versions of InterBeta with middle rows elicited.

EXTRABETA

The final extension to InterBeta that is tested is ExtraBeta. ExtraBeta will be applied using the arithmetic mean and the shifted geometric mean. The arithmetic mean results are the main focus and the shifted geometric mean results can be found in Appendix C. As the geometric and harmonic means were found unsuitable for InterBeta in the previous section, and the foundation of ExtraBeta is the same as that of InterBeta, these means are not considered. Additionally, the interpolation parameters are fixed to the α and β.

For each CPT of Table 5.1 a list of potential combinations of 'good rows' and 'bad rows' is made, that follow the restrictions as stated in Section 6.1.2. Thus, the default weight of a good row is larger than 0.5, and for a bad row, this is less than 0.5. In addition, for each combination, the mean of the good row must be strictly larger than the mean of the bad row. Then, each potential combination of a good and bad row is used as input for ExtraBeta. Some of the main results of ExtraBeta for different input rows are given in this section, starting with the results for the Pollinator Abundance BN.

Pollinator Abundance For the first CPT, there are sixteen potential combinations of good and bad rows that are tested for each expert's assessments. In Figure 7.7 the results of applying ExtraBeta to reconstruct all experts' CPTs are shown in one graph. The performance measured in terms of the mean KL-divergence is presented against the absolute difference between the means of the input good and bad rows. The orange dots represent the results of InterBeta, which is equal to having the best and worst row as input for ExtraBeta. The yellow dots are all results for which either the best row is included in the input or the worst row, the blue dots are the results when the best and worst row are not used.

The results when using the shifted geometric mean instead of the arithmetic mean are shown in Figure C.2 in Appendix C. The figure is very similar to Figure 7.7, thus the following remarks hold for both mean functions.

There seems to be a trend, that as the difference of the means of the good and bad row gets larger, the performance improves. This is mainly clear for the parent weights version, for the best and worst version this trend is also visible to a lesser extent. For all versions, there are many combinations of input rows that don't include the best or worst row which perform as good as, or even better than, InterBeta. When the row weights version is considered, there exists little difference in performance between using different input rows. Especially when the difference in mean is larger than 0.2 there are no significant performance differences.

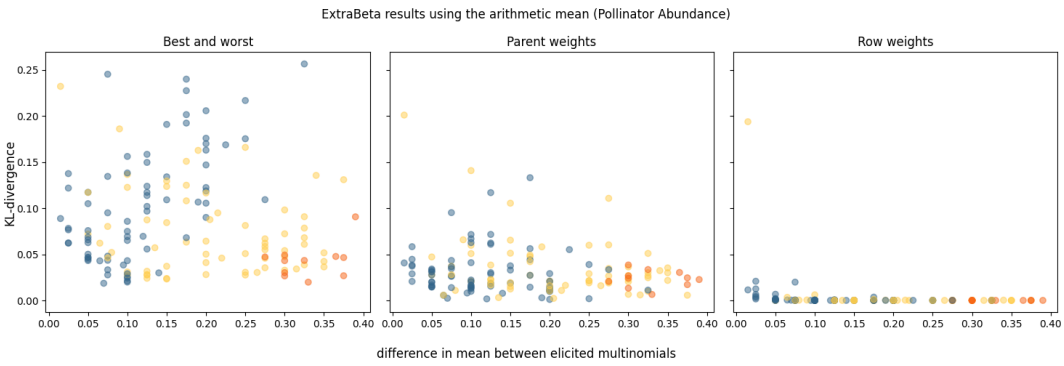


Figure 7.7: Results of reconstructing Pollinator Abundance CPTs using ExtraBeta (arithmetic, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results when either the best row or the worst row is included as input (yellow), and the results when the best and worst row are both not used (blue).

Food Security For each of the Food Security CPTs, there are 25 combinations of good and bad rows. The results of using ExtraBeta are shown in Figure 7.8, using the same format as the Pollinator Abundance results. The performance results of ExtraBeta using the shifted geometric mean are shown in Appendix C, in Figure C.3. In this case there is an even more clear trend visible, as the distance between the means grows, the performance of ExtraBeta improves. For the parent weights, state weights, and row weights versions; an elbow is visible in the graph when the difference between the means of the elicited multinomials is equal to 0.25. The effect of using the best or worst row as input is less large than was seen for the Pollinator Abundance CPTs.

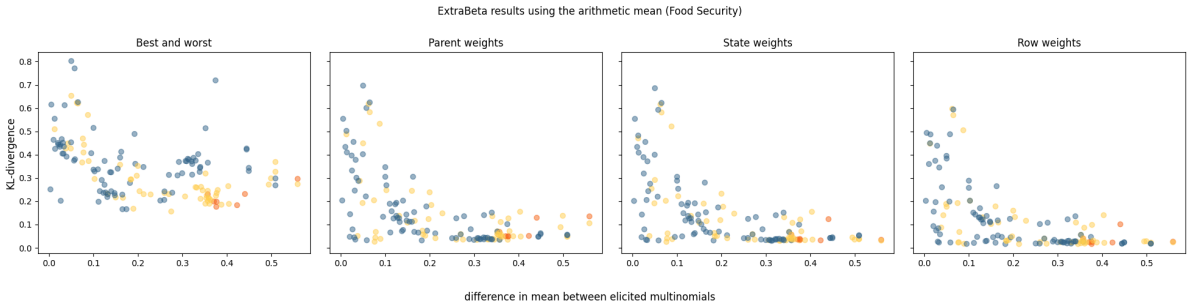


Figure 7.8: Results of reconstructing Food Security CPTs using ExtraBeta (arithmetic, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results when either the best row or the worst row is included as input (yellow), and the results when the best and worst row are both not used (blue).

Therefore, in addition to marking the results based on whether or not the best or worst row was included as input for ExtraBeta, the results are also colored based on what state the dominant parent is in. As stated in Section 5, it was found that the *equivalised income* has the largest influence on the child node[1]. In Figure 7.9, the same results as in Figure 7.8 are shown, but with a different type of coloring. In this case, the dots are colored green for which the *equivalised income* is in its best state for the good row and in its worst state for the bad row. The same type of plot for the shifted geometric mean is shown in Figure C.4 in Appendix C.

There is a visible distinction in performance. If the dominating parent is fixed to the extreme states as input for ExtraBeta, the performance is in general better than using other rows. Especially for the parent, state, and row weights versions of ExtraBeta, the performance is close to InterBeta when fixing the dominant parent. This is mainly due to the fact that fixing the dominant parent to its extremes states makes sure that the input rows are significantly different scenarios, which has the results that the difference between the input row-means is relatively large.

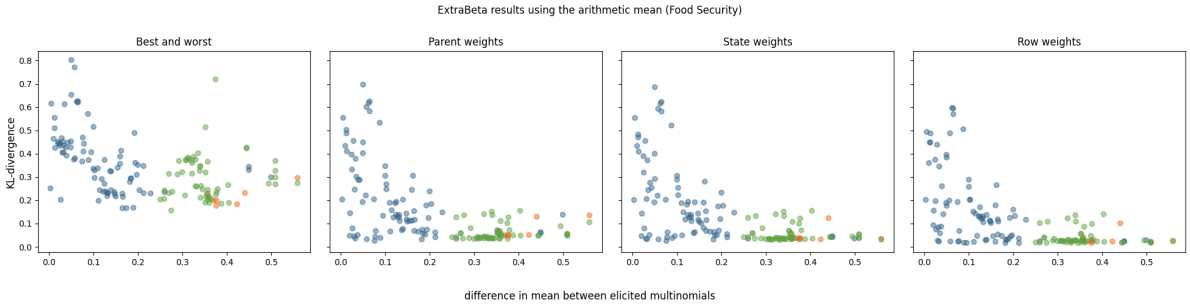


Figure 7.9: Results of reconstructing Security CPTs using ExtraBeta (arithmetic, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results when the good row has *equivalised income* in its best state and for the bad row *equivalised income* is in its worst state (green), and the results for remaining combinations (blue).

Polar Bears The final set of elicited CPTs that ExtraBeta is tested on is part of the Polar Bears BN. Four of the Polar Bear CPTs are not considered as they were too large to test all possible input row combinations within a reasonable time, these CPTs each have more than 144 values. The results of reconstructing all considered CPTs of the Polar Bears BN can be found in Appendix C in Figure C.5 and C.6 for the arithmetic mean, and in Figures C.7 and C.8 for the shifted geometric mean. Three of the CPTs are chosen, each with a dominant parent node. The ExtraBeta reconstruction results of these three CPTs are shown in Figure 7.10.

As was found for the Pollinator Abundance and Food Security CPTs, there is a negative trend for the mean KL-divergence as the mean between the input rows becomes larger. As more weights are added to the method, the mean KL-divergence decreases for all combinations of input rows. Focusing on the parent and state weights results of ExtraBeta on the SASur CPT, the performance of ExtraBeta starts to plateau when the mean difference of the input row becomes larger than 0.5. For the row weights versions, this already hap-

[1] This is supported by the weights that are found for the parent weights version of InterBeta when applied to the Food Security CPTs as shown in Figure 8.1b.

pens for a mean difference of 0.2. For the TerrPry and Recr CPTs, the trend only starts to degrade when the state weights or row weights versions are considered.

When the dominant parent node is set to the best state for the good row and to its worst state for the bad row (as shown by the green dots) a relatively large difference between the input row means is guaranteed. As a result, the ExtraBeta results of setting the dominating parent to its extreme states for the input rows are close to the InterBeta results. For the Recr CPT, the results are extremely close, which is due to the parent being fully dominant. When a fully dominant parent is in its best state, the child node distribution is equal to that of the best row, and when it is in its worst state, the child node distribution is equal to that of the worst row. As weights are added to the ExtraBeta and InterBeta methods, the difference in performance decreases considerably.

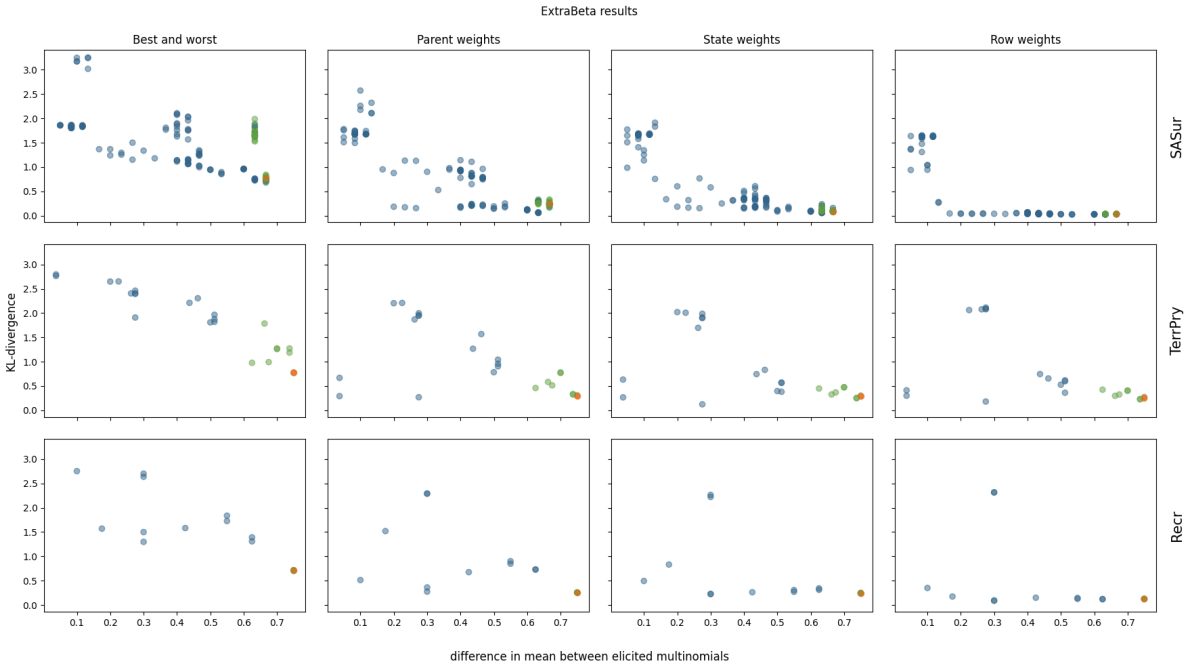


Figure 7.10: Results of reconstructing three Polar Bears CPTs using ExtraBeta (arithmetic, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results of the dominant parent node fixed to its best and worst state for the good and bad row respectively (green), and the results for the remaining combinations (blue).

In general, for part of the good and bad input rows of ExtraBeta, the results are very similar to the results of InterBeta. In some cases, the performance of ExtraBeta is even better than InterBeta, but at the cost of an increased elicitation burden. The elicitation burden is increased due to experts having to choose the to-be-elicited rows themselves. To increase the likelihood of experts selecting rows as input that perform similar to, or better than, InterBeta, the difference between the means of the input rows should be chosen as large as possible. In an elicitation, experts should therefore be guided in the row-selection process. For instance, when a dominating parent is present, it was found to be good practice to set this to the extreme states. This particular method will be revisited in Chapter 8, where the idea is tested on simulated CPTs.

7.1.1. ROW-BY-ROW BETA PARAMETERS

As a final analysis of InterBeta applied to the CPTs from Table 5.1, the child node distribution at an individual row level is considered. For each of the CPTs beta distributions are fit to the separate CPT rows using the same method as was used to fit the best and worst row of the InterBeta method. The found parameters can then be plotted and compared, to see what are the effects of interpolating parameters.

To study the calculated parameters by InterBeta, which depend on what parameters are interpolated between, the interpolation curves between the best and worst rows are plotted for each CPT. This is done for both the α, β and the mean and variance in side-by-side graphs. Note that these parameters relate to each other by Equation (4.9), so each point in the α/β graph pairs with a point in the mean/variance graph. Finally, the fitted beta distribution parameters for each separate row can be compared with the interpolation curves.

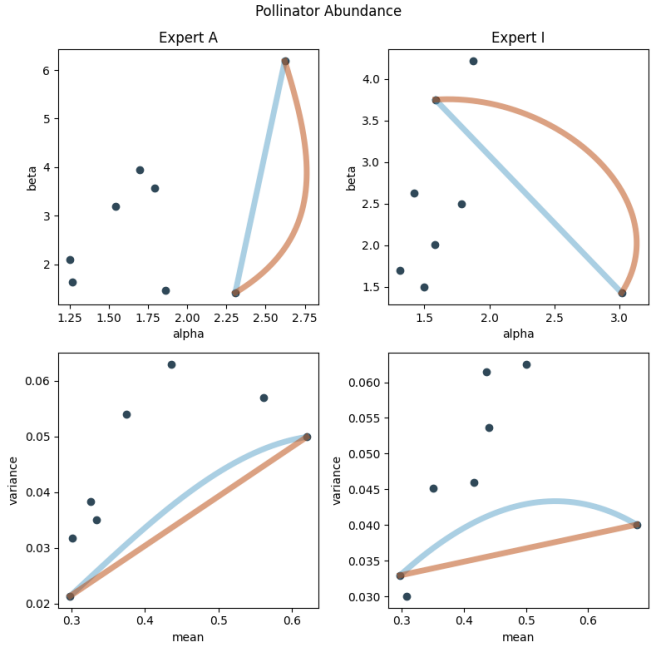
The three expert-elicited BNs are each treated separately, starting with the Pollinator Abundance BN.

POLLINATOR ABUNDANCE

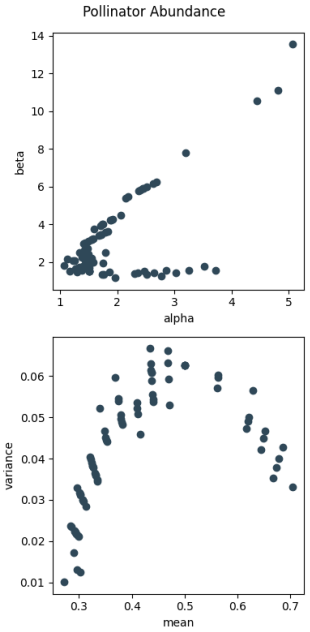
For experts A and I, fitted beta distributions to their elicitations are shown in Figure 7.11a. For a full overview of all experts see Figure C.9a and C.9b in the Appendices. In the figures, each dot represents the parameters of a fitted child node distribution to one CPT row, the lines show the interpolation line between the best and worst row, for the α, β in blue and for the mean, variance in red. So, all of the parameters of the constructed CPT rows by InterBeta lie on these lines, the exact positioning depends on the weight parameters.

If the α, β are interpolated, the corresponding blue curve for the mean and variance looks concave, with a higher variance for intermediate rows. For the elicited CPTs, the variance obtained by interpolating α/β is always higher than when the mean and variance are interpolated. This idea was further tested in Appendix B.2, where it was found that the variance is not always concave. However, in almost all cases, the variance curve obtained by interpolating α/β lays above the linearly interpolated variance. For all experts, there is at most one fitted mean/variance parameter point that lies under the interpolation line between the best and worst row, and thus most lie above. This could suggest that a concave relationship between the mean and variance is most fitting. The best-case scenario would be for one of the interpolation curves to go through all scattered CPT row parameter points. Although this was not always found in practice, interpolating the α, β does come a little closer to reaching the true fitted parameters.

Figure 7.11b shows all of the fitted beta distributions to all CPT rows of all experts in one graph. There is a clear trend visible for the mean and variance, where the variance looks to have a quadratic relationship with respect to the mean, or, at least there seems to be a higher variance for the intermediate rows than for the best and worst row. For the α, β parameters, some sort of rotated quadratic relationship can be seen.



(a) Fitted beta distribution parameters for each CPT row for Experts A and I, including the interpolation line for α, β (blue) and for the mean and variance (red).



(b) Fitted beta distribution parameters of all elicitations in one figure.

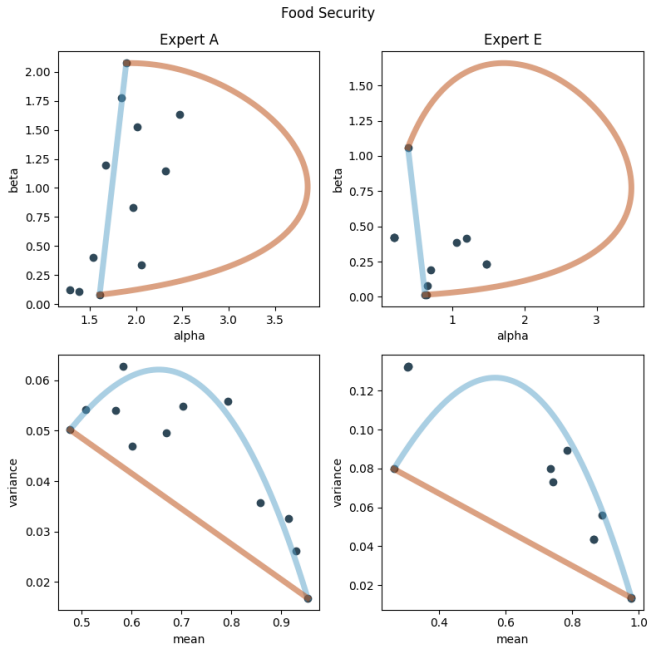
Figure 7.11: Overview of fitted beta distribution parameters to the elicited Pollinator Abundance CPT rows.

FOOD SECURITY

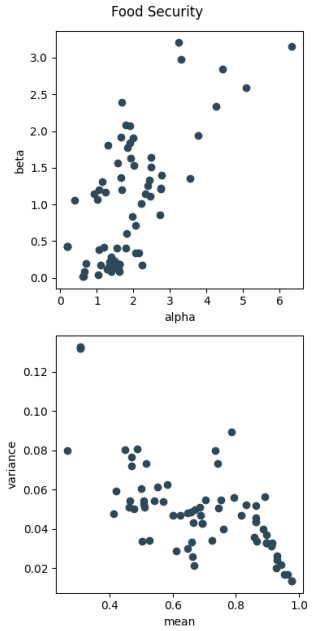
Similar to the pollinator abundance overview, two elicited CPTs are selected for further investigation. Figure 7.12a contains the fitted beta distribution parameters for experts A and E, and the full overview can be seen in the Appendices in Figures C.10a and C.10b. Once again, the curves show the interpolation lines that are used by InterBeta to determine the beta distribution parameters. For both experts, the blue line, representing the linear interpolation of the α, β, fits the fitted parameter points better than the red line. The mean and

variance do not seem to have a linear relationship but rather a relationship approximating a concave one, as was found for the Pollinator Abundance case. Most of the fitted mean/variance parameter points are above the linear interpolation line between the best and worst row.

When all of the fitted beta distributions of all experts are considered at once, like in Figure 7.12b, there are no immediately visible relationships present. Most of the experts agree on a mean close to one and a low variance for the worst row, which can be seen in the graph from a large density of points close to a mean of one and a variance of 0.02. For the best row, the experts' means lie closer to 0.5 and are more spread out. This might be part of the reason there is no clear relationship, as the mean does not cover the complete range of $(0,1)$.



(a) Fitted beta distribution parameters for each CPT row for Experts A and E, including the interpolation line for α, β (blue) and for the mean and variance (red).



(b) Fitted beta distribution parameters of all elicitations in one figure.

Figure 7.12: Overview of fitted beta distribution parameters to the elicited Food Security CPT rows.

POLAR BEARS

For the Polar Bears BN, there is more variability, as the included CPTs are each modeling different concepts. For a selection of CPTs the fitted beta distribution parameters are scattered in Figure 7.13, for the full overview, see Figures C.11a and C.11b. Not only the interpolation lines between the best and worst row parameters are shown, but also the line for the middle rows is included. In this case, the interpolation line of the mean and variance resulted in a curve for the α, β that contains values $\alpha >> 100, \beta >> 100$, which would have made the graphs unreadable and it is thus left out.

A quick look would suggest that the piecewise interpolation curves of the InterBeta version with middle rows elicited fit the points better than the original InterBeta which only relies on the best and worst row. However, it was previously found that, in general, InterBeta with elicited middle rows does not perform better than the original InterBeta. One potential cause is the lowered variance for the intermediate rows. The elicited middle rows of the Polar Bears BN have a variance close to zero, which also forces the variance of the other intermediate rows to be smaller than when this middle row is not used. So although more CPT rows are approximated more closely, also more CPT rows are approximated worse.

For most of the specified CPTs included in the Polar Bears network, the best and worst rows have a variance close to zero and a mean close to one or zero, respectively. This leads to the interpolation curves being very similar for each CPT, each having an approximate maximum variance of around 0.03. All intermediate rows have a variance that is larger than the best/worst row.

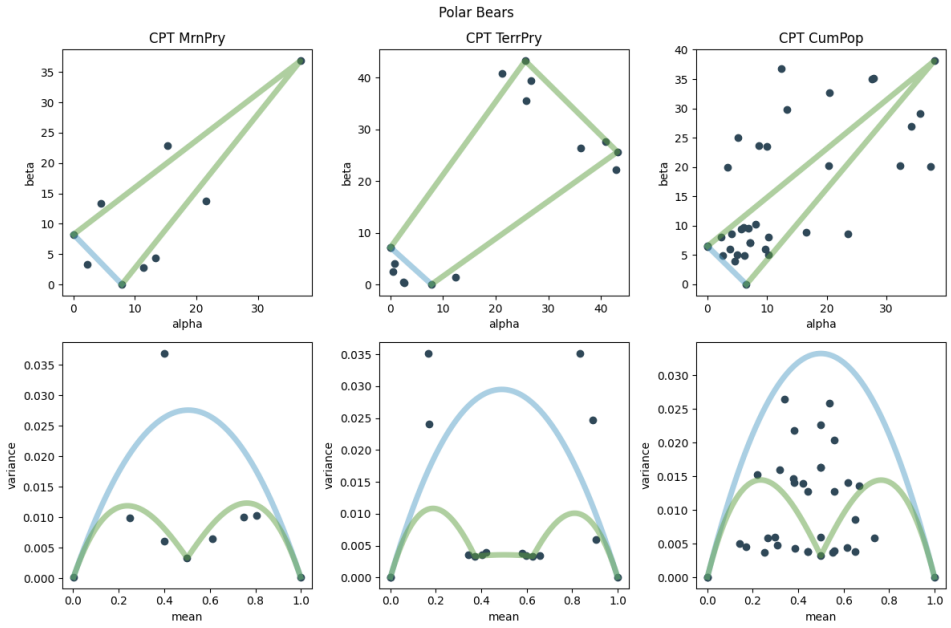


Figure 7.13: Fitted beta distribution parameters for each CPT row for CPTs MrnPry, TerrPry, and CumPop, including the interpolation line for α, β between the best and worst row (blue) and between the best, worst and middle rows (green).

7

7.2. RNM

Following InterBeta and all its versions and extensions, RNM and AutoRNM are applied to reconstruct all CPTs included in Table 5.1. Although not all of the CPTs conform to the criteria of RNM, the methods are implemented in such a way that this does not hinder the CPT construction. The Household Food Security CPT and nine out of sixteen of the Polar Bear network CPTs do not have an equal number of states for all parent nodes and the child node. This means that Proposition 4.4.1 cannot be attained for 16 out of 34 CPTs. The remaining CPTs do have an equal number of states for all parents and the child node, in addition, the elicited CPT values conform to Proposition 4.4.1.

In Figure 7.14 the performance results are shown for both RNM versions, where the performance is measured in terms of the mean KL-divergence. Table C.1, in Appendix C, also contains these results in addition to the percentage of agreement between the constructed CPT and the fully elicited CPT. For all of the included results of AutoRNM, the set of elicited rows that were used as input was $S_a^+ \cup S_b^+$, as defined by Equations (6.5) and (6.6).

The Figure shows that both versions have similar performance. For the Pollinator Abundance CPTs, AutoRNM performs slightly better than RNM for all experts apart from expert F. This can be explained by the fact that the elicited rows cover all of the CPT, so the full CPT was used as input. Thus, the weights could be optimized based on the full CPT, without the restriction of having rounded weights, which is a perk over the original RNM. The reason why AutoRNM does not always perform better when constructing Pollinator Abundance CPTs is likely the imperfect weight optimization. The method uses a greedy search algorithm with 1000 iterations, in comparison to a grid search which is used for finding optimized weights for RNM.

Also for the other BNs, AutoRNM can outperform RNM, even though the weights are optimized using less data. The cause for AutoRNM's better performance for certain CPTs is that the weight and variance parameters are not rounded, whereas for the original version, the weight parameters are rounded to the nearest half and the variance is rounded to four decimals since it cannot be expected from experts to be more precise than that. This allows the two versions of RNM to be compared in a more realistic manner. This highlights the benefits of AutoRNM, as it gives an alternative to eliciting weights and is able to choose more precise weights.

In general, for most of the CPTs, there exists little difference between the performance of RNM and AutoRNM.The average absolute difference in KL-divergence is 0.028 for the Polar Bears BN, 0.019 for the Food Security BN, and 0.001 for the Pollinator Abundance BN. The maximal absolute difference is found to be 0.068 for the Polar Bear Ice CPT. Comparing the difference in the percentage of agreement is similar, with an average difference between the original and extended version of 4.4% for the Polar Bears BN, 13.1% for the Food

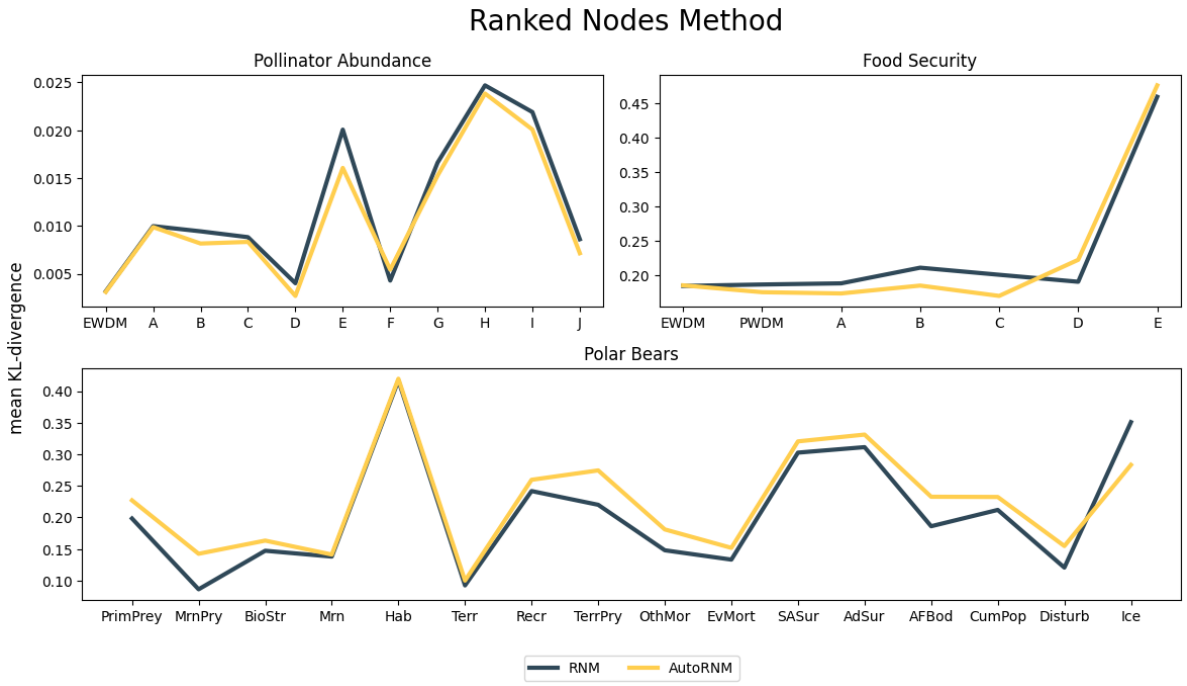


Figure 7.14: Mean KL-divergences of constructed CPTs using the original Ranked Nodes Method versus the extended version.

Security BN, and 3.4% for the Pollinator Abundance BN. A maximal difference of 25% is found for the Polar Bear Mrn CPT and experts C and E of the Food Security CPT.

The Polar Bear CPTs are ordered by the number of values in the CPT, with PrimPrey being the smallest and Ice the largest. There is no clear trend visible of how the performance changes when CPTs become larger.

In general there is not one version that performs better for every CPT. This mainly shows that optimizing the weight and variance parameters based on only the extreme scenarios, where weights are not rounded to halves, is not always inferior to optimizing based on the full CPT. Thus, the main difference is in the number and type of parameters that are needed as input.

In most cases, the original RNM requires fewer parameters to be elicited than AutoRNM. Although, for RNM trial and error is used to find appropriate weights, even when multiple trials are required, RNM still needs fewer parameters than AutoRNM. Let τ be the number of trials needed to find weights for RNM, then AutoRNM requires less parameters if:

$$2+2ns_c \leq 2n+2+\tau(n+1),$$
$$s_c \leq \frac{2n+\tau(n+1)}{2n},$$

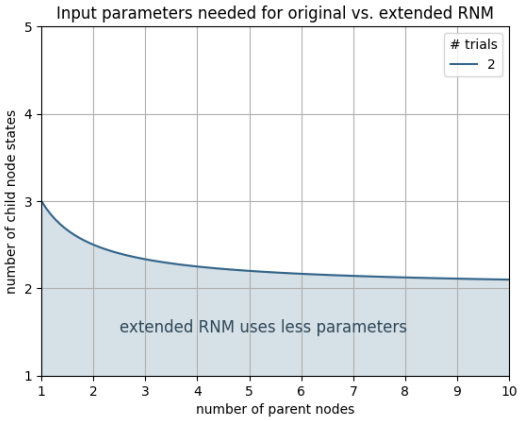


Figure 7.15: Number of child node states and parent nodes in a BN such that the original RNM and extended RNM require an equal amount of input parameters, in case the original RNM requires 2 trials to find suitable weights.

where n is the number of parent nodes. The graph of this inequality is shown in Figure 7.15 for $\tau = 2$. In Appendix C a more detailed version with a graph for multiple values of τ can be found in Figure C.13. Assuming that the elicitation burden is equal for the different types of parameters, and the child node has two states, it is less burdensome for the experts to use AutoRNM. However, when the child node has more than two states, the original version requires fewer parameters to be elicited, assuming the parameters are found in two trials.

In Section 7.4 both RNM and AutoRNM will once again be compared, together with InterBeta and Functional Interpolation, but then also in relation to the burden.

7.3. FUNCTIONAL INTERPOLATION

The results of using the Functional Interpolation method to construct CPTs are shown in Figure 7.16, and the exact values are also shown in Table C.1 in Appendix C. First, note that the y-axis of the Pollinator Abundance graph is in the order of $1e-7$, which makes the difference between the different distributions very small. For those CPTs the performance is very good, as the KL-divergence is close to zero. This is because, for the Pollinator Abundance CPTs, the entire CPT was used as input. Thus the only cause for the KL-divergence not to be equal to zero is the distributions not exactly fitting to the elicited multinomials. The main result from the application of the Functional Interpolation method to the Pollinator Abundance CPTs is that the truncated normal distributions cannot be fit to the CPT rows as accurately as the other distributions.

Moving on to the Food Security BN, once again the truncated normal does not perform the best. For these CPTs, the beta distribution seems to be the best choice. In this case, a partially elicited CPT is used as input, but a significant part is used, only four out of twelve elicited CPT rows are not used as input. There is also little difference to be found between interpolating the α, β or mean and variance of the beta distribution. This may also be explained by the low number of CPT rows that need to be calculated based on interpolation.

The real differences are starting to show for the CPTs in the Polar Bears BN. For each of the included CPTs at most 44% of the elicited CPT is used as input, which means that the CPT construction relies more on interpolated parameters than when applied to the Pollinator Abundance or Food Security CPTs. For most CPTs, the truncated normal distribution is once again the least good option, and the beta distribution where the mean and variance are interpolated is often the best option. Only for the Recr CPT, the results of the beta distribution with mean/variance interpolation are drastically worse.

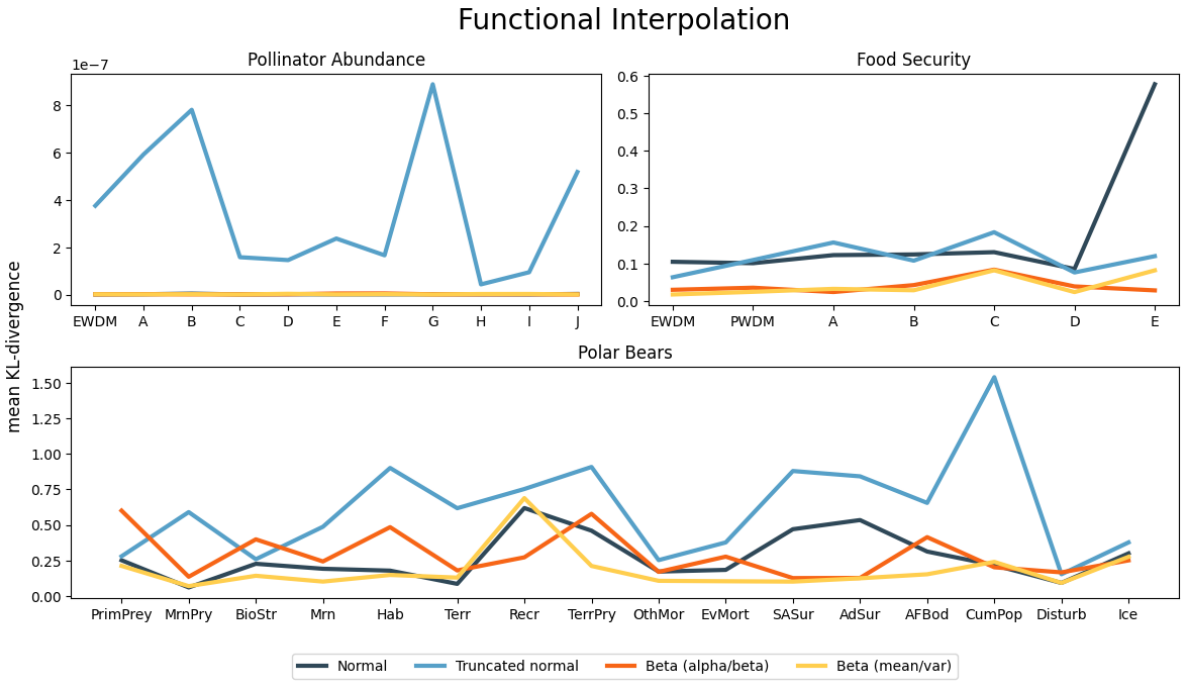


Figure 7.16: Mean KL-divergences of constructed CPTs using the Functional Interpolation method with different probability distributions.

Using the Functional Interpolation method with the beta distribution is very similar to InterBeta, the only difference being that there are no weights that are specified and more than just the mean and worst row is elicited. It would be possible to also elicit weights for the Functional Interpolation method, but since the elicitation load is already significantly higher than for InterBeta, this would only be good practice for large CPTs with nodes that contain many states. In that case, both parent weights and parent state weights would be options to consider. Row weights are not applicable, as the interpolation is in a hypercube and not on a curve.

The results of using Functional Interpolation to reconstruct CPTs have shown that the beta distribution is a better fitting distribution than the normal and truncated normal distributions. To substantiate this finding, the three distributions are fit to all CPT rows individually as well. The multinomials that are derived from the discretized distributions are then once again compared to the original CPT rows. The results can be seen in Figure 7.17. For each of the BNs, the beta distribution has the lowest KL-divergence, which is significantly lower than the fitting errors found for the normal and truncated normal distribution. The difference between the normal and the truncated normal distribution depends on the BN. For the Food Security BN, the normal distribution fits the worst by far. For the other BNs, the truncated normal distribution has the highest error. These results highlight that the beta distribution is the best-fitting distribution when used to represent multinomials.

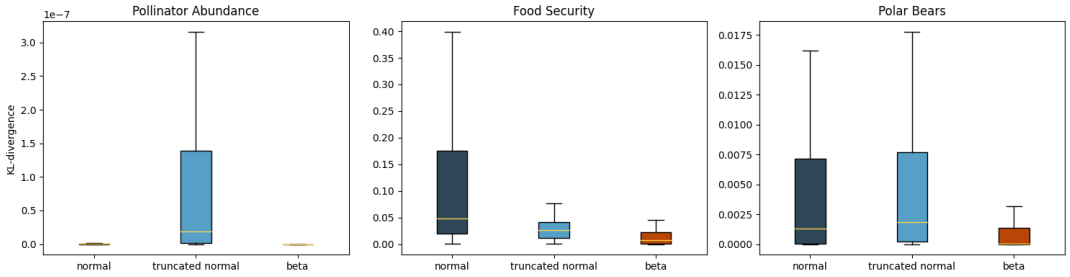


Figure 7.17: Fitting error of the normal, truncated normal, and beta distribution over the CPT rows of all CPTs in Table 5.1, in terms of the KL-divergence.

7.4. COMPARISON

In this section, the performance (as measured by the KL-divergence) results of InterBeta, RNM, AutoRNM, and Functional Interpolation will be compared, both in terms of accuracy and elicitation burden. The results of ExtraBeta are left out of this comparison, as there are too many combinations of input parameters that were tested. The ExtraBeta results are comparable to the InterBeta results for certain input rows, but with an elicitation burden that is increased with two parameters, as shown in Section 7.1.

The results for InterBeta and InterBeta with middle rows elicited are shown for all four versions: best and worst, parent weights, parent state weights, and row weights, where the best-performing mean function and interpolation parameters are used. Both RNM and AutoRNM are included in the comparison, and for Functional Interpolation, only the best-performing distribution is taken. A full overview of the results for all CPTs can be seen in Figure C.12, alongside the Tables C.1, to C.3 in Appendix C. Only a selection of CPTs is chosen to be discussed in more detail.

Figure 7.18 shows the results for four of the fully elicited CPTs, two CPTs are part of the Polar Bears network: Ice and Recr. The other two CPTs are the PWDM decision maker's results for Food Security and the EWDM CPT for the Pollinator Abundance BN. Starting with the Ice CPT on the top left, which is the largest CPT that is included in this study. Overall, InterBeta with row weights has the lowest mean KL-divergence, but also requires 78 parameters to be elicited. The original InterBeta using only the best and worst row already outperforms RNM, AutoRNM and Functional Interpolation whilst requiring fewer parameters as input. Also eliciting parent weights decreases the KL-divergence by almost 30%, but this decrease flattens when the parent state weights are elicited, with only roughly an 8% decrease compared to the parent weights. Therefore, InterBeta with parent weights is a good choice, that can reduce the number of parameters that need to be elicited with 96%.

Similar results are obtained for the Recr CPT, where the original InterBeta has relatively good performance whilst requiring the least amount of parameters. InterBeta with mids elicited and row weights has the overall best performance but needs almost half of the CPT size as input. The mean KL-divergence of RNM is in between the parent weights and parent state weights version of InterBeta.

For the EWDM of the Food Security CPT, InterBeta outperforms RNM with all versions. Then using parent weights improves the performance significantly. There is little added value to using parent state weights or even row weights. Similar results are found for the experts A-E for Food Security, considering both performance and required input, the parent weights or parent state weights options are the best.

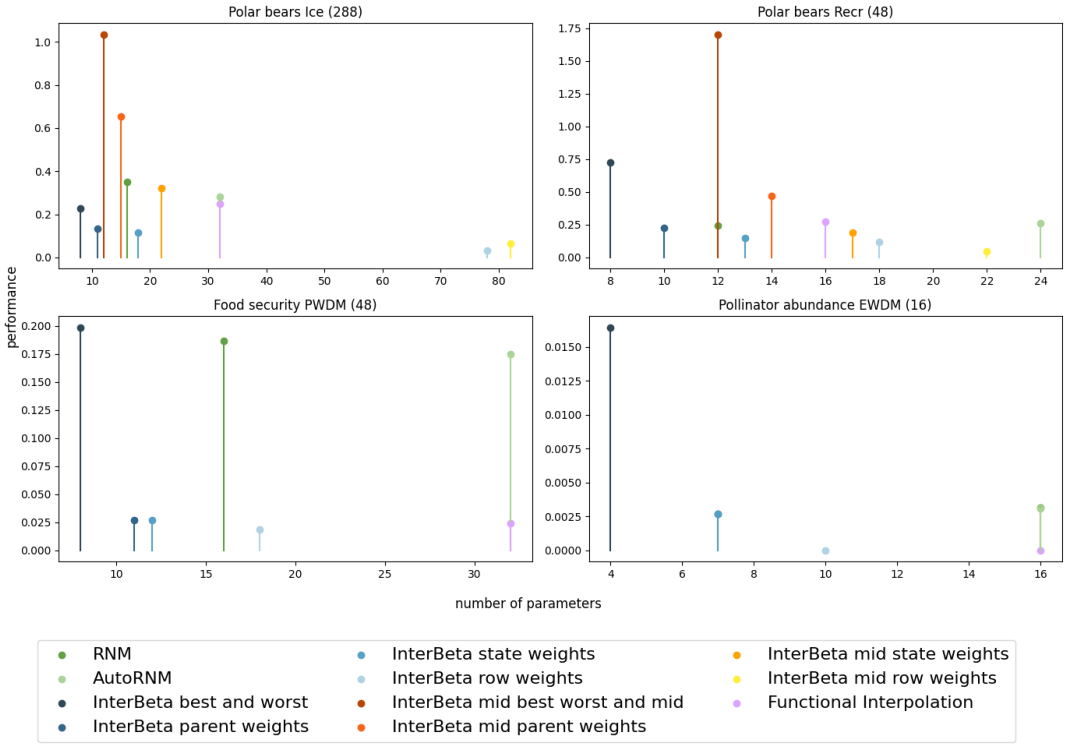


Figure 7.18: Comparison of the KL-divergence between four elicited CPTs and constructed CPTs for the following methods: InterBeta, RNM, and Functional Interpolation.

Finally, for the Pollinator Abundance CPTs, it is found that both InterBeta with row weights and Functional Interpolation can reconstruct the CPT with a KL divergence smaller than 10^{-4} and a percentage of agreement score of 100%. This comes as no surprise for the Functional Interpolation method as it requires the full CPT as input. InterBeta with row weights needs almost 2/3 of the CPT size as input. Note that the scale of the mean KL-divergence is in the order of 10^{-2} which means that all methods perform relatively well. Even InterBeta using only the best and worst row, which requires only four parameters to be elicited, has a mean KL-divergence of less than 0.055 for all experts' assessments A-J.

Finally, comparing the overall performances of the CPT construction methods, InterBeta is in general the best choice. The method offers the most flexibility with the different versions. In general, the parent weights version provides a significant improvement over the best and worst version, with only a relatively small burden increase. The state weights or row weights versions do provide even better performances, but this comes at a relatively high cost.

The next best-performing method seems to be Functional Interpolation. In many cases, it performs similarly to the row weights version of InterBeta. For BNs where the parent nodes have more than three states, the Functional Interpolation method requires fewer parameters to be elicited in all cases but one. Especially if experts are more comfortable in assessing probabilities or relative frequencies than assigning weights to rows, Functional Interpolation should be chosen over InterBeta with row weights.

7.5. CONCLUSION AND DISCUSSION

For the InterBeta method, it was found that using the arithmetic mean or the shifted geometric mean yielded the best results. The performance difference between interpolating the α/β or the mean/variance is small, but in general in favor of the α/β. This is later also supported by the fact that linearly interpolating the α/β almost always results in a non-linear variance curve that lies above the linearly interpolated variance, as shown in Appendix B.2. This was in line with the variances found for the individual rows of the fully elicited CPTs.

The variation of InterBeta where additional middle rows are elicited, did not show to be promising for all CPTs. The best and worst version of the original InterBeta outperformed InterBeta with elicited middle rows,

for all Polar Bears CPTs. In two out of the sixteen Polar Bear CPTs, the parent and state weights versions of InterBeta with elicited middle rows performed better than the original InterBeta method. Due to the significant increase in elicitation burden and the mediocre CPT reconstruction performance of eliciting middle rows, this is not recommended. In future work, it could be investigated whether eliciting other types of middle rows helps improve the method, for example, the rows that the Functional Interpolation method uses.

The ExtraBeta variation did have promising results. For several combinations of input CPT rows, that were not equal to the best and worst row, the original CPT can be reconstructed at least as good as by InterBeta. It was found that the larger the distance between the means of the input CPT rows, the smaller the mean KL-divergence between the constructed and true CPT. If there is a dominant parent node, that has a significantly larger influence on the child node distribution than the other parent nodes, fixing this node to the best state for one input row and to its worst state for the other forces a large difference between the CPT row means. This results in a lower KL-divergence.

The next method that was included in the comparison is the Ranked Nodes Method (RNM) and the extended version, AutoRNM. The main conclusion that can be drawn is that in most cases there is no significant difference between both versions of RNM. Thus the main difference is the elicitation burden. If experts are more comfortable giving weights to parent nodes and finding a CPT using trial and error, the original RNM version is the better choice. Otherwise, if experts are more comfortable in assessing probabilities or relative frequencies, the extended version is recommended.

The final model included in the comparison is the Functional Interpolation method. It was found that it is best practice to use the beta distribution, only for a few cases the normal distribution was a better fit. Overall, Functional Interpolation was among the best-performing methods for constructing CPTs. The main downside is the relatively large elicitation burden it places on the experts.

From the comparison of the CPT construction methods it was found that InterBeta with parent weights is the best option for most CPTs. Other methods, state and row weights versions of InterBeta and Functional Interpolation, generally outperform the parent weights version, but with a significantly increased elicitation burden. Only the best and worst version of InterBeta requires fewer parameters but has a significantly worse CPT construction accuracy. Therefore, in the next chapter, InterBeta will be further investigated when applied to reconstruct simulated CPTs.

8

INTERBETA APPLIED TO SIMULATED CPTS

Following the application of CPT construction methods to fully elicited CPTs that were part of past studies, simulated CPTs will be used for further investigations involving InterBeta (the best performing method) and the proposed extension ExtraBeta. The simulated CPTs will be based on correlation structures found in the previously elicited CPTs. This chapter starts with a short analysis of the correlation structures that can be found in the elicited CPTs. Based on the common correlation structures, a selection of structures is chosen on which the simulations are based. Following this analysis, an overview is given of the method for simulating CPTs.

Then, the setup and results of the performed simulation studies are described. Spread over individual sections the following research questions are answered:

1. How do the arithmetic mean and shifted geometric mean compare in effectiveness for different combinations of parent node state counts and correlation structures?

2. What is the influence of a growing number of parent states and child states on the performance of InterBeta, and which InterBeta versions are optimal for different combinations of parent and child state counts?

3. Does varying the discretization intervals of the child node influence the performance of InterBeta?

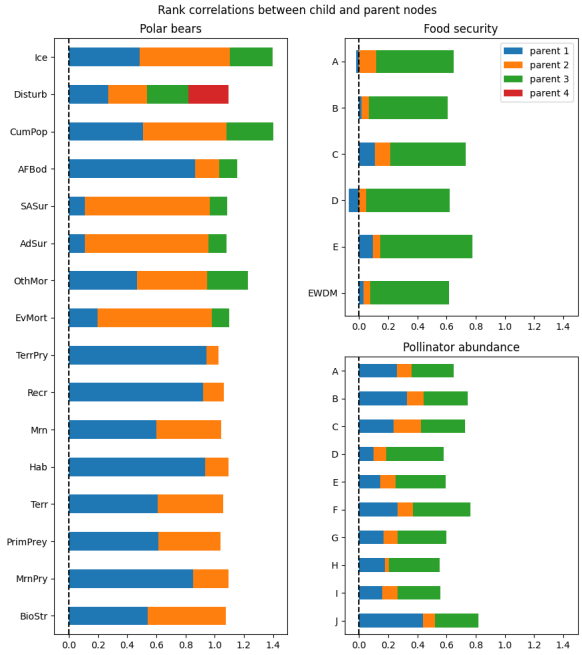
4. What guidelines can be formulated for the ExtraBeta method, to help experts choose appropriate input rows?

For each of the questions above, the performance will be measured both in terms of the mean KL-divergence and the percentage of agreement between the CPTs constructed with InterBeta (or ExtraBeta) and the simulated 'true' CPTs, using the formulae in Section 4.8. To assess the elicitation burden, a short overview is given on how the CPT size grows as the number of child node states and parent node states increases, and on how this growth affects the number of required InterBeta parameters.

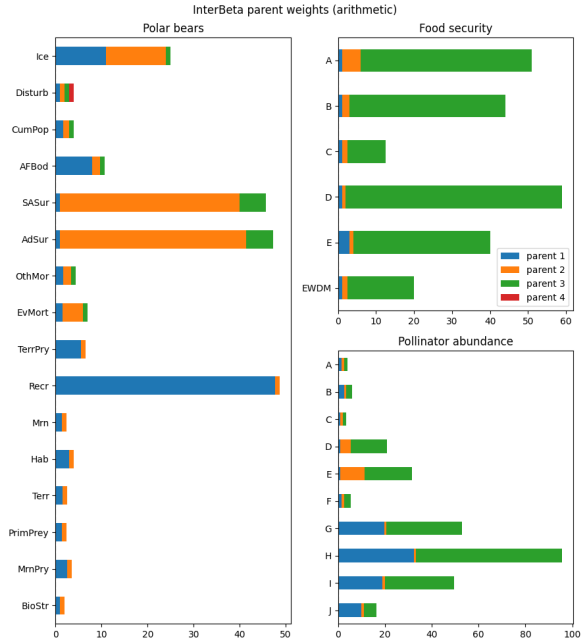
8.1. CORRELATION STRUCTURES

The correlation structures corresponding to the elicited CPTs are used to simulate CPTs. To calculate the correlation structure of a CPT, first a set of 10,000 points is sampled from the sub-BN that consists of the child node (described by the CPT that we want to determine the correlation structure of) and its parent nodes. As the distributions of the parent nodes are not generally known, these are represented by uniform marginal distributions. To sample a point from the sub-BN, first the parent node states are sampled from uniform distributions. The sampled parent node states determine the row (multinomial) of the CPT that is used to sample the child node state from. The Spearman rank correlation matrix is then calculated using the sample of 10,000 points.

The maximum absolute value of the Spearman rank correlation between any of the parent nodes was found to be 0.018, which is close to zero. This was to be expected as the parents are represented as independent in the BNs.



(a) Spearman rank correlations.



(b) Optimized parent weights.

Figure 8.1: Spearman rank correlations between the child node and all of the parent nodes, and optimized parent weights for the InterBeta method with α, β interpolation using the arithmetic mean, for all elicited CPTs. In the plots, blue is for parent 1, orange for parent 2, green for parent 3, and red for parent 4.

The Spearman rank correlations between the child node and all of the parent nodes are shown in Figure 8.1a. The correlations are displayed using a stacked bar plot, where the total length of each bar represents the sum of the correlations, which may be larger than one. Alongside the correlations, the found optimized parent weights of the InterBeta method are shown in Figure 8.1b. For some of the CPTs - such as for Disturb, Mrn, Terr, PrimPrey, and Biostr - the found correlations are similar between the child and each parent. In that case, also the found parent weights are of similar magnitude. On the other hand, for certain CPTs, such as SASur, AdSur and all of the Food Security CPTs, the correlations are significantly different for all parents. This is reflected by the found parent weights as well, where the weights may differ by an order of magnitude.

In addition, these findings can also be compared to the InterBeta results as shown in Figures C.12 and C.1 in Appendix C. The CPTs for which the largest difference occurs between the performance of the best and worst version and the parent weights version, also turn out to be the CPTs with the largest differences between the child-parent correlations.

Apart from the relative differences of the correlations within one CPT, there also exist absolute differences between the CPTs. The sum of the correlations of the Polar bears BN is considerably larger than the sums of the correlations for both the Food security BN and the Pollinator abundance BN. For experts A and C of the Food Security BN, negative correlations are found between the child node and parent node 1 (Varroa control).

The relationship between the correlations and the optimized parent weights for InterBeta becomes more evident when they are plotted against each other. In Figure 8.2 the normalized values of the correlations between the child and parent nodes are plotted against the normalized values of the parent weights. The correlations and weights are both normalized such that they sum to one, by dividing by the sum of the correlations (or parent weights) of one CPT. This is done to be able to assess the relative magnitudes. There is a strong positive correlation present.

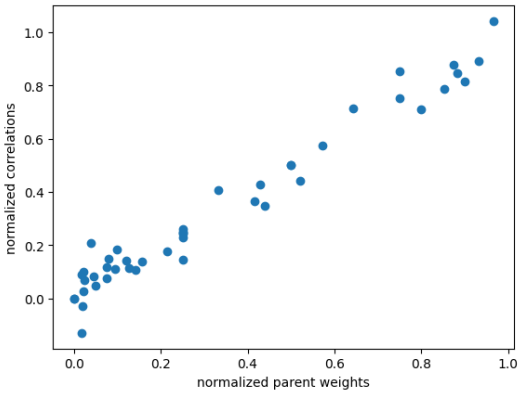


Figure 8.2: Normalized child-parent correlations plotted against normalized optimal parent weights for InterBeta.

Thus, the differences between correlation structures are linked closely to the optimal parent weights for InterBeta and influence the performances of the different versions of InterBeta. Four types of correlation structures between the child node and the parent nodes are chosen, based on the observed correlation structures of the elicited CPTs, to use for the simulations in this chapter:

- *Equal low (eqL)*: equal correlations between the child node and all parent nodes, with a sum close to 0.7: $(0.23, 0.23, 0.23)$. Similar to Pollinator abundance expert C.
- *Equal high (eqH)*: equal correlations between the child node and all parent nodes, with a sum close to 1.1: $(0.37, 0.37, 0.37)$. Similar to Disturb, OthMor, BioStr.
- *Increasing (incr)*: correlations are relatively increasing, summing to : $(0.15, 0.3, 0.45)$. Similar to Pollinator abundance expert G.
- *Outliers (out)*: correlation for one parent significantly larger than for the other: $(0.1, 0.1, 0.8)$. Similar to SASur and AdSur.

The correlations between the parent nodes are set to zero. This means that the correlation matrix between the parent nodes is equal to the identity matrix. Thus the following Spearman rank correlation matrices are used:

$$\text{EqL:} \begin{bmatrix} 1 & 0 & 0 & 0.23 \\ 0 & 1 & 0 & 0.23 \\ 0 & 0 & 1 & 0.23 \\ 0.23 & 0.23 & 0.23 & 1 \end{bmatrix} \quad \text{EqH:} \begin{bmatrix} 1 & 0 & 0 & 0.37 \\ 0 & 1 & 0 & 0.37 \\ 0 & 0 & 1 & 0.37 \\ 0.37 & 0.37 & 0.37 & 1 \end{bmatrix}$$

$$\text{Incr:} \begin{bmatrix} 1 & 0 & 0 & 0.15 \\ 0 & 1 & 0 & 0.30 \\ 0 & 0 & 1 & 0.45 \\ 0.15 & 0.30 & 0.45 & 1 \end{bmatrix} \quad \text{Out:} \begin{bmatrix} 1 & 0 & 0 & 0.10 \\ 0 & 1 & 0 & 0.10 \\ 0 & 0 & 1 & 0.90 \\ 0.10 & 0.10 & 0.90 & 1 \end{bmatrix}$$

8.2. CPT SIMULATION METHOD

Given the correlation structures defined above, a multivariate normal distribution is constructed, which is then converted into a Gaussian BN which is convenient for simulations. To specify the multivariate normal distribution, the covariance matrix is calculated from the corresponding Pearson correlation matrix. Once the (continuous) Gaussian BN is specified, for each CPT that is to be simulated, a sample is taken on which a discrete BN is fit.

The first step of the process is to convert the Spearman's correlation (ρ) matrices into Pearson's correlation (r) matrices, using:

$$r = \frac{1}{2} \sin\left(\frac{\pi}{6}\rho\right).$$

To ensure that the found Pearson's correlation matrix is positive semi-definite, the closest positive semi-definite matrix is calculated using the algorithm by Qi and Sun, 2010. Once this is found, it is converted to a covariance matrix (Σ), using:

$$\Sigma = \text{diag}(\sigma) \cdot r \cdot \text{diag}(\sigma),$$

where $\text{diag}(\sigma)$ is the matrix with the variance on the main diagonal and has zero entries outside of the main diagonal. Σ is then the covariance matrix that will be used for sampling. In addition to the covariance matrix, a vector containing the mean values of all the variables is needed. For the simulations, the null vector is used for the mean.

The CPTs are generated with the help of the `R` package `bnlearn` (Scutari, 2010). The variance matrix and mean vector are then used to specify a multivariate normal distribution, which is converted into a Gaussian Bayesian network using the R function `mvnorm2gbn` (Pourahmadi, 2011). From this new continuous BN, a sample is taken, which is then discretized into the wanted number of levels according to the number of parent and child states, using Hartemink's Algorithm (Hartemink, 2001). This method was chosen, as it was previously found to best replicate the correlation structure of BNs (Marcot & Hanea, 2021). Finally, a discrete

BN is fit to the discrete data using `bn.fit()`. The fitted CPT of the child node can then be used for the simulations.

For each replication of the simulation, a new sample is taken from the continuous BN, which is once again discretized such that a discrete BN can be fit to it. This ensures that in each iteration of the simulations a slightly different CPT is generated.

Unless specified otherwise, for each of the simulation studies, the BN structure is kept the same, where the CPT of one child node with three parent nodes is generated. The simulations are all repeated 100 times, and the mean and 95% confidence intervals of the performance measures (mean KL-divergence, and percentage of agreement) are reported. The confidence intervals are determined based on the normal distribution:

$$CI = \left[\bar{X} - \frac{1.96 \cdot S}{100}, \bar{X} + \frac{1.96 \cdot S}{100}\right],$$

where $\bar{X}$ is the sample mean and S^2 is the sample variance.

8.3. ELICITATION BURDEN

The size of a CPT was previously defined in Equation 2.6, and is dependent on the number of child node states s_C and parent node states s_i for each parent $i \in \{1, ..., n\}$. For InterBeta, the elicitation burden also depends on the chosen version/variation:

- best and worst: $2 \cdot s_C$,
- parent weights: $2 \cdot s_C + n$,
- state weights: $2 \cdot s_C + \sum_{i=1}^{n} s_i$,
- row weights: $2 \cdot s_C + \prod_{i=1}^{n} s_i$.

For all versions, the relative elicitation burden with respect to a full CPT elicitation is shown in Figure 8.3. The same structure notation for the BNs is used on the x-axis as in Table 5.1, which is denoted by $(s_1, s_2, s_3) -> s_C$. As the number of parent nodes grows, the differences between the relative elicitation burden of the best and worst, parent and state weights versions decreases. At the same time, the relative elicitation burden of the row weights version stays large.

The relative elicitation burden of the row weights version depends heavily on the number of child states. The following relationship can be found:

$$\text{Relative elicitation burden row weights:} \quad \frac{2 \cdot s_C + \prod_{i=1}^{n} s_i}{s_C \cdot \prod_{i=1}^{n} s_i} = \frac{2}{\prod_{i=1}^{n} s_i} + \frac{1}{s_C} > \frac{1}{s_C}$$

Thus, for the row weights version of InterBeta, at least $\frac{1}{s_C}$ of a CPT needs to be elicited. So, the row weights version is the most burden-reducing for a large number of child states but stays the most burdensome version.

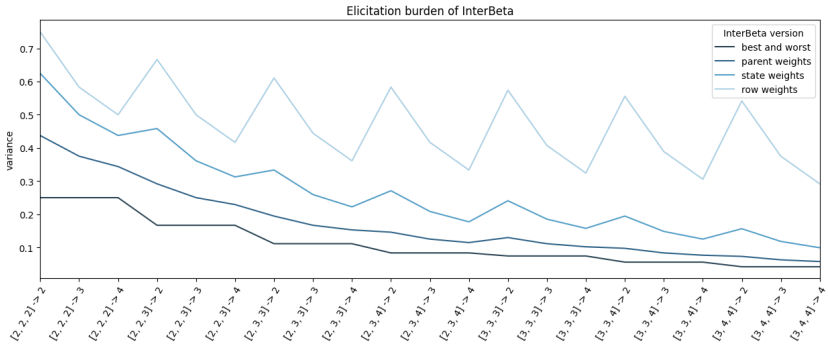


Figure 8.3: Relative elicitation burden for the InterBeta versions, for increasing numbers of parent states and child states. For a fixed structure with one child node with three independent parent nodes.

8.4. ARITHMETIC MEAN VERSUS SHIFTED GEOMETRIC MEAN

The first question that is addressed in this chapter is *How do the arithmetic mean and shifted geometric mean compare in effectiveness for different combinations of parent node state counts and correlation structures?* To answer this question, both means are tested on simulated data with a varying number of parent node states and correlation structures. The BN structure is fixed to one child node with three independent parent nodes. In addition, the number of child node states is fixed to three.

The results are shown in Figure 8.4, based on the mean KL-divergence. The results of the percentage of agreement are shown in Figure C.14, in Appendix C. The number of parent states for each parent are varied between two and four on the x-axis, denoted by (s_1, s_2, s_3), where s_i is the number of states for parent i. Also note that the row weights version of InterBeta is not included in this graph, as this version does make use of the mean function.

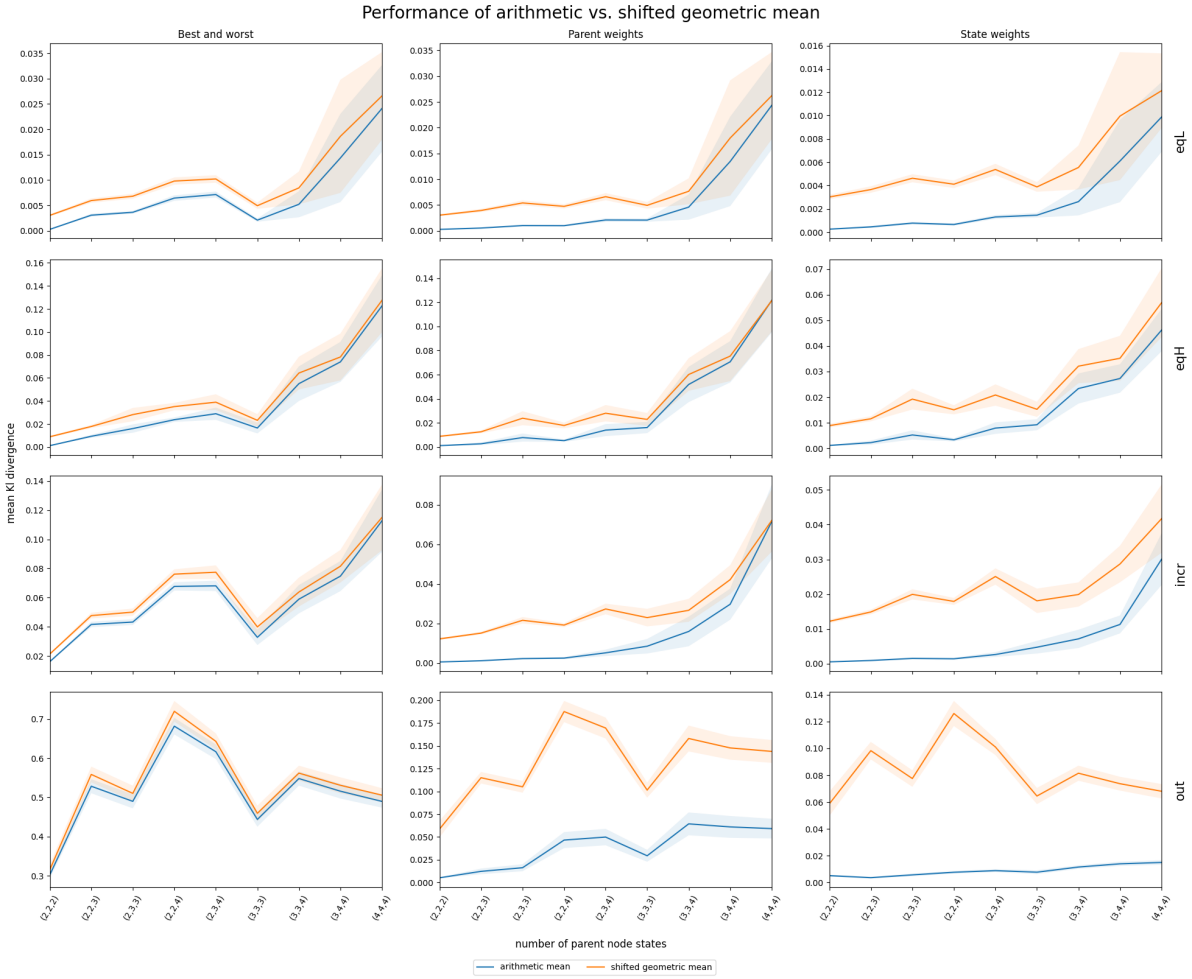


Figure 8.4: Mean and 95% confidence interval of InterBeta performance (KL-divergence), over 100 replications, on simulated data with four different correlation structures. The arithmetic and shifted geometric mean are compared, with α, β as interpolation parameters.

First, the performance differences between the arithmetic mean and the shifted geometric mean are discussed. The arithmetic mean performs better than, or as well as, the shifted geometric mean for all combinations of correlation structure and number of parent nodes. This difference in performance is mainly visible for the parent and state weights versions. For the *eqL*, *eqH*, *Incr* correlation structures, the performance difference becomes smaller as the number of parent nodes increases. When each parent node has three or more states, the 95% confidence intervals become wider and overlap for both mean functions. In future research, the range of the number of parent node states could be enlarged, to see how the graph continues for larger numbers of parent states. For the *Out* correlation structure there is little difference in performance for the best and worst version of InterBeta, but this changes when parent or state weights are added. In that case, the arithmetic has a significantly lower mean KL-divergence.

Considering the percentage of agreement results as shown in Figure C.14 in Appendix C, the arithmetic mean scores better than the shifted geometric mean in most situations. Only for the *Out* correlation structure, when the best and worst version of InterBeta is used, does the shifted geometric mean perform better than the arithmetic mean for certain combinations of parent node state counts. This is mainly seen when the number of parent states is not equal for each parent.

Another thing that stands out is the dip in the graph of the best and worst version results when the parent nodes have three states each. What is especially interesting, is that the best and worst version of InterBeta performs better when all parent nodes have three states, than when the parent nodes have varying numbers of states. When the number of states for each parent node grows larger than three, the graph quickly rises if there is no outlier in the correlation structure, like in *Out*.

In general, for most cases, the arithmetic mean should be chosen over the shifted geometric mean. Only when the number of parent states is not equal for each parent, the correlation structure is assumed to have an outlier, and the best and worst version of InterBeta is chosen; it is better to use the shifted geometric mean. However, when it is known that there is an outlier, it is recommended to use (parent) weights, which means that the situation for which the shifted geometric mean is better should not occur. Therefore, based on the performed simulations, the arithmetic mean is the preferred mean when the parent nodes have at most four states.

8.5. NUMBER OF PARENT STATES VERSUS NUMBER OF CHILD STATES

After comparing the performance of the arithmetic mean and shifted geometric mean, the following simulation study is performed to investigate the influence of varying the number of child node states and parent node states simultaneously. The structure of the BN is fixed to one child node with three independent parent nodes, where each parent node has an equal amount of states s_p. Throughout this simulation study, the arithmetic mean is used as was concluded in the previous section.

Let's start with the performance results in terms of the mean KL-divergence of the *eqL* correlation structure in Figure 8.5a. For the best and worst, and parent weights versions of InterBeta, there is a clear distinction in performance when different numbers of parent states are considered. Only once the number of child states reaches six, the confidence intervals of the two and three parent states scenarios start to overlap. The largest difference in performance between subsequent scenarios is when the number of parent states is increased from three to four. For the state and row weights versions, the performance difference for different numbers of parent states is less pronounced.

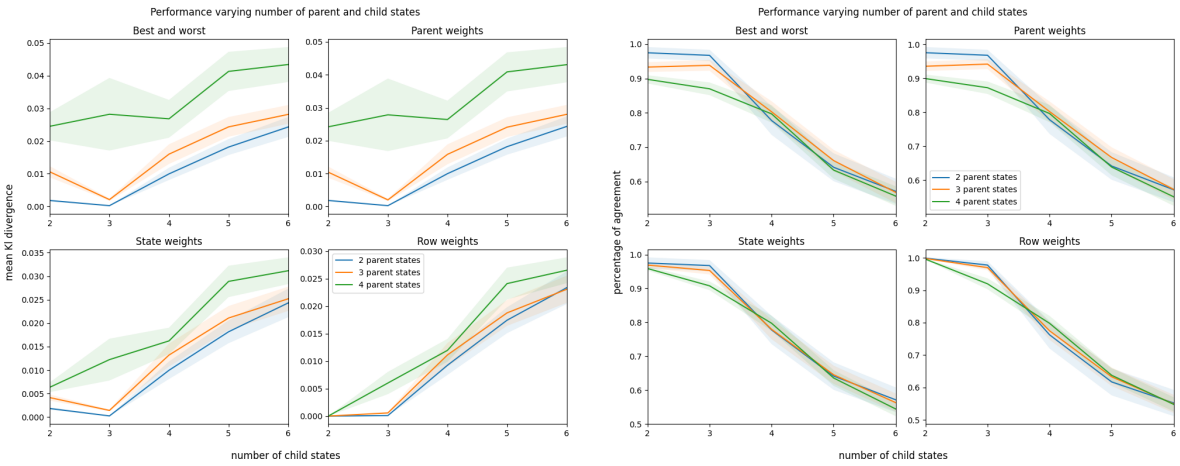
In general, as the number of child node states increases, the performance deteriorates. However, this is only seen when the number of child states becomes greater than three. When the number of child states grows from two to three states, the KL divergence decreases when the parent nodes have less than four states.

The performance measured in terms of the percentage of agreement is shown in Figure 8.5b. There is only a clear distinction between the lines when the child node has three states or fewer. So, in terms of the percentage of agreement, if the child node has more than three states, increasing the number of parent node states has no significant effect on the performance. In addition, it is once again found that, if the number of child states is increased from two to three states, the performance does not decrease significantly.

In Figure 8.6 the performance results of applying InterBeta to reconstruct simulated CPTs with an *eqH* correlation structure are shown. For both the mean KL-divergence and the percentage of agreement results, there is a significant gap between the four parent states scenario and the two or three states scenarios. For the KL-divergence, in Figure 8.6a, this is especially apparent. Increasing the number of parent states from three to four states has a significant effect on the performance, and thus should be done with care. What also is remarkable, is that the graphs are not all (only) increasing. For the four parent node states scenario, the curve starts to decline when the number of child states reaches four or five.

Now, consider the results in terms of the percentage of agreement, in Figure 8.6b. If the BN has parents with at most three states, increasing the number of child node states from two to three does not significantly decrease the performance, both in terms of the mean KL-divergence and the percentage of agreement. Increasing the number of parent states from three to four does have a significant effect.

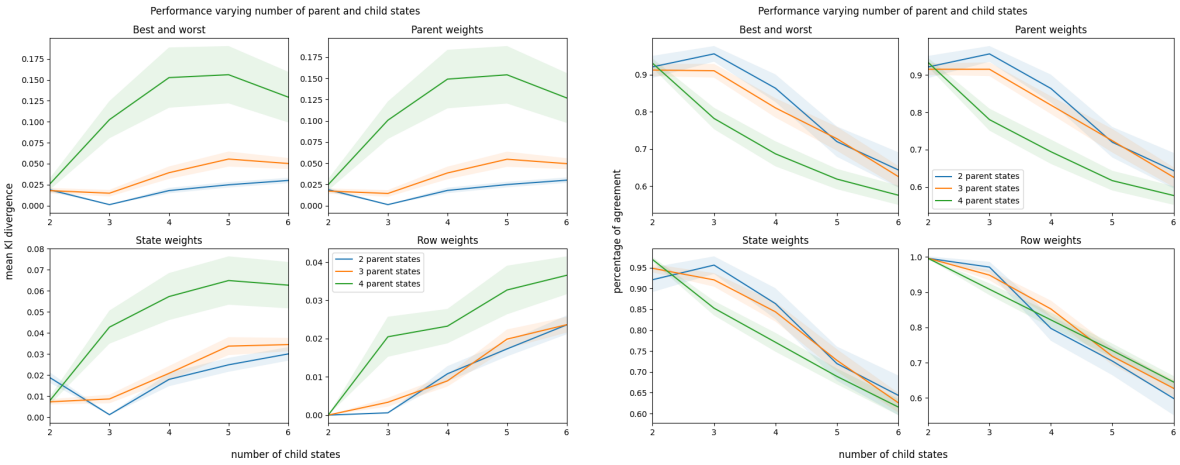
When the correlation structure is changed to *Incr*, the results still lead to some of the same conclusions. In Figure 8.7a it can be seen that the performance is not necessarily monotone decreasing with respect to the number of child nodes. There is also still mainly a large difference between the situation with three parent states and four parent states. In particular for the state and row weights version of InterBeta, increasing the



(a) Mean KL-divergence.

(b) Percentage of agreement.

Figure 8.5: Mean and 95% confidence interval of InterBeta performance, over 100 replications, on simulated data with *Equal low* correlation structure. The arithmetic mean is used and α, β were used as interpolation parameters.



(a) Mean KL-divergence.

(b) Percentage of agreement.

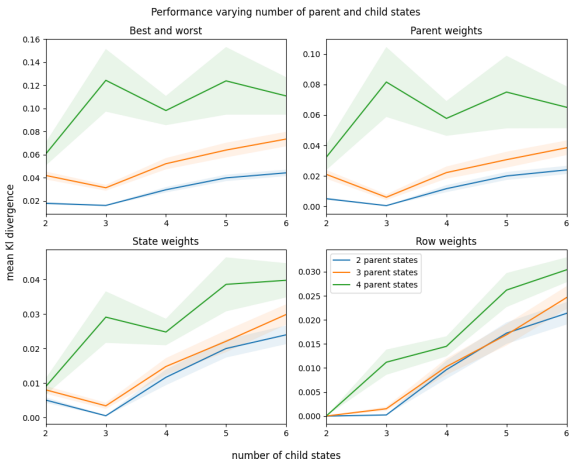
Figure 8.6: Mean and 95% confidence interval of InterBeta performance, over 100 replications, on simulated data with *Equal high* correlation structure. The arithmetic mean is used and α, β were used as interpolation parameters.

number of parent states from two to three does not lead to a significant decrease in performance.

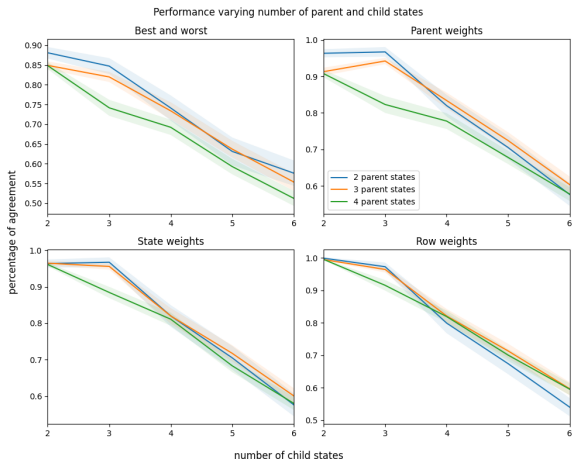
Focusing on the performance measured by the percentage of agreement, as shown in Figure 8.7b, there is much overlap between the situations. If the child node has at least four states, there is no significant difference in the percentage of agreement for the different numbers of parent states. When varying the number of child states, increasing the number from two to three leads to the smallest decrease in agreement.

The final correlation structure that is considered is *Out*, of which the results are shown in Figure 8.8. In terms of the mean KL-divergence, as presented in Figure 8.8a, the gap between the four parent states situation and the three parent states situation has decreased in regards to the other correlation structures. For the best and worst, and parent weights versions the graphs decline for more than four child node states. In this case, the effect on the KL-divergence is larger when moving from two parent states to three parent states, than when moving from three to four. For the state and row weights versions, there is no significant difference between the different numbers of parent states, the main difference is determined by the number of child states.

The performance in terms of the percentage of agreement is depicted in Figure 8.8b. The performance is decreasing in terms of the number of child states, but not necessarily in terms of the number of parent states. For the states and row weights versions, the case with two states for each parent results in worse performance than the case with three or four parent states. This highlights the importance of measuring performance



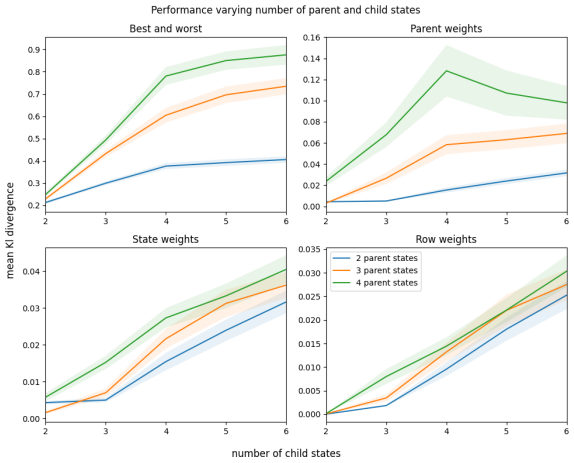
(a) Mean KL-divergence.



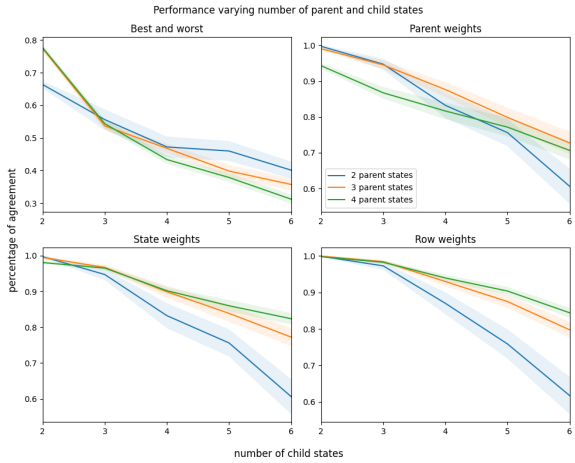
(b) Percentage of agreement.

Figure 8.7: Mean and 95% confidence interval of InterBeta performance, over 100 replications, on simulated data with *Increasing* correlation structure. The arithmetic mean is used and α, β were used as interpolation parameters.

using more than one metric.



(a) Mean KL-divergence.



(b) Percentage of agreement.

Figure 8.8: Mean and 95% confidence interval of InterBeta performance, over 100 replications, on simulated data with *Outlier* correlation structure. The arithmetic mean is used and α, β were used as interpolation parameters.

In addition to the graphs that focus on comparing the influence of the growing number of node states, for both the parent and child nodes, Figures C.15 and C.16, in Appendix C, show individual results of InterBeta for each combination of number of child node states and parent node states. Considering the mean KL-divergence, the largest effect on performance is seen when the child node goes from having three states to having four states. Similarly, for the number of parent node states, the largest impact is seen when shifting from three to four parent node states. In addition, when each parent node has three states or less, and the correlation structure does not have outliers (i.e. a dominating parent node) the difference between the performance of the different InterBeta versions is small. Only when the correlation structure has an outlier, the best and worst version of InterBeta performs significantly worse.

Similar results are obtained when considering the percentage of agreement, as presented in Figure C.16. The performance differences are minimal between having two or three states for each parent node. The largest differences appear when the number of child states becomes four or more.

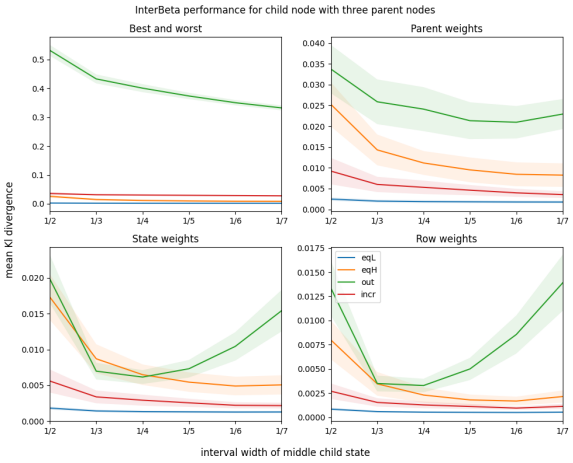
8.6. DISCRETIZATION INTERVAL WIDTHS

One potential 'improvement' of the InterBeta method, that has not been investigated before, is to allow the discretization intervals of the child node to be of different sizes. In the original method, when the beta distribution is fit and discretized, the discretization intervals are fixed to equal widths. In this simulation study, the influence of varying the discretization interval widths is examined.

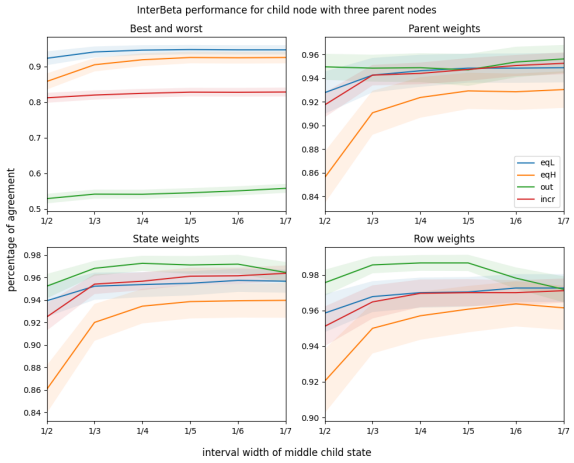
The structure of the BN is once again fixed to having one child node with three independent parent nodes, where the parent nodes and child node each have three states. The arithmetic mean is used with the α and β as interpolation parameters. The width of the middle child node state interval is varied from $I_{mid} = \frac{1}{2}$ to $I_{mid} = \frac{1}{7}$, where the best and worst state have an equal width of $I_{best} = I_{worst} = \frac{1-I_{mid}}{2}$.

The results of the simulations, are shown in Figure 8.9. The performance in terms of the mean KL-divergence can be seen in Figure 8.9a. In most cases, there are significant differences in performance between the different correlation structures. For the *eqL*, *eqH* and *incr* correlation structures, the mean KL-divergence is minimized for a smaller middle state interval width, with a minimum at $1/6$ or $1/7$ for all InterBeta versions. For the *out* correlation structure, if state weights or row weights are used, there is a clear optimal middle state size of $1/3$ or $1/4$.

Similar results can be found for the performance measured by the percentage of agreement in Figure 8.9b. For all InterBeta versions and correlation structures, $I_{mid} = 1/2$ is the worst performing interval width. Overall, the performance remains fairly constant for $I_{mid} \in [1/3, 1/7]$. So, in particular, if an approximation of the correlation structure is not known beforehand, there is not enough evidence to alter the discretization intervals.



(a) Mean KL-divergence.



(b) Percentage of agreement.

Figure 8.9: Mean and 95% confidence interval of InterBeta performance (percentage of agreement), over 100 replications, on simulated data with four different types of correlation structures, where the child node discretization interval width is varied. The arithmetic mean is used and α, β were used as interpolation parameters.

8.7. EXTRABETA: EFFECT OF DOMINANT PARENTS

The final simulation study that is included in this thesis investigates the ExtraBeta version of InterBeta, that was presented in Section 6.1.1. For this study, two additional correlation structures are defined that contain an outlier or in this context a dominating parent.

The two new correlation structures are variations on the *out* correlation structure but with a less pronounced outlier. The most subtle outlier is in the *outL* correlation structure, the child-parent correlation is twice as large for the dominating parent as it is for the other parents. The *outM* correlation structure contains a correlation outlier which is three times as large as the other child-parent correlations. The following two matrices are used as input:

$$\text{outL:} \begin{bmatrix} 1 & 0 & 0 & 0.25 \\ 0 & 1 & 0 & 0.25 \\ 0 & 0 & 1 & 0.5 \\ 0.25 & 0.25 & 0.5 & 1 \end{bmatrix} \quad \text{outM:} \begin{bmatrix} 1 & 0 & 0 & 0.2 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & 0.7 \\ 0.2 & 0.2 & 0.7 & 1 \end{bmatrix}.$$

The results of applying ExtraBeta to reconstruct simulated CPTs are shown in Figure 8.10. Each dot represents the mean accuracy of ExtraBeta over ten repetitions of the method with a set of input rows. The green dots represent the results when the third parent node is fixed to its best state for the "good" row and to its worst state for the "bad" row. As for the other simulation studies, the arithmetic mean is used and the α, β were selected as the interpolation parameters.

For *eqL* and *eqH* there is little effect when the third parent is set to its extreme states. This is mainly because the third parent is a randomly chosen parent node instead of a parent node with a dominating influence on the child node. However, what is remarkable is that the InterBeta results are relatively bad. ExtraBeta has a better reconstruction accuracy than InterBeta for more than half of the tested combinations of input rows. The trend that was found for ExtraBeta in Chapter 7, that the accuracy improves as the difference in means of the input rows increases, is also found here.

When the *incr* correlation structure is focused on, the effect of setting the third parent to its extremes becomes apparent. The third parent is the dominating parent in this case, as it has the highest correlation with respect to the child node variable. The downward trend of the KL-divergence versus the difference between the input multinomial means is present once again. This is mainly visible for the ExtraBeta versions that use parent, state or row weights. For those ExtraBeta versions, fixing the dominating parent to its extreme states results in constructed CPTs that have an accuracy which is not significantly different from the CPTs generated by InterBeta.

For the correlation structures that contain a clear outlier: *outL, outM*, and *out*; the accuracy performance trend is even more visible. Having weights as input greatly improves the CPT reconstruction accuracy in comparison to just using the "good" and "bad" row as input. For the *outL* correlation structure, if the parent, state or row weights version is considered, we see that ExtraBeta with fixed dominating parent states outperforms InterBeta. For *outM* and *out*, the CPT reconstruction accuracy of ExtraBeta with fixed dominating parent states is not significantly different from InterBeta for the parent, state or row weights versions.

So, in general, if one parent node can be identified as a dominating parent, ExtraBeta is a good alternative for InterBeta, as was seen for the *incr, outL, outM* and *out* correlation structures. Of these correlation structures, *incr* has the smallest differences between the parent-child correlations. The dominating parent has a correlation with the child node that is 1.5 times as large as the second parent node's correlation, and 3 times as large as the remaining parent node's correlation. Thus, from these simulations it could be deducted that a dominating parent must have a parent-child correlation which is at least 1.5 times as large as the parent-child correlations of the other parent nodes.

ELICITATION PROTOCOL

Based on the simulation results, a potential elicitation protocol for ExtraBeta is proposed in Appendix C, in Figure C.17. This protocol provides suggestions for what should be elicited, either based on prior knowledge of the relative influence of parent nodes on the child node or, if such knowledge is unavailable, based on elicited weights of the parent nodes. If it is found that there are significant differences between the parent influences, this will be used to guide experts what input CPT rows can be chosen to assess. In practice, if experts have an appropriate understanding of CPTs, an empty CPT can be used in the elicitation. In the table, the potential "good" and "bad" rows can then be highlighted, from which the experts can choose two to assess.

After two input rows have been elicited from the experts, these can be checked by the modeller. If it is found that the "good" and "bad" rows are similar, in the sense that the means of the two rows are not very different, it can be considered to repeat the input row elicitation process. Otherwise, the elicited input rows (and weights) can be used as input for ExtraBeta.

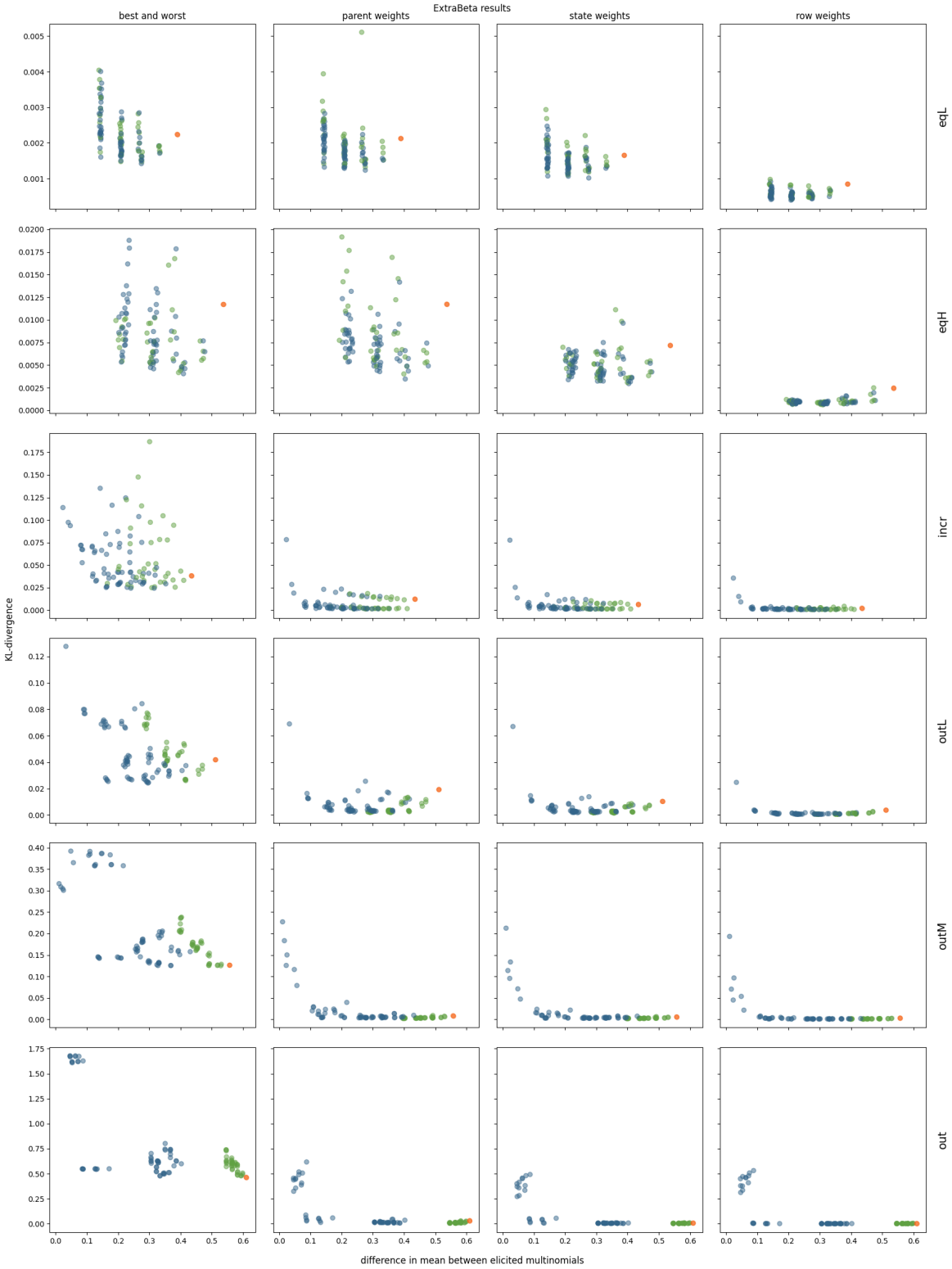


Figure 8.10: Mean results of reconstructing 10 repetitions of simulated CPTs using ExtraBeta (arithmetic, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results when the good row has the dominating parent in its best state, and for the bad row the dominating parent is in its worst state (green), and the results for remaining combinations (blue).

9

DISCUSSION

This chapter contains a discussion of the entire research. The discussion topics will follow the order of the thesis sections, starting with the theoretical foundation and finishing with the simulation results. After the main findings, possible limitations, and implications are discussed; a set of recommendations are given for future research.

Theoretical background To start, an overview is given on the theory behind BNs. This chapter highlighted the presence of BN applications in literature and what challenges exist in BN applications. It was found that in particular static discrete BNs are often used, and the use of experts for BN quantification is common. The challenges found included those related to the size of the BNs. This highlights the importance of studying elicitation burden-reducing methods. One aspect that is missing is an overview of the sizes that CPTs take in practice.

An overview of expert judgment followed, which started with an overview of heuristics and biases that exist when experts reason and answer questions under uncertainty. During elicitations, the goal is to minimize the presence of bias, for which SEJ methods have been developed to ask questions in a structured way. This is well-defined for assessing frequencies of events for example, but to a lesser extent when weights are considered. Because weights are relative by definition, the elicitation of weight parameters is subject to anchoring. The first assessed weight will likely serve as an anchor point for experts to determine the other weights. This should be taken into account when weights are assessed, for example by having experts order the weights before assessment. Otherwise, indirect methods can be used from which weights can be calculated.

Most of the discussed SEJ methods elicit quantiles. Thus, not only a best estimate is assessed, but also a 5th and 95th percentile are elicited. These can be used to help limit bias, and to calibrate experts. However, these extra assessed percentiles are not further used to quantify the CPTs.

The chapter continued with an overview of elicitation formats, which concluded with the notion that visualization helps elicitation. For example, by using scales or probability wheels, experts can be helped to assess probabilities or frequencies. Similar visual tools could be developed to elicit other types of parameters such as weights, in addition, verbal anchors could be thought of that help experts quantify the strength of influences.

The expert judgment part is finished with a discussion on the parameter elicitation burden. The elicitation burden does not only depend on the number of parameters that need to be assessed but should also depend on the complexity of the elicited parameters. In this thesis, the elicitation burden was taken to be equal for all types of parameters, as a measure of elicitation burden for different types of parameters does not exist. Apart from the burden, the assessment of complex parameters may also be more prone to errors and biases than more 'natural' parameters. For a fair comparison of different CPT construction methods, it would be beneficial to make a distinction between the complexity of different types of parameters.

CPT construction methods The most straightforward way to quantify a CPT, for which no data is available, is to have all probabilities assessed by experts. Four more methods that decrease the burden on experts

are introduced. The first method that was considered is the Noisy-OR/MAX method. This method was not further considered in the application phase of the research, as it is not able to capture the joint effects of parent nodes. In addition, the elicitation is complex and can lead to confusion among experts.

The next CPT construction method is RNM. The main limitation of the method is that the elicitation process contains a trial and error phase to determine weights and a variance parameter. This is addressed by the AutoRNM extension. Another limitation is the fact that a single variance parameter is used for determining all CPT rows. Allowing the variance to be flexible would enable the method to distinguish between intermediate and extreme rows.

For the InterBeta method, limitations included no flexibility in the rows that need to be assessed by experts, interpolation between the best and worst row being linear, and having set weights of zero for the worst parent states. The first limitation is addressed by the ExtraBeta variation, and the second by eliciting middle rows in addition to the best and worst row. The third limitation was explored briefly, by allowing non-zero weights to be given to the worst state, but this did not improve upon the state weights version and thus was not incorporated in the thesis.

The final method included in the overview is Functional Interpolation, of which the main limitations were identified as a lack of weight parameters and flexibility in distribution choice. In the application phase, other distributions were tested. However, the inclusion of weight parameters was not tested, as the method already requires more parameters to be elicited than RNM and InterBeta in most cases. Only for very large CPTs with many parent states, should the effect of the inclusion of weight parameters on reconstruction accuracy, be investigated.

Implementations of methods The RNM, Functional Interpolation method, and InterBeta were implemented in Python. In addition, algorithms were written to find optimal weights to use for RNM and InterBeta. These algorithms included grid search methods for finding integer weights and greedy search algorithms to find non-integer type parameters. This means that the accuracy of the optimized weights was also dependent on the number of iterations that the algorithm takes, which was chosen such that algorithms could finish in a reasonable time (i.e., within 10 hours for one simulation study).

It would be possible to increase the number of iterations to increase the potential accuracy, but the use of optimally fitted weights to reconstruct CPTs is already questionable on its own. In a real-life application, experts will be used to assess the weights. It is not known with what accuracy experts can assess weights and variance parameters. In this thesis, it is assumed that the most precise weight assessments by the experts are rounded to the nearest half. For row weights, this is adjusted to two decimals.

Case study The extensions that were developed were tested to reconstruct fully elicited CPTs. The fully elicited CPTs were part of three BNs, that modeled the abundance of pollinators in the UK, the persistence of Polar Bears, and food security in Victoria, Australia. These elicited CPTs were assumed to be the "true" CPTs, and it was tested how well the CPT construction methods were able to reconstruct these true CPTs. However, it is not known how "true" these fully elicited CPTs are, for example, due to the fatigue of experts after assessing many values, part of the CPTs could have been assessed less thoroughly. This fatigue plays less of a role in the CPT construction methods, as only a fraction of the number of parameters need to be elicited. Thus, although the CPT construction methods cannot exactly produce the same CPTs as were obtained from the full elicitation, this does not immediately imply that the constructed CPT by one of the construction methods is less "true".

Nevertheless, the RNM, InterBeta, and Functional Interpolation method will be applied to reconstruct the elicited CPTs as accurately as possible. The results of the individual methods with their extensions will be first discussed separately before the comparison of methods is discussed.

To start, each of the InterBeta versions: best and worst, parent weights, state weights, and row weights; were applied to reconstruct the fully elicited CPTs. The harmonic, geometric, shifted geometric and arithmetic mean were considered to calculate row weights, which were then used to either interpolate the α/β or the mean/variance of the beta distribution. For all methods, it was found that the geometric and harmonic mean do not perform well. This can be explained by the fact that they map a large part of the CPT rows to the worst row. The results of using the arithmetic or shifted geometric mean were close together for all versions. There was not enough evidence to decide which mean performed better in general.

Considering the interpolation parameters, it was found that it depended on the CPT whether the α/β or the mean/variance could reconstruct CPTs better. In particular for the row weights version, the perfor-

mance difference between interpolating the α/β or the mean/variance is found to be varying per CPT. From this analysis alone, no characteristics of CPTs were found to predict better performance for one of the sets of interpolation parameters. Only the size of the CPT could be identified as a factor, as α/β interpolation performed better than mean/variance interpolation for all CPTs with more than 81 values.

Following the original InterBeta methods, a first extension is investigated, where in addition to the best and worst rows also middle rows are elicited. The hypothesis was that this would give more freedom to the method to find fitting beta distribution parameters. This showed to only be true for the row weights version and in a few cases for the parent and state weights versions. It was therefore concluded that it was not worth further testing this on simulated CPTs. What could be of interest is to choose a different set of input rows, for example, based on the Functional Interpolation method.

To increase the input flexibility of InterBeta, the ExtraBeta variation was implemented. This version does not force experts to assess the best and worst row but instead allows other good and bad rows to be used as input. For example, when an expert is asked to assess the best row which happens to be a very rare situation. This can cause the expert to use another situation which is in their frame of knowledge, to anchor to and adjust to find the best row. This is prone to biases. If instead, the expert would have been allowed to assess a CPT row they are comfortable with, this bias could have been avoided.

The final part of the InterBeta investigation contains a comparison of interpolated beta distribution parameters to optimally fitted parameters for each CPT row. This showed that interpolating the α/β approximates the optimal fit parameters better than when the mean/variance is interpolated. This investigation was a cause to investigate the variance curves further. In Appendix B.2 it is shown that the variance curve, calculated from α/β interpolation, is not necessarily concave for all values of the mean. However, it was also shown that for almost all simulated CPTs the variance curve of the α,β interpolation lies above the linearly interpolated variance.

The RNM and its extension, AutoRNM, had a very similar CPT reconstruction accuracy. The two methods were also compared in terms of elicitation burden, this posed a challenge as the original RNM depends on a trial and error phase to find weight parameters. Therefore, it was chosen to assume that the experts would be able to find fitting weights in two trials for the comparisons. This made the original RNM much less burdensome to elicit than AutoRNM for most cases. If the difference in elicitation complexity is also taken into account, when comparing weights and probabilities, for example, the burden is even harder to compare. Thus, if experts are not comfortable assessing weights and a trial-and-error-based process is not appropriate, the AutoRNM poses a good alternative.

The final method that was tested on elicited data was Functional Interpolation, for which three distributions were implemented: beta, normal, and truncated normal. It was found that the beta distribution performed the best, and was better fitting even on a row-by-row level better fitting to multinomials.

The final part of the case study is the comparison between all of the methods, where both the CPT reconstruction accuracy and elicitation burden are taken into account. It was found that the parent weights version of InterBeta is the best method. As it scores among the best in terms of accuracy whilst requiring the second-fewest input parameters. The Functional Interpolation method remains interesting for constructing large CPTs, but for what size this is the case is not clear yet. As the InterBeta method and Functional Interpolation have some similarities, the two methods could also be combined, to create a version of InterBeta that uses the same input rows as Functional Interpolation, but also allows for weights to be elicited. For what CPT construction error the increased burden is "worth it" remains an unanswered question.

Simulation study The best-performing method, InterBeta, was further analyzed through a series of simulation studies. To start, the correlation structures present in the elicited data were investigated. A strong correlation was found between the child-parent correlations and the optimized InterBeta parent weights. This suggests that instead of eliciting weights, the correlation structure can also be elicited and used as input. In future studies this can be studied in more detail, to not only consider the relative correlation strengths but also if the absolute correlation strength has an effect on the parent weights.

The simulation study started by comparing InterBeta results when using the arithmetic mean and the shifted geometric mean. Overall, the arithmetic mean was found to be the better choice. However, the simulation was only performed for parent nodes with a maximum of four states each. The accuracy results were found to converge for the two means as the number of states increased, from which the question arises, what happens for parent nodes with more than four states? In future analyses, this can be investigated as well.

Another aspect that stood out, was that InterBeta performs better when all parent nodes have three states, than when one of the parent nodes has two states and the others have three for example. In the following

simulation, it was also found that increasing the number of child states from two to three states leads to an improvement in accuracy in many cases. One hypothesis for this behavior is due to the flexibility of fitting the best and worst rows. If there are only two child node states, the beta distribution is fit on a multinomial with two values, then there are many α, β that can accurately represent this multinomial. However, as the number of values in a multinomial rise, the number of beta distributions that can accurately represent this multinomial decreases rapidly. So, if a child node has only two states, then there is a lot of ambiguity present for the intermediate rows. How this works exactly would be material for future work. Nevertheless, it can be concluded that simplifying a CPT by reducing the number of node states does not necessarily imply an increase in performance.

Then, the discretization intervals that were used to fit and discretize the beta distribution were varied. It was found that varying the width of the intervals does have an effect on the accuracy of InterBeta, but that this is dependent on the correlation structure of the CPT. So, if this correlation structure is not known before elicitation, there is not enough evidence to alter the discretization interval widths. Including varying widths would increase the complexity of InterBeta without much accuracy gain, therefore I do not think this is an aspect worth researching further.

The final simulation study concerned the effect of dominant parents on the accuracy of ExtraBeta. It was found that when there is a correlation structure with a significantly dominating parent node, and either parent, state or row weights are used, the results of ExtraBeta (with the dominating parent fixed to its extreme states in the input) are as good as the results of InterBeta. Also for correlation structures without a dominating parent, the ExtraBeta method performs at least as well as InterBeta for a large number of input rows. In future studies, the proposed elicitation guidelines should be tested and defined in more detail. In addition, for example for the parent weights version, it would be worth investigating how the optimized parent weights change as different CPT rows are used as input for ExtraBeta.

9.1. RECOMMENDATIONS

Some recommendations follow from the discussion above. Most of these recommendations have already been introduced in the discussion but in this section they are summarized in a list, for clarity. Some of the following recommendations could be combined in one large case study.

- **Including uncertainty assessments:** Currently, none of the CPT construction methods explicitly use the elicited 5th and 95th percentiles of experts. Although these are used to guide experts in the elicitation process, it is wasteful to not further consider these assessments when constructing CPTs. For example, if the best and worst row assessments of a group of experts are aggregated before applying the InterBeta method, the beta distribution can be fit in a different way that includes uncertainty.

- **Guidelines for the elicitation of weight parameters:** Many of the methods that can be used to construct CPTs rely on experts being able to assess weight parameters. It should be further researched how well experts can assess weights, and what tools should be used to help weight elicitations. For example, visual guides and verbal anchors can be developed. This could be investigated as part of a case study where experts are asked to assess weights in different ways. The accuracy of the assessments can be compared as well as the elicitation experiences of the experts, such that a burden/accuracy trade-off can be made.

- **Burden measure of parameters:** At the moment there is no measure to compare the elicitation burden of different types of parameters. To be able to make a more accurate comparison of the elicitation burden of different CPT construction methods, it should be investigated whether large differences in elicitation burden exist between parameters. If so, how would these differences impact the burden/accuracy trade-off of the investigated methods? This could also be part of a case study, where experts are asked to assess different types of parameters. Possible ways to measure the burden include: timing the assessments, including a survey that questions the experts' experiences, and the parameters can be elicited as part of a calibration exercise that is scored afterwards.

- **Flexible variance for RNM:** Although InterBeta was determined as the best performing CPT construction method in this thesis, it remains interesting if RNM can benefit from having a non-constant variance parameter. From the row-by-row investigation of beta distribution parameters, it was found that the variance is often higher for the intermediate rows than for the best and worst rows. Therefore, having the variance be specified by a quadratic function could be of interest to RNM and AutoRNM.

However, this may be complex to elicit for RNM, so it could be considered to elicit a minimal and maximal variance parameter for the CPT instead. For AutoRNM, a variance curve can be fit on the elicited CPT rows.

- **Case study of InterBeta:** The performance of InterBeta on previously elicited CPTs and simulated CPTs was shown to have potential. But, whether this remains promising for applications with actual experts, should be further tested by a case study. In an ideal situation, the case study would both include InterBeta elicitations of all versions, including the ExtraBeta variation, and full elicitations of CPTs. This would require a substantial number of experts to be involved, such that multiple groups of experts can be formed that each complete a different elicitation protocol. This is likely to be an unfeasible requirement, even if only the parent weights version is tested, at least three groups of experts would be necessary. One group would assess the best row, worst row and parent weights of InterBeta, one group would assess the full CPT, and the last group would need to follow the ExtraBeta elicitation protocol. As an alternative one can consider a calibration exercise with students.

- **Combining InterBeta and Functional Interpolation:** The comparison of the CPT construction methods when applied to reconstruct fully elicited CPTs showed that the Functional Interpolation method can reconstruct CPTs with high accuracy. The downside of the method is the relatively large elicitation burden, but this relative burden decreases as the to-be-reconstructed CPTs grow. Therefore, the Functional Interpolation method could also be combined with InterBeta, to create a version with multiple elicited rows and the option to input weight parameters. How this method performs and for what CPT size the method is useful, could be researched in future work.

- **Weights versus correlations:** Finally, it was found that parent weights and child-parent correlations could potentially be used interchangeably as input for InterBeta. In future research, this should be studied in more detail. Extensive simulation studies could be performed to study CPT reconstruction accuracy differences between using parent weights or correlation structures as input. Further studies can also investigate a possible relationship between the state weights and correlations.

- **Dependent parents:** The correlation structures that have been used in the simulation studies all define zero correlation between the parent nodes. It would be of interest to investigate whether adding dependencies between parents changes InterBeta performances.

- **ExtraBeta dominating parents and weights:** From the simulation study, it was found that when one parent node has a parent-child correlation which is 1.5 times as large as the other parent-child correlations, it could be identified as the dominating parent. In future studies this should be tested, to find out how much more influential a dominating parent exactly has to be, perhaps both in terms of correlations and parent weights. In addition, the optimized parent (or state/row) weights that are found for each set of input rows could be compared, to investigate if a different elicitation protocol would be needed for InterBeta and ExtraBeta.

- **Meaning of KL-divergence:** The mean KL-divergence was used as the primary measure to compare two CPTs in this thesis. A KL-divergence close to zero is aimed for, but it is not known what a "small" KL-divergence is. It would be interesting to assess how the KL-divergence between two multinomials (with an equal number of categories) varies, as one of the multinomials is gradually changed. This would

10

Conclusion

When no data is available to parameterize Bayesian Networks (BNs), the Conditional Probability Tables (CPTs) can be specified by experts instead. This thesis aimed to investigate the burden/accuracy trade-off of CPT construction methods. Three questions were formulated:

- How do existing methods, such as RNM, InterBeta and Functional Interpolation compare against each other, in terms of accuracy and elicitation burden, when applied to reconstructing existing fully elicited CPTs?
- How can each method be improved, to offer more flexibility, improve accuracy, or limit the expert burden?
- How should the InterBeta method be tailored given the network structure, underlying correlation structure, or other factors?

For each of the methods, potential improvements were tested against the original versions of the method, and part of the tested potential improvements were found to be of interest. To begin, the shifted geometric mean was added to the possible mean functions of InterBeta, where a constant $\delta = 1$ was added to all entries before applying the geometric mean and finally subtracted from the result. This new mean function resulted in a better reconstruction accuracy than the arithmetic, geometric, and harmonic mean for certain CPTs.

A second extension to the InterBeta method was to include intermediate CPT rows as input. The middle rows that were chosen as input were chosen such that all parent nodes were in their middle states. In general, this did not perform as well as the original InterBeta method for reconstructing CPTs accurately. The increased elicitation burden that is required for eliciting middle rows makes the version not competitive to the original InterBeta method.

A final investigated extension to the InterBeta method was to allow for other rows than just the best and worst to be elicited, which was named ExtraBeta. This new version was able to perform as well as InterBeta for certain combinations of input rows. By choosing input rows that have vastly different distributions, with means that differ by at least 0.5, the accuracy was the closest to the original InterBeta method. If a dominating parent is present, that has a significantly greater influence on the child node than the other parent nodes. Setting this node to its extreme states for the input rows guarantees that the means of the two input rows are far apart.

The RNM was extended to AutoRNM, replacing the trial and error procedure for finding weight and variance parameters by eliciting CPT rows, which were then used to algorithmically optimize fitting weights and variance. The accuracy of AutoRNM was found to be very similar to the accuracy of the original RNM. Therefore, depending on what experts are comfortable with, both methods could be used interchangeably.

Finally, the Functional Interpolation method was extended to use the truncated normal and beta distributions in addition to the normal distribution. It was found that the beta distribution was best able to accurately reconstruct CPTs. Fitting the distributions to individual CPT rows also showed that the beta distribution is best able to represent multinomials.

Following the separately tested extensions, the accuracy of reconstructing previously elicited CPTs with respect to the elicitation burden was compared for all CPT construction methods. The accuracy was set out

against the number of parameters that each method requires as input. It was found that the InterBeta method was the preferred method. In particular, the parent weights version was found to reconstruct CPTs with good accuracy and a relatively low number of input parameters. The best and worst version of InterBeta is the only method that requires less parameters to be elicited than the parent weights version but has a considerably worse accuracy.

It was found that using the α, β as interpolation parameters gives more accurate reconstructions of CPTs than when the mean and variance are interpolated. The arithmetic and shifted geometric means were found to be the best-performing mean functions. Since the ExtraBeta version was able to reconstruct CPTs as well as the original InterBeta method for certain input rows, this version is also still considered. The extended version where middle rows were elicited was not considered a good alternative.

The final research question was addressed by applying the InterBeta method to reconstruct simulated CPTs, with correlation structures based on the fully elicited CPTs that all methods were previously tested on. To be specific, the following questions were answered:

1. How do the arithmetic mean and shifted geometric mean compare in effectiveness for different combinations of parent node state counts and correlation structures?

2. Considering different correlation structures, what is the influence of a growing number of parent states and child states on the performance of InterBeta? Which InterBeta versions are optimal for different combinations of parent and child state counts?

3. Does varying the discretization intervals of the child node influence the performance of InterBeta?

4. What guidelines can be formulated for the ExtraBeta method, to help experts choose appropriate input rows?

For each of the questions, the BN structure was kept fixed, with one child node with three independent parent nodes. The number of states for the parent and child nodes was varied, as well as the correlation structure used to simulate the CPTs. It was first found that the normalized correlations between the child and parents of the elicited CPTs are highly correlated with the normalized parent weights that InterBeta uses.

For the first question, the child node was fixed to having three states and the number of states for each parent node was varied between two to four. The arithmetic mean had a better accuracy than the shifted geometric mean for most scenarios. If there is no outlier in the correlation structure, the difference in accuracy between the two mean functions decreases as the number of parent states increases.

In general, a growing number of parent states and child states has a negative influence on the accuracy of InterBeta. It was found that the largest decrease in accuracy is found when the number of parent node states or child node states is increased from three to four.

Regarding the optimal versions of InterBeta for the different situations, in general, the parent weights version remains the safest choice, as it guarantees an improvement on the best and worst version. However, if there is a reason to believe that there are parent nodes which are much more influential on the child node than others, a more educated choice can be made. If the correlations between the child and all of its parents are equal, then there is no performance difference between the parent weights version and the best and worst version of InterBeta. If the correlations between the child and parents are not equal, the best and worst version should not be considered. The state weights version becomes a viable option when the number of parent states is three or larger, it has the largest impact on accuracy when the correlations between the child and its parents are non-equal.

Varying the discretization interval widths does affect the accuracy of InterBeta, but this differs for distinct correlation structures. The optimal discretization width also varies for the different correlation structures. A minor preference was found for slightly decreasing the width of the discretization interval of the middle child state. This was not found to be enough evidence for altering the discretization widths for all situations.

Finally, the ExtraBeta method was tested on simulated data. It was found that ExtraBeta is most appropriate for situations where the parent nodes are thought to have different levels of influence on the child node. In that case, fixing the dominating parent in the elicitations can guarantee that appropriate input rows are elicited.

BIBLIOGRAPHY

Agena. (2018). Agenarisk.

Albert, I., Donnet, S., Guihenneuc-Jouyaux, C., Low-Choy, S., Mengersen, K., & Rousseau, J. (2012). Combining expert opinions in prior elicitation. *Bayesian Analysis, 7*, 503–532. https://doi.org/10.1214/12-BA717

Atwood, T. C., Marcot, B. G., Douglas, D. C., Amstrup, S. C., Rode, K. D., Durner, G. M., & Bromaghin, J. F. (2016). Forecasting the relative influence of environmental and anthropogenic stressors on polar bears. *Ecosphere, 7*. https://doi.org/10.1002/ecs2.1370

Aulia, R., Tan, H., & Sriramula, S. (2021). Dynamic reliability analysis for residual life assessment of corroded subsea pipelines. *Ships and Offshore Structures, 16*, 410–422. https://doi.org/10.1080/17445302.2020.1735834

Barons, M. J., Hanea, A. M., Wright, S. K., Baldock, K. C., Wilfert, L., Chandler, D., Datta, S., Fannon, J., Hartfield, C., Lucas, A., Ollerton, J., Potts, S. G., & Carreck, N. L. (2018). Assessment of the response of pollinator abundance to environmental pressures using structured expert elicitation. *Journal of Apicultural Research, 57*, 593–604. https://doi.org/10.1080/00218839.2018.1494891

Barons, M. J., Mascaro, S., & Hanea, A. M. (2022). Balancing the elicitation burden and the richness of expert input when quantifying discrete bayesian networks. *Risk Analysis, 42*, 1196–1234.

Barrios, M., Guilera, G., Nuño, L., & Gómez-Benito, J. (2021). Consensus in the delphi method: What makes a decision change? *Technological Forecasting and Social Change, 163*. https://doi.org/10.1016/j.techfore.2020.120484

Billari, F. C., Graziani, R., & Melilli, E. (2014). Stochastic population forecasting based on combinations of expert evaluations within the bayesian paradigm. *Demography, 51*, 1933–1954. https://doi.org/10.1007/s13524-014-0318-5

Bolt, J. H., & Gaag, L. C. V. D. (2010). An empirical study of the use of the noisy-or model in a real-life bayesian network. *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Methods*, 11–20.

Brown, B. B. (1968). Delphi process: A methodology used for the elicitation of opinions of experts.

Burgman, M., Layman, H., & French, S. (2021). Eliciting model structures for multivariate probabilistic risk analysis. *Frontiers in Applied Mathematics and Statistics, 7*. https://doi.org/10.3389/fams.2021.668037

Cain, J. (2001). *Planning improvements in natural resources management guidelines for using bayesian networks to support the planning and management of development programmes in the water sector and beyond*.

Chen, G., Li, G., Xie, M., Xu, Q., & Zhang, G. (2024). A probabilistic analysis method based on noisy-or gate bayesian network for hydrogen leakage of proton exchange membrane fuel cell. *Reliability Engineering and System Safety, 243*. https://doi.org/10.1016/j.ress.2023.109862

Chen, S. H., & Pollino, C. A. (2012). Good practice in bayesian network modelling. *Environmental Modelling and Software, 37*, 134–145. https://doi.org/10.1016/j.envsoft.2012.03.012

Chockalingam, S., Pieters, W., Teixeira, A., & van Gelder, P. (2017). Bayesian network models in cyber security: A systematic review. *Nordic Conference on Secure IT Systems*, 105–122. http://www.springer.com/series/7410

Christophersen, A., Deligne, N. I., Hanea, A. M., Chardot, L., Fournier, N., & Aspinall, W. P. (2018). Bayesian network modeling and expert elicitation for probabilistic eruption forecasting: Pilot study for whakaari/white island, new zealand. *Frontiers in Earth Science, 6*. https://doi.org/10.3389/feart.2018.00211

Cooke, R. M., & Goossens, L. H. J. (1999). Procedures guide for structured expert judgment. *Project report to the European Commission, EUR*.

Cooke, R. M. (1991). *Experts in uncertainty: Opinion and subjective probability in science*. Oxford University Press, USA.

Corp., N. S. (2007). Netica.

Dalkey, N. C., Brown, B. B., & Cochran, S. (1969). *The delphi method: An experimental study of group opinion* (Vol. 3). Rand Corporation Santa Monica, CA.

Das, B. (2004). Generating conditional probabilities for bayesian networks: Easing the knowledge acquisition problem.

de la Cruz, R., & Kreft, J.-U. (2018). Geometric mean extension for data sets with zeros. *arXiv preprint*. http://arxiv.org/abs/1806.06403

Diez, F. J. (1993). Parameter adjustment in bayes networks. the generalized noisy or-gate. *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, 99–105.

EFSA. (2014). Guidance on expert knowledge elicitation in food and feed safety risk assessment. *EFSA Journal*, *12*, 3734.

Fenton, N. E., Neil, M., & Caballero, J. G. (2007). Using ranked nodes to model qualitative judgments in bayesian networks. *IEEE Transactions on Knowledge and Data Engineering*, *19*, 1241–1251. https://doi.org/10.1109/TKDE.2007.1068

Fenton, N. E., Noguchi, T., & Neil, M. (2019). An extension to the noisy-or function to resolve the 'explaining away' deficiency for practical bayesian network problems. *IEEE Transactions on Knowledge and Data Engineering*, *31*, 2441–2445. https://doi.org/10.1109/TKDE.2019.2891680

Freire, A., Perkusich, M., Saraiva, R., Almeida, H., & Perkusich, A. (2018). A bayesian networks-based approach to assess and improve the teamwork quality of agile teams. *Information and Software Technology*, *100*, 119–132. https://doi.org/10.1016/j.infsof.2018.04.004

Gaag, L. C. V. D., Renooij, S., Witteman, C. L. M., Aleman, B. M. P., & Taal, B. G. (1999). How to elicit many probabilities. *Conference on Uncertainty in Artificial Intelligence*, 647–654.

Gigerenzer, G., & Edwards, A. (2003). Simple tools for understanding risks: From innumeracy to insight. *BMJ*, *327*, 741–744.

Gosling, J. P. (2018). Shelf: The sheffield elicitation framework. Springer New York LLC. https://doi.org/10.1007/978-3-319-65052-4_4

Grime, M. M., & Wright, G. (2016, August). Delphi method. Wiley. https://doi.org/10.1002/9781118445112.stat07879

Hanea, A. M., McBride, M. F., Burgman, M. A., Wintle, B. C., Fidler, F., Flander, L., Twardy, C. R., Manning, B., & Mascaro, S. (2017). Investigate discuss estimate aggregate for structured expert judgement. *International Journal of Forecasting*, *33*, 267–279. https://doi.org/10.1016/j.ijforecast.2016.02.008

Hanea, A. M., Hilton, Z., Knight, B., & Robinson, A. P. (2022). Co-designing and building an expert-elicited non-parametric bayesian network model: Demonstrating a methodology using a bonamia ostreae spread risk case study. *Risk Analysis*, *42*, 1235–1254. https://doi.org/10.1111/risa.13904

Hartemink, A. J. (2001). Principled computational methods for the validation and discovery of genetic regulatory networks.

Hartley, D., & French, S. (2021). Bayesian modelling of dependence between experts: Some comparisons with cooke's classical model. In A. Hanea, G. Nane, T. Bedford, & S. French (Eds.). http://www.springer.com/series/6161

Hassall, K. L., Dailey, G., Zawadzka, J., Milne, A. E., Harris, J. A., Corstanje, R., & Whitmore, A. P. (2019). Facilitating the elicitation of beliefs for use in bayesian belief modelling. *Environmental Modelling and Software*, *122*. https://doi.org/10.1016/j.envsoft.2019.104539

Hemming, V., Burgman, M. A., Hanea, A. M., McBride, M. F., & Wintle, B. C. (2018). A practical guide to structured expert elicitation using the idea protocol. *Methods in Ecology and Evolution*, *9*, 169–180. https://doi.org/10.1111/2041-210X.12857

Henrion, M. (1988). Practical issues in constructing a bayes' belief network. *Uncertainty in Artificial Intelligence*, *3*, 132–139.

Janis, I. L. (1971). Groupthink. *Psychology today magazine*, *36*, 84–90.

Ji, C., Su, X., Qin, Z., & Nawaz, A. (2022). Probability analysis of construction risk based on noisy-or gate bayesian networks. *Reliability Engineering and System Safety*, *217*. https://doi.org/10.1016/j.ress.2021.107974

Kaikkonen, L., Parviainen, T., Rahikainen, M., Uusitalo, L., & Lehikoinen, A. (2021). Bayesian networks in environmental risk assessment: A review. *Integrated Environmental Assessment and Management*, *17*, 62–78. https://doi.org/10.1002/ieam.4332

Kaya, R., & Yet, B. (2019). Building bayesian networks based on dematel for multiple criteria decision problems: A supplier selection case study. *Expert Systems with Applications*, *134*, 234–248. https://doi.org/10.1016/j.eswa.2019.05.053

Kemp-Benedict, E. (2008). Elicitation techniques for bayesian network models. *SEI working paper*. www.seius.organdwww.sei.se

Kim, J. H., & Pearl, J. (1983). A computational model for causal and diagnostic reasoning in inference systems. *International Joint Conference on Artificial Intelligence*.

Kitson, N. K., Constantinou, A. C., Guo, Z., Liu, Y., & Chobtham, K. (2023). A survey of bayesian network structure learning. *Artificial Intelligence Review, 56*, 8721–8814. https://doi.org/10.1007/s10462-022-10351-w

Klayman, J., Soll, J. B., Gonzalez-Vallejo, C., & Barlas, S. (1999). Overconfidence: It depends on how, what, and whom you ask. *Organizational Behavior and Human Decision Processes, 79*, 216–247.

Kleve, S., & Barons, M. J. (2021). A structured expert judgement elicitation approach: How can it inform sound intervention decision-making to support household food security? *Public Health Nutrition, 24*, 2050–2061. https://doi.org/10.1017/S1368980021000525

Knochenhauer, M., Swaling, H., Dedda, F. D., Hansson, F., Sjökvist, S., & Sunnegård, K. (2013). *Nks-293, using bayesian belief network (bbn) modelling for rapid source term prediction – final report*. www.nks.org

Kyrimi, E., McLachlan, S., Dube, K., Neves, M. R., Fahmi, A., & Fenton, N. (2021). A comprehensive scoping review of bayesian networks in healthcare: Past, present and future. *Artificial Intelligence in Medicine, 117*. https://doi.org/10.1016/j.artmed.2021.102108

Laitila, P., & Virtanen, K. (2016). Improving construction of conditional probability tables for ranked nodes in bayesian networks. *IEEE Transactions on Knowledge and Data Engineering, 28*, 1691–1705. https://doi.org/10.1109/TKDE.2016.2535229

Laitila, P., & Virtanen, K. (2020). On theoretical principle and practical applicability of ranked nodes method for constructing conditional probability tables of bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50*, 1943–1955. https://doi.org/10.1109/TSMC.2018.2792058

Laitila, P., & Virtanen, K. (2022). Portraying probabilistic relationships of continuous nodes in bayesian networks with ranked nodes method. *Decision Support Systems, 154*. https://doi.org/10.1016/j.dss.2021.113709

Landuyt, D., Broekx, S., D'hondt, R., Engelen, G., Aertsens, J., & Goethals, P. L. (2013, August). A review of bayesian belief networks in ecosystem service modelling. https://doi.org/10.1016/j.envsoft.2013.03.011

Langer, E. J. (1975). The illusion of control. *Journal of Personality and Social Psychology, 32*, 311–328.

Lenzner, T., Kaczmirek, L., & Lenzner, A. (2010). Cognitive burden of survey questions and response times: A psycholinguistic experiment. *Applied Cognitive Psychology, 24*, 1003–1020. https://doi.org/10.1002/acp.1602

Luo, T., & Liu, Y. (2023). Machine truth serum: A surprisingly popular approach to improving ensemble methods. *Machine Learning, 112*, 789–815. https://doi.org/10.1007/s10994-022-06183-y

Marcot, B. G., & Hanea, A. M. (2021). What is an optimal value of k in k-fold cross-validation in discrete bayesian network analysis? *Computational Statistics, 36*, 2009–2031. https://doi.org/10.1007/s00180-020-00999-9

Marcot, B. G., Steventon, J. D., Sutherland, G. D., & Mccann, R. K. (2006). Guidelines for developing and updating bayesian belief networks applied to ecological modeling and conservation. *Canadian Journal of Forest Research - Revue Canadienne De Recherche Forestiere, 36*, 3063–3074. http://cjfr.mc.ca

Mascaro, S., & Woodberry, O. (2022). A flexible method for parameterizing ranked nodes in bayesian networks using beta distributions. *Risk Analysis, 42*, 1179–1195. https://doi.org/10.1111/risa.13915

Mihajlovic, V., & Petkovic, M. (2001). Dynamic bayesian networks: A state of the art.

Mkrtchyan, L., Podofillini, L., & Dang, V. N. (2016). Methods for building conditional probability tables of bayesian belief networks from limited judgment: An evaluation for human reliability application. *Reliability Engineering and System Safety, 151*, 93–112. https://doi.org/10.1016/j.ress.2016.01.004

Morris, P. A. (1977). Combining expert judgments: A bayesian approach. *MANAGEMENT SCIENCE, 23*, 679–693.

Mosleh, A., & Apostolakis, G. (1986). The assessment of probability distributions from expert opinions with an application to seismic fragility curves. *Risk analysis, 6*, 447–461.

Nickerson, R. S. (1998). Confirmation bias: A ubiquitous phenomenon in many guises. *Review of General Psychology, 2*, 175–220.

O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., Oakley, J. E., & Rakow, T. (2006). Uncertain judgements: Eliciting experts' probabilities.

Olesen, K. G., Kjaerulff, U., Jensen, F., Jensen, F. V., Falck, B., Andreassen, S., & Andersen, S. K. (1989). A munin network for the median nerve-a case study on loops. *Applied Artificial Intelligence, 3*, 385–403. https://doi.org/10.1080/08839518908949933

Page, S. E. (2008). *The difference : How the power of diversity creates better groups, firms, schools, and societies.*

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann. https://doi.org/10.1016/C2009-0-27609-4

Podofillini, L., Mkrtchyan, L., & Dang, V. (2014). Aggregating expert-elicited error probabilities to build hra models. CRC Press. https://www.routledge.com/Safety-and-Reliability-Methodology-and-Applications/Nowakowski-Mlynczak-Jodejko-Pietruczuk-Werbinska-Wojciechowska/p/book/9781138026810

Pourahmadi, M. (2011). Covariance estimation: The glm and regularization perspectives. *Statistical Science*, *26*, 369–387. https://doi.org/10.1214/11-STS358

Prelec, D. (2004). A bayesian truth serum for subjective data. *Science*, *306*, 462–466. https://www.science.org

Qi, H., & Sun, D. (2010). An augmented lagrangian dual approach for the h-weighted nearest correlation matrix problem. *IMA Journal of Numerical Analysis*, *31*, 491–511. https://doi.org/10.1093/imanum/drp031

Renooij, S. (2001). Probability elicitation for belief networks: Issues to consider. *Knowledge Engineering Review*, *16*, 255–269. https://doi.org/10.1017/S0269888901000145

Renooij, S., & Witteman, C. (1999). Talking probabilities: Communicating probabilistic information with words and numbers. *International Journal of Approximate Reasoning*, *22*, 169–194.

Robinson, R. W. (1976). Counting unlabeled acyclic digraphs.

Røed, W., Mosleh, A., Vinnem, J. E., & Aven, T. (2009). On the use of the hybrid causal logic method in offshore risk analysis. *Reliability Engineering and System Safety*, *94*, 445–455. https://doi.org/10.1016/j.ress.2008.04.003

Röhrbein, F., Eggert, J., & Körner, E. (2009). Child-friendly divorcing: Incremental hierarchy learning in bayesian networks. *Proceedings of International Joint Conference on Neural Networks*, 2711–2716.

Rowe, G., & Wright, G. (2001). Expert opinions in forecasting: The role of the delphi technique. *Principles of forecasting: A handbook for researchers and practitioners*, 125–144.

Sakib, N., Hossain, N. U. I., Nur, F., Talluri, S., Jaradat, R., & Lawrence, J. M. (2021). An assessment of probabilistic disaster in the oil and gas supply chain leveraging bayesian belief network. *International Journal of Production Economics*, *235*. https://doi.org/10.1016/j.ijpe.2021.108107

Schoemaker, P. J. H., & Tetlock, P. E. (2016). Superforecasting: How to upgrade your company's judgment.

Scutari, M. (2010). Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, *35*, 1–22. https://doi.org/10.18637/jss.v035.i03

Si, S. L., You, X. Y., Liu, H. C., & Zhang, P. (2018). Dematel technique: A systematic review of the state-of-the-art literature on methodologies and applications. https://doi.org/10.1155/2018/3696457

Skjong, R., & Wentworth, B. H. (2001). Expert judgment and risk perception. *International Offshore and Polar Engineering Conference*, 537–544.

Soll, J. B., & Klayman, J. (2004). Overconfidence in interval estimates. *Journal of Experimental Psychology: Learning Memory and Cognition*, *30*, 299–314. https://doi.org/10.1037/0278-7393.30.2.299

Speirs-Bridge, A., Fidler, F., McBride, M., Flander, L., Cumming, G., & Burgman, M. (2010). Reducing overconfidence in the interval judgments of experts. *Risk Analysis*, *30*, 512–523. https://doi.org/10.1111/j.1539-6924.2009.01337.x

Surowiecki, J. (2004). *The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations*. Doubleday, Anchor.

Tetlock, P. E., & Gardner, D. (2016). *Superforecasting: The art and science of prediction*.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *Science*, *185*, 1124–1131. https://www.science.org

Wallsten, T. S., Budescu, D. V., Rapoport, A., Zwick, R., & Forsyth, B. (1986). Measuring the vague meanings of probability terms. *Journal of Experimental Psychology: General*, *115*, 348–365.

Wisse, B. W., Gosliga, S. P. V., Elst, N. P. V., & Barros, A. I. (2008). Relieving the elicitation burden of bayesian belief networks. *BMA*.

Yanore, L., Sok, J., & Lansink, A. O. (2023). Do dutch farmers invest in expansion despite increased policy uncertainty? a participatory bayesian network approach. *Agribusiness*. https://doi.org/10.1002/agr.21834

Zagorecki, A., & Druzdzel, M. (2004). An empirical study of probability elicitation under noisy-or assumption. *Flairs conference*, 880–885. www.aaai.org

Zagorecki, A., & Druzdzel, M. J. (2013). Knowledge engineering for bayesian networks: How common are noisy-max distributions in practice'. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, *43*, 186–195. https://doi.org/10.1109/TSMCA.2012.2189880

Zio, E., Mustafayeva, M., & Montanaro, A. (2022). A bayesian belief network model for the risk assessment and management of premature screen-out during hydraulic fracturing. *Reliability Engineering and System Safety*, *218*. https://doi.org/10.1016/j.ress.2021.108094

A

OTHER CPT CONSTRUCTION METHODS

This Appendix contains overviews of CPT construction methods that are not included in the applications of this thesis. These methods will be described in less detail than the previous, but the main method will still be illustrated.

A.1. STATIC/DYNAMIC RANKED NODES METHOD

In this version, two large limitations of the two previously discussed versions of RNM are addressed (Laitila & Virtanen, 2022). This new version allows for the parent nodes and child node to have different numbers of states, and for this version the discretization is no longer fixed throughout the parameterization process. The goal of the new approach is to make the construction of CPTs possible for arbitrary discretizations of continuous nodes. In addition, the Dynamic discretization approach also allows for point-valued evidence to be entered into the network.

In the original version, Equation (4.6) is used to calculate values for the CPT. This equation is based on a regression model. Let $\mathcal{X}_1,...,\mathcal{X}_n$ be continuous random variables on the interval $[0,1]$, the child node random variable $\mathcal{X}_C$ depends on these according to the following regression model:

$$\mathcal{X}_C = f(\mathcal{X}_1,...,\mathcal{X}_n,\mathbf{w}) + e, \quad e \sim N(0,\sigma^2), \tag{A.1}$$

where $\mathbf{w}$ and σ are elicited parameters, and f is the chosen weight function. By noting that $X_i = x_i$ is equivalent to $\mathcal{X}_i \in [a_i,b_i] = [h_i(A_i),h_i(B_i)]$, the following equivalence relation can be found:

$$\begin{aligned} &\mathbb{P}(X_C = x_C \mid X_1 = x_1,...,X_n = x_n) = \mathbb{P}(X_C = [A_C,B_C] \mid X_1 = [A_1,B_1],...,X_n = [A_n,B_n]) \\ &= \mathbb{P}(\mathcal{X}_C \in [h_C(A_C),h_C(B_C)] \mid \mathcal{X}_1 \in [h_1(A_1),h_1(B_1)],...,\mathcal{X}_n \in [h_n(A_n),h_n(B_n)]). \end{aligned} \tag{A.2}$$

This relation forms the basis for constructing a CPT of X_C with arbitrary discretizations of all nodes in the network X_i, $i \in \{1,...,n,C\}$. To construct a CPT, it is necessary to know the piecewise linear mappings h_i and the RNM parameters. The h_i is determined in the beginning when RNM-compatible discretizations are generated, the RNM parameters are determined afterwards.

Both the static and dynamic discretization approaches consist of six steps, which can be seen in Figure A.1. The first four steps are the same for both approaches, starting with the addition of an auxiliary node Y. This node is placed between the parent nodes and the child node and is needed for the construction of the CPT. In the next step, an initial discretization is made for nodes $X_1,...,X_n,X_C$ which are RNM-compatible, in the same way as was presented in Section 4.4.2. The Y node receives states that are state intervals on the $[0,1]$ interval of equal widths, which are associated with the states of the child node X_C. For each state discretization interval $[A_C,B_C]$, node Y has a corresponding state interval of $[h_C(A_C),h_C(B_C)]$. The piecewise linear mappings h_i remain the same throughout the application of the method.

The third step is optional. The discretizations of the nodes may now be changed for the elicitation of the RNM parameters. The intervals can be of varying width, and each node is allowed to have a different number of states. Note that, if the child node is rediscretized, the Y node must also be rediscretized accordingly by using the set h_C mappings defined in step two.

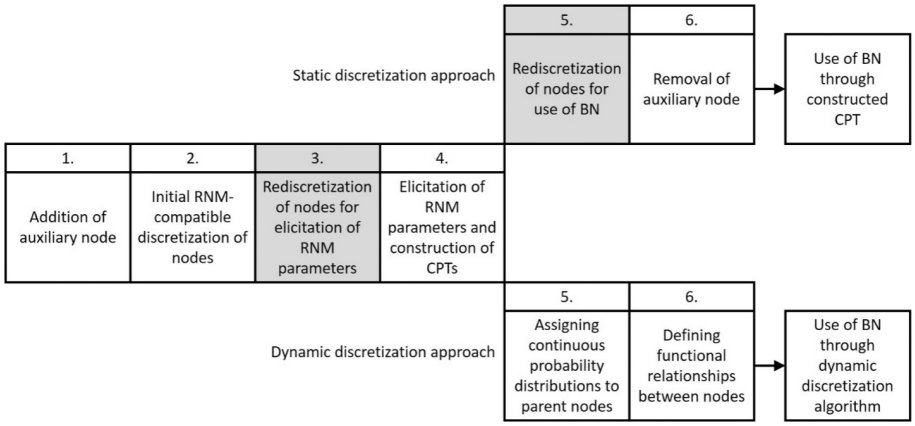


Figure A.1: Steps of the static discretization approach and the dynamic discretization approach, as in Laitila and Virtanen, 2022, where optional steps are colored grey.

The fourth step is to use expert elicitation to determine a weight function and corresponding parameters $\mathbf{w}, \sigma^2$. The method requires experts to give mode assessments for the child node in two scenarios: where each parent node is in the best state, and where each parent node is in the worst state. The variance parameter is still determined by trial and error, to fit the experts' views.

For the static approach, the optional fifth step supplies an extra opportunity to change the node discretizations, like in step three. At that point, the probabilistic relationship between X_C and $X_1, ..., X_n$ is already set. In the final step, the auxiliary node Y is removed to retrieve to original BN again. The CPT of Y is taken to be the CPT for the child node X_C.

For the dynamic approach, in the fifth step, continuous probability distributions are assigned to the parent nodes. The distributions may be estimated from data, or chosen by experts, and can be normal distributions or (piecewise) uniform distributions. Then, in the sixth step, functional relationships are defined between nodes. The relationship between X_C and Y is the inverse of the piecewise linear mapping h_C:

$$X_C = h_C^{-1}(Y) \tag{A.3}$$

The relationship between Y and the parent nodes $X_1, ..., X_n$ is defined by the truncated normal distribution on the interval $[0,1]$:

$$Y \sim TNorm(\mu, \sigma^2, 0, 1), \quad \mu = f(h_1(X_1), ..., h_n(X_n); \mathbf{w}), \tag{A.4}$$

where $TNORM$ is the truncated normal distribution.

After this step, the dynamic discretization algorithm can be applied to use the BN. Iteratively, new evidence is entered into the BN, which updates the discretizations of the nodes accordingly. The CPT of the child node is calculated based on the current discretizations and functional relationships. Then, for each node, the evidence is used to compute discrete marginal probability distributions, which are used to compute entropy error values over the discretization intervals. For each node, the interval with the largest entropy error is split into two. Simultaneously, consecutive intervals with entropy error zero are merged. The iterative process is stopped when a stopping criterion is met, such as a predetermined number of iterations, or some convergence threshold.

For the dynamic approach, the auxiliary node Y remains part of the BN, but in a purely computational role. This way, also during the use of the BN, the discretizations can still be updated. The dynamic approach is especially useful when accurate statistics are needed since the dynamic approach can produce smaller discretization intervals.

The Static and Dynamic approaches once again ask experts for their mode assessments in $2n$ scenarios, as well as an initial discretization of the nodes. For the static approach, there are two additional optional rediscretizations, for which each node state is reconsidered, resulting in $2\sum_{i=1}^{n} s_i$ extra considerations. For the dy-

namic approach, there is only one opportunity for rediscretization, but there also need to be chosen continuous probability distributions. So, in total between $(n+1)\cdot N_{iter}+2n+s+2$ and $(n+1)\cdot N_{iter}+2n+s+2+2\sum_{i=1}^{n} s_i$ values need to be chosen.

A.2. ELICITATION BBN

The next method that is discussed is the Elicitation BBN method (EBBN) (Wisse et al., 2008). The method is limited to BNs with ranked child nodes, where all parent nodes are ranked sucht that they have a poisitive influence on the child node. It is first defined what a positive (or negative) dominant node is, then it is set out what parameters need to be elicited, after which the CPT construction method is described.

A parent node $X_k \in pa(X_C)$ can be positive (or negative) dominant, if a positive dominant node is in the most positive state, the child node has the same probabilities for the states as when all nodes are in their most positive state. A negative dominant node forces the child node state probabilities according to all parent nodes being in the least favorable states.

The assessments that are required from the experts, can be elicited in the following steps:

(i) For each child node state x_C, the combination of parent node states $a_{x_C} = \{X_1 = x_1, ..., X_n = x_n\}$ that maximizes the probability of the child node being in that state $\mathbb{P}(X_C = x_C \mid a_{x_C})$, as well as the probabilities $\mathbb{P}(X_C \mid a_{x_C})$.

(ii) For all parent nodes, assess the probabilities of the child node being in either the best or the worst state, given that the considered parent node is in its most favorable state while the other parent nodes are in their least favorable states, denoted by a_{neg,k_+}.

(iii) For each parent node, determine if they are a positive/negative dominant node or not.

Consider a general BN fragment with n parent nodes, s_i parent node states, and s_C child node states. Also assume that all nodes in the network are ranked, such as for RNM and InterBeta. The first step requires s_C assessments for each child state: s_C^2. The second step requires $2n$ assessments, and finally, each parent node needs to be considered. So in total $2n + s_C + s_C^2$ assessments are needed.

Next, individual and joint influence factors $\mathcal{I}$ are determined. The factor $0 \leq \mathcal{I} \leq 1$ effectively orders all combinations of parent node states, where, $\mathcal{I}(a_{neg}) = 0$ when all parent states are in their most negative state. Similarly, $\mathcal{I}(a_{pos}) = 1$. The individual influence factor of a state x_i^k of parent node X_i is defined as:

$$\mathcal{I}_i(x_i^k) := \frac{s_i - \text{rank}(x_i^k)}{s_i - 1}.$$

The joint influence factor for a combination of parent node states $a = \{X_1 = x_1, ..., X_n = x_n\}$ is given by:

$$\mathcal{I}_{joint}(a) := \frac{\sum_{i:X_i \in pa(X_C)} \mathcal{I}_i(x_i^k) \cdot (\text{rank}(x_i^k) - 1)}{\sum_{i:X_i \in pa(X_C)} (s_i - 1)}.$$

The joint influence factors are then used to create piecewise linear mappings $f_{x_C} : [0,1] \to [0,1]$ through the points $\left(\mathcal{I}_{joint}(a_{x_C}), \mathbb{P}(X_C = x_C \mid a_{x_C})\right)$. These linear mappings can be used to determine $\mathbb{P}(X_C \mid a)$ via $\mathbb{P}(X_C \mid \mathcal{I}_{joint}(a))$. To also account for individual effects of parent states, the average of $\mathbb{P}_i(X_C \mid a)$ is taken over the interval $(\min\{\mathcal{I}_i(x_i^k), \mathcal{I}_{joint}(a)\}, \max\{\mathcal{I}_i(x_i^k), \mathcal{I}_{joint}(a)\})$ and denoted by $\overline{\mathbb{P}_i(X_C \mid a)}$. Then the weighted average is taken to find $\mathbb{P}(X_C \mid a)$:

$$\mathbb{P}(X_C \mid a) = \sum_{k:X_k \in pa(X_C)} w_k \cdot \overline{\mathbb{P}_k(X_C \mid a)},$$

A

where the weights are calculated by:

$$w_k = \frac{1}{2}\frac{\delta_k^+}{\sum_{l:X_l \in pa(X_C)} \delta_l^+} + \frac{1}{2}\frac{\delta_k^-}{\sum_{l:X_l \in pa(X_C)} \delta_l^-},$$
$$\text{where } \begin{cases} \delta_k^+ = \mathbb{P}(X_C = x_{c,max} \,|\, a_{neg,k_+}) - \mathbb{P}(X_C = x_{c,max} \,|\, a_{neg}), \\ \delta_k^- = \mathbb{P}(X_C = x_{c,min} \,|\, a_{neg}) - \mathbb{P}(X_C = x_{c,min} \,|\, a_{neg,k^+}). \end{cases}$$

To account for the dominant parent nodes, for any combination of parent node states a_d for which this node is in its dominant state, $\mathbb{P}(X_C \,|\, a_d)$ is set to $\mathbb{P}(X_C \,|\, a_{neg})$ or $\mathbb{P}(X_C \,|\, a_{pos})$.

A.3. WEIGHTED SUM ALGORITHM

This overview of the weighted sum algorithm is based on the paper by Das, 2004. The method is not limited to a set number of parent nodes, or node states. In addition, the node states do not necessarily need to be ranked, and nodes may have a different number of states from the others.

Instead of eliciting a predetermined set of questions about certain combinations of parent states, this method includes the experts in the process of determining what probabilities to elicit. Since different experts have different frames of knowledge, where some combinations of parent states make sense and others do not. The method assumes that experts can answer the questions about combinations of parent nodes that conform to their way of thinking.

Experts need to answer a maximum of $\sum_{i=1}^{n} s_i$ questions based on compatible parent configurations. Let parent node X_i be in state x_i^k, the state x_j^l of parent node X_j is compatible with x_i^k if according to the expert's mental model, x_j^l is most likely to coexist with x_i^k (Das, 2004). So the state of node X_j is chosen such that the conditional probability $\mathbb{P}(X_j = x_j^l \,|\, X_i = x_i^k)$ is maximal. The set $\{Comp(X_i = x_i^k)\}$ denotes the compatible parental configuration, which includes the states of the parent nodes that are compatible with x_i^k, and x_i^k itself. So there are $\sum_{i=1}^{n} s_i$ possible compatible parent configurations, however, in some cases these configurations are equal. This means that the number of compatible parent configurations is at least the maximum number of states of a parent node $\max s_i$, and at most $\prod_{i=1}^{n} s_i$.

To illustrate, consider the example BN of Section 2.3.5. When an expert is asked to find the compatible parental configuration for the node *Entertainment quality* being in state *High*, they might choose the *Availability of food/drinks* to be *High* and also the *Number of people* to be *High*. At the same time, when the expert is asked to construct a compatible parental configuration for the node *Availability of food/drinks* being in the state *High*, they also select the other two nodes to be in the *High* state. Thus, two parental configurations are found that are equal, which only has to be elicited once.

The experts are then asked to give the probability distribution over the child node states for the compatible parental configurations of all parent states. So a multinomial is elicited for each compatible parental configuration, resulting in a maximum of $s_c \sum_{i=1}^{n} s_i$ values to be elicited. In addition to these multinomials, also parent weights $w_1, ..., w_n$ are elicited from the expert. Thus, the elicitation process consists of three steps that are repeated for each parent node (and state):

(i) *Given that parent node i is in state x_i^j, what are the states of the other parent nodes that are most compatible?*

(ii) *Given the compatible parent configuration of node i being in state x_i^j, what is the distribution of the child node?*

(iii) *For each parent node i, what is the weight of this parent node?*

The following weighted sum algorithm can then be used to determine the full CPT:

$$\mathbb{P}(X_c = x_c \,|\, X_1 = x_1, ..., X_n = x_n) = \sum_{i=i}^{n} w_i \mathbb{P}(X_c = x_c \,|\, \{Comp(X_i = x_i)\}). \tag{A.5}$$

Thus, each value in the CPT is calculated using the compatible parent configurations. By taking the weighted average over the compatible parent configurations of all states in the particular scenario: $\{Comp(X_1 = x_1)\}$ $, ..., \{Comp(X_n = x_n)\}$.

A.4. CAIN'S METHOD

Another method that uses interpolation is also known as Cain's method (Cain, 2001). An overview will be given of the method for the general case of n discrete parent nodes with arbitrary numbers of states, and a discrete child node with an arbitrary number of states. The method applies to BNs with node states that can be ranked, such that each node has a state that has the most 'positive' effect on the child node and a state that has the most 'negative' influence on the child node.

To start, the child node distribution for several CPT rows needs to be elicited, depending on the number of parent nodes and parent node states. These elicitations form a so-called Elicited Probability Table (EPT). The EPT consists of the two cases where all parent nodes are in their most positive state, or all parent nodes are in their most negative state. In addition, the cases where all parent nodes are in their most positive state except for one parent node are elicited, this other node is in an arbitrary other state. Thus when the parent nodes X_i each have s_i states, a total of $2+\sum_{i=1}^{n}(s_i-1)$ multinomials for the child node need to be elicited.

Similar to some of the elicitation questions of the previously described CPT construction methods, the following questions can be used during the elicitation process:

- *Given that parent node 1 is in state $x_1^{\uparrow}$ and ... and parent node n is in state $x_n^{\uparrow}$, what is the child node distribution?*
- *Given that parent node 1 is in state $x_1^{\downarrow}$ and ... and parent node n is in state $x_n^{\downarrow}$, what is the child node distribution?*
- *Given that parent node i is in state $x_i^{\uparrow}$ for all $i \neq j$, and parent node j is in state x_j^{k}, what is the child node distribution?*

Where $x_i^{\uparrow}$ denotes the most positive state of node X_i and $x_i^{\downarrow}$ the most negative.

In the next step, the experts' assessments are averaged. This may be done by assigning a weight to each expert, but there are no strict guidelines for this. Then, for $n-1$ parent nodes an Interpolation Factor (IF) needs to be calculated, which calculates the change of the probability of the child node when a parent node switches state.

When parent X_i changes from state x_i^a to state x_i^b the following IF is needed, for the child being in state x_c:

$$IF_{i:a-b,X_c=x_c} = \frac{\mathbb{P}(X_c = x_c \,|\, X_j = x_j^{\uparrow}\,\forall\, j \neq i, X_i = x_i^b) - \mathbb{P}(X_c = x_c \,|\, X_j = x_j^{\downarrow}\,\forall\, j)}{\mathbb{P}(X_c = x_c \,|\, X_j = x_j^{\uparrow}\,\forall\, j) - \mathbb{P}(X_c = x_c \,|\, X_j = x_j^{\downarrow}\,\forall\, j)}. \tag{A.6}$$

The EPT can then be extended to construct the CPT for the BN, by calculating values for the missing rows. The rows are calculated one by one by using the row that is most similar to them and then applying the IF. So, for example:

$$\begin{aligned}
&\mathbb{P}(X_c = x_c \,|\, X_1 = x_1^{\uparrow}, ..., X_{n-2} = x_{n-2}^{\uparrow}, X_{n-1} = x_{n-1}^{\uparrow}, X_n = x_n^a) \\
&\quad = \left(\mathbb{P}(X_c = x_c \,|\, X_1 = x_1^{\uparrow}, ..., X_{n-1}^a, X_n = x_n^a) - \mathbb{P}(X_c = x_c \,|\, X_1 = x_1^{\downarrow}, ..., X_n = x_n^{\downarrow})\right) \cdot IF_{n-1:\uparrow-a,X_c=x_c} \\
&\qquad + \mathbb{P}(X_c = x_c \,|\, X_1 = x_1^{\downarrow}, ..., X_n = x_n^{\downarrow}),
\end{aligned}$$

where node X_n goes from the most positive state, to state a.

A.5. RØED'S METHOD

The next method was presented by Røed et al., 2009, and constructs CPTs using the distance between the parent node states and the child state. The assumption is made that a probability assigned to a child state that is much different from the parent states should be smaller than the probability assigned to a child state similar to the parent states. For instance, when ranked nodes are considered, if all parent states are in their best state, the probability that the child state is also in their best state should be the largest. The probability that the child is in their worst state should then be the smallest.

To start, weights are calculated for each parent. These can be determined by using indirect expert judgment. Instead of directly asking experts to give weights to the parent nodes, the relative change of the expected value of the child node $\mathbb{E}[X_c]$ is elicited when one parent node is changed from the best state to the

worst, while the other parents remain in a middle state. This question is repeated for all parent nodes. The found values are normalized to find w_i, where $\sum_{i=1}^{n} w_i = 1$.

Next, a distance measure is calculated:

$$Z_j = \sum_{i=1}^{n} |Z_{ij}| \, w_i,$$

where Z_{ij} is the 'distance' between the state of parent i and the child state j. In the default case, when the nodes are ranked, the distances are taken to be the difference between the rank of the child node state and the parent node state. The absolute value in the measure ensures that the difference is equal in both directions, if necessary it is also possible to alter this measure such that positive and negative distances are different. In that case, experts could be asked to assess the distances between the states of all parent nodes and the child node states.

A probability distribution can then be calculated by:

$$P_j = \frac{e^{-R \cdot Z_j}}{\sum_{j=a}^{f} e^{-R \cdot Z_j}},$$

where $P_j \in [0,1]$ and R is an index that determines the spread of the probability mass. A high R ensures that the probability that the child is in a state distant from its parent states is small, $R = 0$ gives a uniform distribution. One method for eliciting the parameter R from experts suggested by Røed et al., 2009, focuses on the relative difference between a perfect and an average situation.

So, in total, n parent weights, the index R, and perhaps the distances between parent states and child states need to be elicited from experts. Let x_i^- be the middle/average state of a node i, then the following questions are part of the elicitation process:

(i) The parent weights can be elicited directly, or by using the following two questions:

  (a) *Given that parent node i is in state x_i^- for all $i \neq j$, and parent node j is in state $x_n^{\uparrow}$, in what state is the mode of the child node?*

  (b) *Given that parent node i is in state x_i^- for all $i \neq j$, and parent node j is in state $x_n^{\downarrow}$, in what state is the mode of the child node?*

(ii) Indirect elicitation of R: *Assume that all parent nodes are in their most positive state, so node i is in state $x_i^{\uparrow}$ for all i. How much higher would the probability be that the child node is in its highest state $x_C^{\uparrow}$ than in an average state x_C^-?*

(iii) If the distances Z_{ij} are not as wanted: *For each combination of a parent node state x_i^j and a child node state x_C^l, what is the distance between x_i^j and x_C^l?*

A.6. ACE

One software that is developed to help initialize CPTs is the Application for Conditional Probability Elicitation (ACE) (Hassall et al., 2019). The method that is described in this section is incorporated into the ACE software, for simplicity it will be referred to as the ACE method. The method helps to initialize CPTs efficiently, which can then be further refined to fit the experts' views. So, unlike the other methods discussed, this ACE does not aim to generate CPTs that can be used immediately.

The first step in the method is to elicit relative importance weights w_i of the parent nodes, from the experts. Next, the type of relationship between the child and each parent is defined. This can be either positive, negative, or something else. If a parent node moving from a lower state to a higher state, according to a previously set ordering of states, increases the probability of the child node being in a higher state as well, then this relation is called positive. The converse is a negative relation. If neither a positive nor negative relation is found, the relation between states can be assigned by hand, by defining a new order.

In short, the following questions can be used:

- *For each parent node i, what is the relative importance weight of this parent node?*
- *What is the type of relationship that exists between parent node i and the child node? It can be either positive, negative, or something else.*

After finishing the elicitation process, for each state j of a parent i, a score can be calculated:

$$P_{ij} = \begin{cases} \frac{j-1}{s_i-1}, & \text{If parent } i \text{ has a positive relationship with the child node,} \\ \frac{m_i-j}{s_i-1}, & \text{If parent } i \text{ has a negative relationship with the child node,} \\ \frac{ord(j)-1}{s_i-1}, & \text{If parent } i \text{ has another relationship with the child node,} \end{cases}$$

where s_i is the number of states of parent i, and $ord(j)$ is the index of state j. For each combination of parent states k, an overall score is calculated:

$$Score_k = \frac{\sum_{i=1}^{n} w_i P_{ik}}{\sum_{i=1}^{n} w_i},$$

where P_{ik} is the parent score associated with parent state combination k. The values in the CPT can then finally be calculated by considering the trapezium between $1-Score_k$, $Score_k$, and the x-axis. This trapezium is then divided into m_c sub-trapezia of equal width, which is the number of child states. The area of the sub-trapezia is equal to half of the probability that the child is in that state, see Figure A.2. The constructed

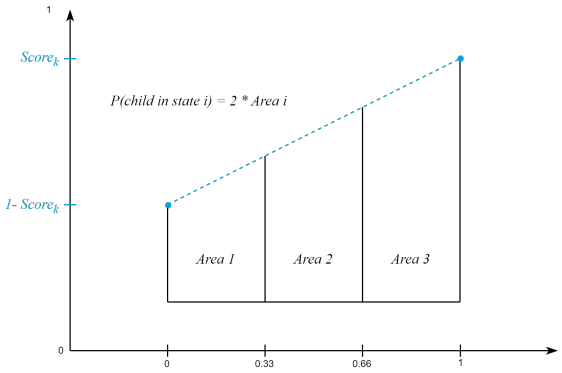


Figure A.2: Trapezium which determines the CPT values for the ACE method.

CPT can then be used as a basis which can be edited such that it matches the experts' beliefs. The method assumes all states to be on an equally spaced linear scale. Due to the trapezium mapping, if the child has an odd number of states s_c, the middle state will always have probability $1/s_c$.

A.7. THE LIKELIHOOD METHOD

The final method discussed is the Likelihood method. When the mental models of experts are assumed to be that of parent states moving the child node away from some typical distribution, the likelihood method is called for (Kemp-Benedict, 2008). Then, the child node's distribution is changed only when certain parent node states occur. The main idea for the method is Bayes' rule:

$$\mathbb{P}(X_C = x_C \,|\, X_1 = x_1, ..., X_n = x_n) \propto L(X_C = x_C \,|\, X_1 = x_1, ..., X_n = x_n)\mathbb{P}(X_C = x_C),$$

where $L(X_C = x_C \,|\, X_1 = x_1, ..., X_n = x_n)$ is the likelihood of the child node X_C being in state x_C, given that the parent nodes are in states $x_1, ..., x_n$. The prior probability for X_C is $\mathbb{P}(X_C = x_C)$ and is given by the typical distribution T_{X_C}, leading to:

$$\mathbb{P}(X_C = x_C \,|\, X_1 = x_1, ..., X_n = x_n) \propto L(X_C = x_C \,|\, X_1 = x_1, ..., X_n = x_n)T_{X_C}. \tag{A.7}$$

The typical distribution T_{X_C} describes a "typical" state of affairs defined by the user. For instance, for a discrete node, it can be: [low, middle, high]:$[1/3, 1/3, 1/3]$ or for a child node with five states: [very low, low, middle, high, very high]:$[1/20, 1/5, 1/2, 1/5, 1/20]$. The typical distribution may also be continuous, but in this description, it is assumed to be discrete.

Equation (A.7) may be interpreted as: when information about the parent nodes is absent, the child node is proportional to the typical distribution, otherwise, the likelihood moves the child node probability away

from the typical distribution. Conventionally, the log-likelihood is taken, where it is assumed that this can be expressed as the sum of independent terms:

$$\log_b L(X_C = x_C \mid X_1 = x_1, ..., X_n = x_n) = \sum_{i=1}^{n} \alpha_{X_c}^{(c)} \alpha_{X_i}^{(i)}$$

$$\Leftrightarrow L(X_C = x_C \mid X_1 = x_1, ..., X_n = x_n) = b^{\alpha_{X_C}^{(C)} \alpha_{X_1}^{(1)} + ... + \alpha_{X_C}^{(C)} \alpha_{X_n}^{(n)}}.$$

Each product $\alpha_{X_C}^{(C)} \alpha_{X_i}^{(i)}$ represents the influence of one parent node. The parameter $\alpha_{X_C}^{(C)}$ represents the child node being in state x_C and $\alpha_{X_i}^{(i)}$ represents parent node X_i being in state x_i. If the product is positive, it increases the likelihood of $X_C = x_C$ occurring when $X_i = x_i$ is present, a negative value decreases this likelihood. The base b is chosen such that the αs are of convenient magnitude.

The typical distribution T_{X_C} and the base b can be elicited from experts. However, the main task for the experts is to assess the values α for each state of each parent node and child node. Thus, a total of $s_C + \sum_{i=1}^{n} s_i$ parameters need to be elicited:

- a typical distribution T_{X_C},
- the base b,
- a weighting factor for each state of the child node $\alpha_{X_C}^{(C)}$,
- a weighting factor for each state of the parent nodes $\alpha_{X_n}^{(n)}$.

More guidelines on how to elicit these values are given in Knochenhauer et al., 2013.

B

SUPPORTING THEOREMS AND PROOFS

This appendix contains proofs and methodological results to support ideas used in this thesis. The first section contains the proof that the KL-divergence of a joint probability distribution depends on the KL-divergences of the child node dependent on its parent nodes. Therefore, in this thesis, the KL-divergence between two CPTs is taken as the mean of the row-by-row KL-divergence.

In the second section, a theoretical counterexample is given to the hypothesis that the variance curve, resulting from interpolating the best and worst α, β, is concave. It is then methodologically shown that, although the variance curve is not always concave, this does not have large implications in practice. As for simulated CPTs, the variance curve from α, β interpolation lies, almost always, above the linearly interpolated variance.

B.1. KULLBACK-LEIBLER DIVERGENCE OF A JOINT PROBABILITY DISTRIBUTION

Lemma B.1.1. *Let X_C be a child node with parent nodes $X_1, ..., X_n$. Then the KL-divergence between joint distributions $P(X_C, X_1, ..., X_n)$ and $Q(X_C, X_1, ..., X_n)$ depends on the KL-divergence between the conditional distributions $P(X_C|X_1, ..., X_n)$ and $Q(X_C|X_1, ..., X_n)$:*

$$D_{KL}(P(X_C, X_1, ..., X_n) \| Q(X_C, X_1, ..., X_n)) = \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_1, ..., x_n) D_{KL}(P(X_C | X_1, ..., X_n) \| Q(X_C | X_1, ..., X_n)) + D_{KL}(P(X_1, ..., X_n) \| Q(X_1, ..., X_n)).$$

Proof. Let X_C be a child node with parent nodes $X_1, ..., X_n$, then the KL-divergence between the joint distributions $P(X_C, X_1, ..., X_n)$ and $Q(X_C, X_1, ..., X_n)$ can be decomposed:

$$\begin{aligned}
&D_{KL}(P(X_C, X_1, ..., X_n) \| Q(X_C, X_1, ..., X_n)) \\
&= \sum_{x_C \in X_C} \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_C, x_1, ..., x_n) \ln \frac{P(x_C, x_1, ..., x_n)}{Q(x_C, x_1, ..., x_n)} \\
&= \sum_{x_C \in X_C} \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_C | x_1, ..., x_n) P(x_1, ..., x_n) \ln \frac{P(x_C | x_1, ..., x_n) P(x_1, ..., x_n)}{Q(x_C | x_1, ..., x_n) P(x_1, ..., x_n)} \\
&= \sum_{x_C \in X_C} \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_C | x_1, ..., x_n) P(x_1, ..., x_n) \left(\ln \frac{P(x_C | x_1, ..., x_n)}{Q(x_C | x_1, ..., x_n)} + \ln \frac{P(x_1, ..., x_n)}{Q(x_1, ..., x_n)} \right) \\
&= \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_1, ..., x_n) \sum_{x_C \in X_C} P(x_C | x_1, ..., x_n) \ln \frac{P(x_C | x_1, ..., x_n)}{Q(x_C | x_1, ..., x_n)} \\
&\qquad + \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_1, ..., x_n) \ln \frac{P(x_1, ..., x_n)}{Q(x_1, ..., x_n)} \cdot \underbrace{\sum_{x_C \in X_C} P(x_C | x_1, ..., x_n)}_{=1} \\
&= \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_1, ..., x_n) D_{KL}(P(X_C | X_1, ..., X_n) \| Q(X_C | X_1, ..., X_n)) \\
&\qquad + D_{KL}(P(X_1, ..., X_n) \| Q(X_1, ..., X_n))
\end{aligned}$$

So the KL-divergence between joint distributions depends on the KL-divergence between the child node conditional on its parent nodes:

$$D_{KL}(P(X_C|X_1,...,X_n)\|Q(X_C|X_1,...,X_n)) = \sum_{x_C \in X_C} P(x_C|x_1,...,x_n) \ln \frac{P(x_C|x_1,...,x_n)}{Q(x_C|x_1,...,x_n)}$$

☺

Note that, if the joint probability distributions of the parent nodes are equal, $P(X_1,...,X_n) = Q(X_1,...,X_n)$, then $D_{KL}(P(X_1,...,X_n)\|Q(X_1,...,X_n)) = 0$. Furthermore, if the variables $X_1,...,X_n$ are assumed to be independent and have a discrete uniform distribution with each s_{X_i} $i = 1,...,n$ possible outcomes (or states). Then the decomposition can be simplified:

$$\begin{aligned}
D_{KL}(P(X_C,X_1,...,X_n)\|Q(X_C,X_1,...,X_n)) &= \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} P(x_1,...,x_n) D_{KL}(P(X_C|X_1,...,X_n)\|Q(X_C|X_1,...,X_n)) \\
&= \sum_{x_1 \in X_1} P(x_1) ... \sum_{x_n \in X_n} P(x_n) D_{KL}(P(X_C|X_1,...,X_n)\|Q(X_C|X_1,...,X_n)) \\
&= \sum_{x_1 \in X_1} \frac{1}{S_{X_1}} ... \sum_{x_n \in X_n} \frac{1}{S_{X_n}} D_{KL}(P(X_C|X_1,...,X_n)\|Q(X_C|X_1,...,X_n)) \\
&= \frac{1}{S_{X_1} \cdot ... \cdot S_{X_n}} \sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} D_{KL}(P(X_C|X_1,...,X_n)\|Q(X_C|X_1,...,X_n))
\end{aligned}$$

For the application of the KL-divergence to compare CPTs, we can note that the product of the number of states for each of the parent variable is equal to the number of rows in a CPT. So, if it is assumed that the parent nodes are independent and are specified by discrete uniform margins, for both the "true" and the "constructed" CPT, then the KL divergence between the joint probability of the true and constructed BN is equal to the average of the KL-divergence between the individual rows of the true and constructed CPT.

B.2. VARIANCE COMPARISON FOR α, β AND MEAN/VARIANCE INTERPOLATION

From the analysis of the beta parameters fit to each row of the fully elicited CPTs, the following conjecture was made:

Conjecture B.2.1. *Suppose the shape and scale parameters of the Beta distribution ($\alpha > 0, \beta > 0$) have a linear relationship, such that $\beta = c\alpha + d$ where $c, d \in \mathbb{R}$. Then there exists a higher-order relationship between the mean and variance of the distribution, which is concave.*

It was found that the conjecture does not hold for all $c, d \in \mathbb{R}$. The method, and one of the found counterexamples are presented here:

Method One can write the mean μ and variance σ^2 of the Beta distribution in terms of the α and β:

$$\mu = \frac{\alpha}{\alpha + \beta},$$
$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

Let the $\alpha, \beta \in \mathbb{R}^+ = \{x \in \mathbb{R} | x > 0\}$ parameters of the Beta distribution be such that $\beta = c\alpha + d$, where the constants $c, d \in \mathbb{R}$. Then using the linear relationship between the α and β, the mean can be written in terms of the α and coefficients c, d. The α can then be isolated to substitute in the variance.

$$\mu = \frac{\alpha}{\alpha+\beta} = \frac{\alpha}{\alpha + c\alpha + d} \quad \Rightarrow \quad \alpha = \frac{\mu d}{1 - \mu - \mu c}.$$

The found α is substituted, the variance can be rewritten in terms of the mean μ and coefficients c, d:

$$\sigma^2 = \mu(\mu - 1)\frac{\mu(1+c) - 1}{1 + d - \mu(1+c)}.$$

Which has a second derivative of:

$$\frac{\mathrm{d}^2}{\mathrm{d}\mu^2}\sigma^2 = \frac{-2(c+1)^3\mu^3 + 6(c+1)^2(d+1)\mu^2 - 6(d+1)^2(c+1)\mu + 2(d+1)(cd+2d+1)}{((c+1)\mu - d - 1)^3}.$$

To find the inflection points of the function, the second derivative is set to zero:

$$\frac{\mathrm{d}^2}{\mathrm{d}\mu^2}\sigma^2 = 0 \quad \Rightarrow \quad \tilde{\mu} = \frac{(d(d+1)(c-d))^{\frac{1}{3}} + d + 1}{c+1} \quad \text{or} \quad \tilde{\mu} = \frac{(-\frac{1}{2} \pm \frac{i\sqrt{3}}{2})(d(d+1)(c-d))^{\frac{1}{3}} + d + 1}{c+1}. \tag{B.1}$$

Four cases can be distinguished between: $c \geq 0 \wedge d \geq 0$, $c \geq 0 \wedge d < 0$, $c < 0 \wedge d \geq 0$, $c < 0 \wedge d < 0$. The fourth case, where both c and d are negative, would imply $\alpha, \beta < 0$ and is therefore not considered. To prove that the variance is concave with respect to the mean, it has to be shown that the inflection points lay outside the range of μ. For the first case a counter example will be shown.

CASE 1 Let $c, d > 0$, first the range of the mean μ is determined. Since $\alpha, \beta > 0$ we have that $\mu > 0$, additionally $\alpha + \beta > \alpha$ implies that $\mu < 1$. The range for the mean can be determined by finding the limit of the mean:

$$\lim_{\alpha\to\infty} \mu = \lim_{\alpha\to\infty} \frac{\alpha}{\alpha + c\alpha + d} = \lim_{\alpha\to\infty} \frac{1}{1 + c + \frac{d}{\alpha}} = \frac{1}{1+c}.$$

Thus, either $\mu \in (0, \frac{1}{1+c})$ or $\mu \in (\frac{1}{1+c}, 1)$ depending on the sign of d and c. In this case we find that:

$$d > 0, c > 0 \quad \Rightarrow \quad \mu = \frac{\alpha}{\alpha + c\alpha + d} < \frac{\alpha}{\alpha + c\alpha} = \frac{1}{1+c} \quad \Rightarrow \quad \mu \in \left(0, \frac{1}{1+c}\right).$$

Using the inflection points found in (B.1) we can distinguish between three cases:

$$\begin{aligned}
c = d \quad &\Rightarrow \quad \tilde{\mu} = \frac{(d(d+1)(c-d))^{\frac{1}{3}} + d + 1}{c+1} = \frac{c+1}{c+1} = 1 > \frac{1}{1+c}, \\
c > d \quad &\Rightarrow \quad \tilde{\mu} = \frac{\overbrace{(d(d+1)(c-d))^{\frac{1}{3}}}^{>0} + d + 1}{c+1} > \frac{1}{1+c}, \\
c < d \quad &\Rightarrow \quad \tilde{\mu} = \frac{-(d(d+1)(d-c))^{\frac{1}{3}} + d + 1}{c+1}, \\
&\tilde{\mu} > \frac{1}{1+c} \quad \Longleftrightarrow \quad d > (d(d+1)(d-c))^{\frac{1}{3}}, \\
&\Longleftrightarrow \quad d^3 > d(d+1)(d-c), \\
&\Longleftrightarrow \quad d^2 > d^2 + d - cd - c, \\
&\Longleftrightarrow \quad d(1-c) < c, \\
&\Longleftrightarrow \quad \begin{cases} d > \frac{c}{1-c} & \text{and} \quad c > 1, \\ d < \frac{c}{1-c} & \text{and} \quad c < 1. \end{cases}
\end{aligned}$$

Note that, for the first case, if $c > 1$ we find that $\frac{c}{1-c} < 0$, thus $d > \frac{c}{1-c}$ holds in this case. So, whenever $d > c > 1$, there is no inflection point inside the interval $\mu \in (0, \frac{1}{1+c})$. However, when $c \in (0,1)$ and $d > \max(c, \frac{c}{1-c})$ there is a inflection point inside the range of μ, which means that the variance cannot be concave on the entire interval.

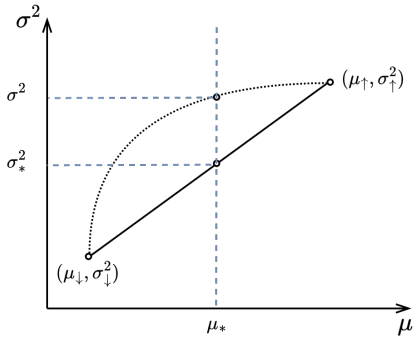
To show that the variance is concave for at least part of the interval, the sign of the second derivative of the variance function is assessed. For one test case in the center of the interval $\mu = \frac{1}{2(1+c)}$, the second derivative of σ^2 with respect to μ is evaluated:

$$\frac{\mathrm{d}^2}{\mathrm{d}\mu^2}\sigma^2\Big|_{\mu=\frac{1}{2(1+c)}} = \frac{-2 + (-16c-8)d^2 + (-16c-12)d}{(2d+1)^3} < 0.$$

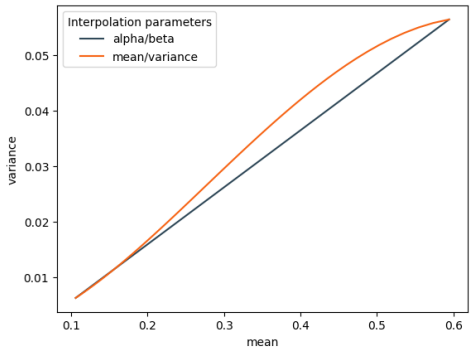
So, at the middle of the interval the function is concave. Thus, if $d > c > 1$, or if $c \geq d$, or if $c \in (0,1)$ and $c < d < \frac{c}{1-c}$, the variance is a concave function with respect to the mean on the entire interval $\mu \in \left(0, \frac{1}{1+c}\right)$.

For the case that $c < 0, d > 0$ and the case that $c > 0, d < 0$ a similar approach can be used, which will result in more counterexamples. The result is that Conjecture B.2.1 does not hold for all $c, d \in \mathbb{R}$.

Simulated CPTs In practice, the result that the variance is not necessarily concave on the whole range of μ does not regularly appear. To test the hypothesis, the simulated CPTs of Chapter 8 are used. For each simulated CPT, a beta distribution is fit to the best and worst row. Then, the α/β and mean/variance parameters are interpolated between and compared for 100 values of μ_*, see Figure B.1a. The μ_* divide the interval $[\mu_\downarrow, \mu_\uparrow]$ in sub-intervals of equal widths. The variance on the interpolation curve of α, β (σ^2) is compared against the linearly interpolated variance (σ^2_*).



(a) Schematic overview of the variance comparison between the obtained curve when interpolating the α, β (dotted) or the mean, variance (solid).



(b) Case for which the variance curve calculated from interpolated α, β is not concave between the best and worst parameters, $(\alpha_{best}, \beta_{best}) = (1.94, 1.33)$ $(\alpha_{worst}, \beta_{worst}) = (1.51, 12.69)$.

Figure B.1: Supporting figures for methodologically showing the difference between the variance curves for α, β interpolation and mean/variance interpolation.

For all simulated CPTs included in the simulations of Section 8.5 the variance curves are compared. To avoid the influence of calculation errors when converting the α, β to the mean and variance, the difference needs to be at least 0.0001. In total 6000 CPTs were tested, for only one tested CPT it was found that $\sigma^2 - \sigma^2_* < -0.0001$, which is shown in Figure B.1b. Even in the case that the variance is not concave on the full interval between the best and worst mean, is the function concave on most of the interval.

Thus, in theory, the variance is not a concave function with respect to the mean but in practice, the variance curve from interpolated α/β parameters lies above the line between the best and worst mean/variance.

C

TABLES AND FIGURES

Table C.1: Results of KL-divergence and percentage of agreement performance for InterBeta, rounded to four decimals and one decimal respectively, the best performance results of each row are printed in boldface. This table contains the information shown in Figure C.12.

CPT	RNM		Functional Interpolation		
(KL-div / % agreement)	original	AutoRNM	normal	t-normal	beta
Polar bears Ice	0.3509 / 56.9	0.2833 / 54.2	0.2996 / 61.1	0.3769 / 70.8	0.2501 / 62.5
Polar bears Disturb	0.1205 / 60.5	0.1547 / 48.1	0.0913 / 80.2	0.1525 / 85.2	0.0912 / 80.2
Polar bears CumPop	0.2119 / 72.2	0.2324 / 72.2	0.2163 / 75.0	1.5414 / 55.6	0.2015 / 75.0
Polar bears AFBod	0.186 / 91.7	0.2327 / 97.2	0.3125 / 69.4	0.6553 / 50.0	0.1522 / 91.7
Polar bears SASur	0.3026 / 63.9	0.3204 / 72.2	0.4699 / 63.9	0.8802 / 61.1	0.1005 / 94.4
Polar bears AdSur	0.3113 / 63.9	0.3312 / 72.2	0.5344 / 63.9	0.8427 / 55.6	0.124 / 94.4
Polar bears OthMor	0.148 / 77.8	0.181 / 70.4	0.1712 / 74.1	0.2521 / 74.1	0.1061 / 85.2
Polar bears EvMort	0.1332 / 100.0	0.1518 / 100.0	0.1833 / 92.6	0.3768 / 66.7	0.1031 / 92.6
Polar bears TerrPry	0.22 / 100.0	0.2745 / 100.0	0.4597 / 62.5	0.9088 / 50.0	0.2109 / 100.0
Polar bears Recr	0.2418 / 100.0	0.2596 / 100.0	0.6192 / 75.0	0.7533 / 58.3	0.2718 / 91.7
Polar bears Mrn	0.138 / 50.0	0.1412 / 75.0	0.1911 / 75.0	0.4872 / 75.0	0.1008 / 91.7
Polar bears Hab	0.417 / 55.6	0.4197 / 55.6	0.178 / 77.8	0.9017 / 55.6	0.1474 / 77.8
Polar bears Terr	0.0923 / 91.7	0.0999 / 91.7	0.0846 / 91.7	0.6179 / 75.0	0.1288 / 91.7
Polar bears PrimPrey	0.1983 / 77.8	0.227 / 77.8	0.2511 / 100.0	0.2791 / 66.7	0.2115 / 77.8
Polar bears MrnPry	0.0862 / 88.9	0.1425 / 88.9	0.0594 / 100.0	0.5905 / 77.8	0.0688 / 88.9
Polar bears BioStr	0.1472 / 77.8	0.1633 / 77.8	0.2262 / 66.7	0.2582 / 66.7	0.1417 / 77.8
Food security EWDM	0.1843 / 66.7	0.1851 / 66.7	0.1043 / 91.7	0.0631 / 91.7	0.0171 / 91.7
Food security PWDM	0.1863 / 50.0	0.1752 / 58.3	0.1005 / 75.0	0.109 / 66.7	0.0242 / 66.7
Food security 1	0.188 / 66.7	0.1733 / 58.3	0.1221 / 75.0	0.1558 / 66.7	0.0237 / 66.7
Food security 2	0.2108 / 58.3	0.1848 / 41.7	0.1238 / 75.0	0.1073 / 66.7	0.0284 / 66.7
Food security 3	0.2005 / 50.0	0.1697 / 75.0	0.1299 / 66.7	0.1831 / 58.3	0.0816 / 58.3
Food security 4	0.1903 / 66.7	0.2223 / 58.3	0.0852 / 100.0	0.0758 / 75.0	0.0235 / 83.3
Food security 5	0.4594 / 41.7	0.4761 / 66.7	0.5775 / 66.7	0.1192 / 100.0	0.028 / 100.0
Pollinator abundance EWDM	0.0032 / 100.0	0.0031 / 100.0	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0
Pollinator abundance 1	0.01 / 100.0	0.0099 / 100.0	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0
Pollinator abundance 2	0.0094 / 100.0	0.0082 / 100.0	0.0 / 100.0	0.0 / 87.5	0.0 / 100.0
Pollinator abundance 3	0.0088 / 100.0	0.0083 / 100.0	0.0 / 100.0	0.0 / 87.5	0.0 / 100.0
Pollinator abundance 4	0.004 / 100.0	0.0027 / 100.0	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0
Pollinator abundance 5	0.0201 / 87.5	0.0161 / 100.0	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0
Pollinator abundance 6	0.0043 / 100.0	0.0054 / 100.0	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0
Pollinator abundance 7	0.0166 / 75.0	0.0153 / 87.5	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0
Pollinator abundance 8	0.0247 / 75.0	0.0238 / 75.0	0.0 / 100.0	0.0 / 87.5	0.0 / 100.0
Pollinator abundance 9	0.0219 / 87.5	0.0201 / 75.0	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0
Pollinator abundance 10	0.0086 / 100.0	0.0071 / 100.0	0.0 / 100.0	0.0 / 100.0	0.0 / 100.0

Table C.2: Results of KL-divergence and percentage of agreement performance for InterBeta, rounded to four decimals and one decimal respectively. This table contains the information shown in Figure C.12.

CPT	InterBeta			
(KL-div / % agreement)	best and worst	parent weights	state weights	row weights
Polar bears Ice	0.2284 / 70.8	0.1347 / 76.4	0.1158 / 73.6	0.0337 / 81.9
Polar bears Disturb	0.1885 / 75.3	0.1885 / 75.3	0.1853 / 72.8	0.1346 / 67.9
Polar bears CumPop	0.3701 / 63.9	0.3554 / 72.2	0.3407 / 77.8	0.2367 / 88.9
Polar bears AFBod	0.704 / 47.2	0.2047 / 97.2	0.1799 / 97.2	0.1324 / 97.2
Polar bears SASur	0.6735 / 58.3	0.2423 / 75.0	0.0792 / 97.2	0.0298 / 100.0
Polar bears AdSur	0.6897 / 58.3	0.201 / 80.6	0.0753 / 97.2	0.0256 / 100.0
Polar bears OthMor	0.1616 / 77.8	0.133 / 77.8	0.1188 / 77.8	0.0766 / 85.2
Polar bears EvMort	0.4037 / 55.6	0.1105 / 100.0	0.1096 / 100.0	0.084 / 92.6
Polar bears TerrPry	0.7762 / 62.5	0.2977 / 100.0	0.2896 / 100.0	0.258 / 100.0
Polar bears Recr	0.7226 / 58.3	0.2264 / 100.0	0.149 / 100.0	0.1165 / 91.7
Polar bears Mrn	0.2361 / 83.3	0.2015 / 83.3	0.1302 / 100.0	0.0545 / 100.0
Polar bears Hab	0.6701 / 55.6	0.3285 / 66.7	0.2724 / 77.8	0.1915 / 88.9
Polar bears Terr	0.1646 / 91.7	0.1394 / 91.7	0.0612 / 83.3	0.0478 / 100.0
Polar bears PrimPrey	0.2239 / 66.7	0.2021 / 100.0	0.2019 / 100.0	0.1528 / 88.9
Polar bears MrnPry	0.341 / 66.7	0.1533 / 88.9	0.1445 / 88.9	0.0707 / 100.0
Polar bears BioStr	0.1641 / 77.8	0.1641 / 77.8	0.1602 / 88.9	0.1063 / 77.8
Food security EWDM	0.1819 / 66.7	0.0288 / 91.7	0.0285 / 91.7	0.0221 / 91.7
Food security PWDM	0.1987 / 41.7	0.0273 / 83.3	0.0272 / 83.3	0.0184 / 75.0
Food security 1	0.1756 / 50.0	0.0477 / 75.0	0.0381 / 75.0	0.0163 / 75.0
Food security 2	0.1982 / 50.0	0.0348 / 91.7	0.0348 / 91.7	0.0247 / 75.0
Food security 3	0.2111 / 66.7	0.1114 / 91.7	0.1054 / 91.7	0.0638 / 83.3
Food security 4	0.2164 / 50.0	0.0346 / 75.0	0.0342 / 75.0	0.0214 / 83.3
Food security 5	0.2759 / 75.0	0.11 / 100.0	0.0307 / 100.0	0.0233 / 100.0
Pollinator abundance EWDM	0.0164 / 87.5	0.0027 / 100.0	0.0027 / 100.0	0.0 / 100.0
Pollinator abundance 1	0.0265 / 87.5	0.0133 / 100.0	0.0133 / 100.0	0.0 / 100.0
Pollinator abundance 2	0.0374 / 87.5	0.0063 / 100.0	0.0063 / 100.0	0.0 / 100.0
Pollinator abundance 3	0.0132 / 87.5	0.0097 / 87.5	0.0097 / 87.5	0.0 / 100.0
Pollinator abundance 4	0.041 / 87.5	0.0031 / 100.0	0.0031 / 100.0	0.0 / 100.0
Pollinator abundance 5	0.0391 / 100.0	0.024 / 100.0	0.024 / 100.0	0.0 / 100.0
Pollinator abundance 6	0.0405 / 87.5	0.0141 / 100.0	0.0141 / 100.0	0.0 / 100.0
Pollinator abundance 7	0.0235 / 100.0	0.012 / 100.0	0.012 / 100.0	0.0 / 100.0
Pollinator abundance 8	0.042 / 87.5	0.0218 / 87.5	0.0218 / 87.5	0.0 / 100.0
Pollinator abundance 9	0.0318 / 87.5	0.0206 / 87.5	0.0206 / 87.5	0.0 / 100.0
Pollinator abundance 10	0.0553 / 87.5	0.0046 / 100.0	0.0046 / 100.0	0.0 / 100.0

Table C.3: Results of KL-divergence and percentage of agreement performance for all InterBeta where also the middle rows are elicited, rounded to four decimals and one decimal respectively. This table contains information shown in Figure C.12.

CPT	InterBeta with elicited middle rows			
(KL-div / % agreement)	best, worst, mid	parent weights	state weights	row weights
Polar bears Ice	1.0349 / 66.7	0.6539 / 80.6	0.3214 / 69.4	0.0636 / 83.3
Polar bears Disturb	1.2857 / 75.3	1.2857 / 75.3	1.27 / 75.3	0.6246 / 82.7
Polar bears CumPop	0.7845 / 66.7	0.4828 / 63.9	0.4663 / 66.7	0.1197 / 86.1
Polar bears AFBod	1.7942 / 47.2	0.3596 / 97.2	0.1425 / 88.9	0.0584 / 91.7
Polar bears SASur	1.6724 / 58.3	0.2478 / 88.9	0.223 / 97.2	0.0337 / 100.0
Polar bears AdSur	1.6483 / 58.3	0.299 / 88.9	0.2623 / 97.2	0.0277 / 100.0
Polar bears OthMor	0.8206 / 77.8	0.7762 / 77.8	0.7429 / 77.8	0.2117 / 92.6
Polar bears EvMort	1.312 / 55.6	0.3012 / 100.0	0.3003 / 100.0	0.1189 / 96.3
Polar bears TerrPry	1.6725 / 56.2	0.0618 / 100.0	0.0524 / 100.0	0.0441 / 100.0
Polar bears Recr	1.702 / 50.0	0.4709 / 100.0	0.1893 / 100.0	0.0501 / 100.0
Polar bears Mrn	0.6355 / 75.0	0.5673 / 91.7	0.3914 / 91.7	0.0722 / 100.0
Polar bears Hab	1.3242 / 55.6	0.075 / 88.9	0.0667 / 88.9	0.0584 / 100.0
Polar bears Terr	0.496 / 91.7	0.3498 / 91.7	0.3084 / 100.0	0.1547 / 100.0
Polar bears PrimPrey	1.0344 / 66.7	0.6767 / 100.0	0.6757 / 100.0	0.3311 / 77.8
Polar bears MrnPry	1.0919 / 66.7	0.2235 / 88.9	0.2043 / 100.0	0.0347 / 100.0
Polar bears BioStr	0.8446 / 77.8	0.7387 / 77.8	0.7387 / 77.8	0.3464 / 77.8

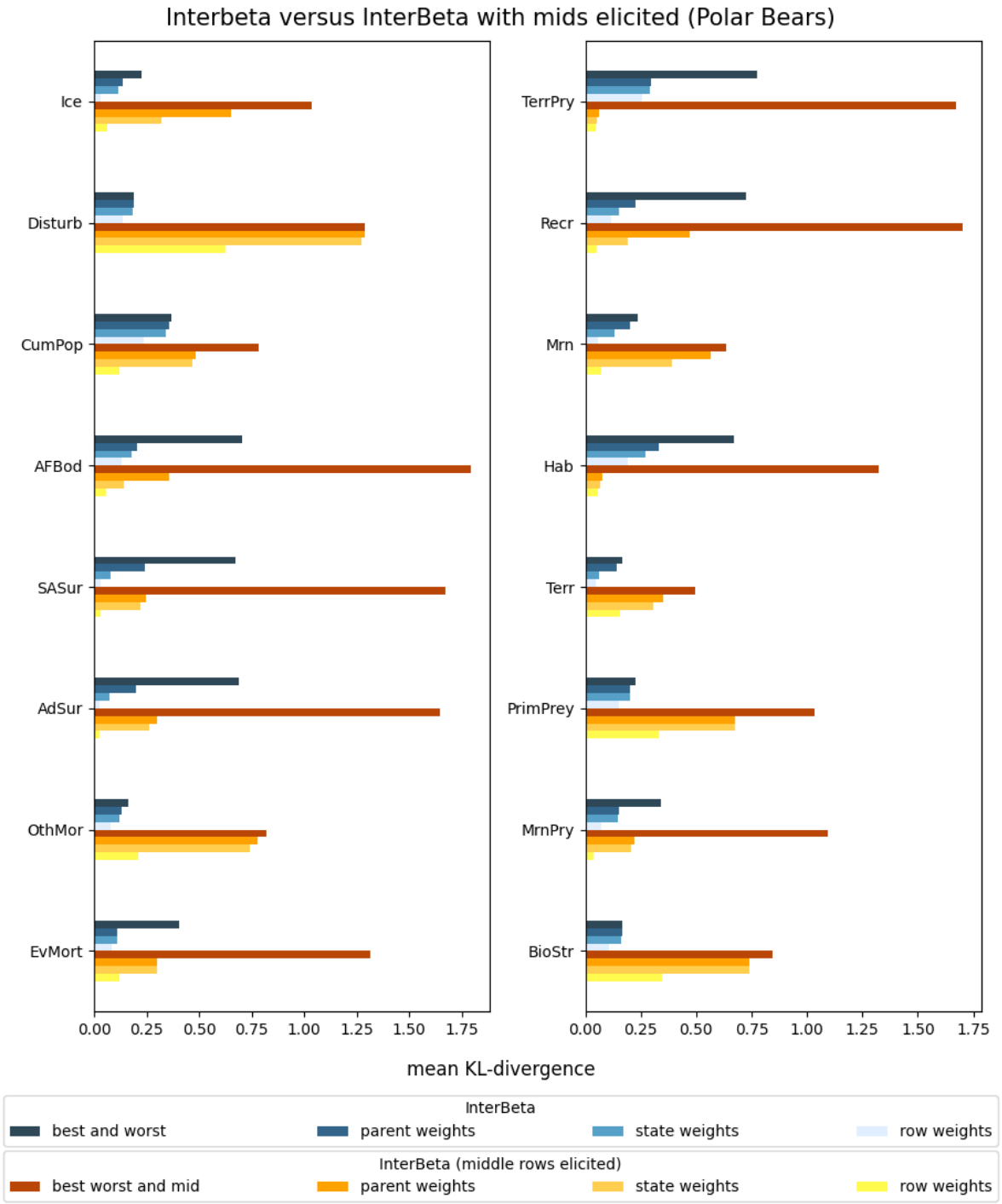


Figure C.1: Comparison of the KL-divergence of all original InterBeta versions versus all InterBeta versions where the middle rows are elicited as well.

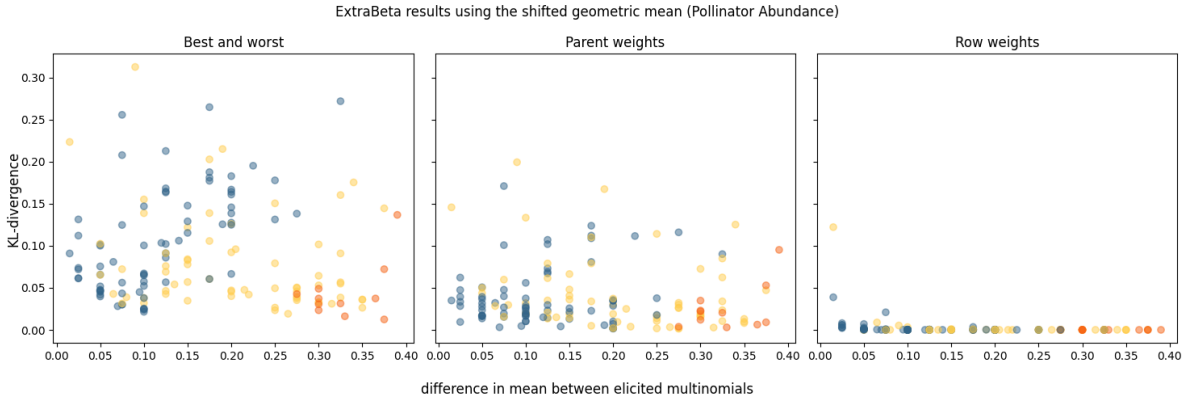


Figure C.2: Results of reconstructing Pollinator Abundance CPTs using ExtraBeta (shifted geometric, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results when either the best row or the worst row is included as input (yellow), and the results when the best and worst row are both not used (blue).

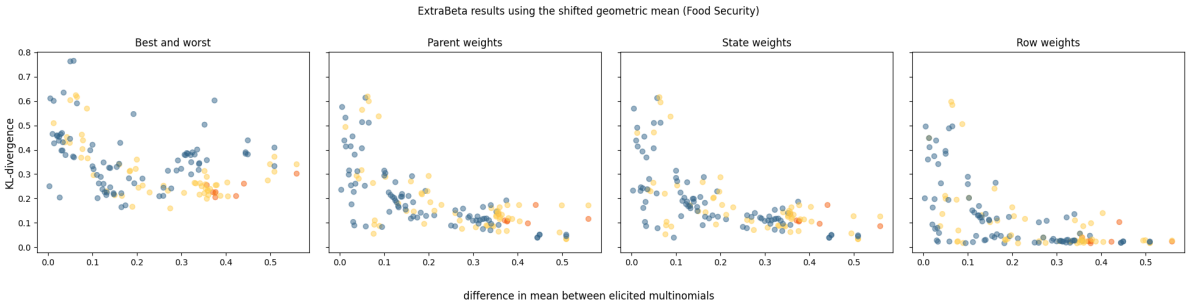


Figure C.3: Results of reconstructing Food Security CPTs using ExtraBeta (shifted geometric, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results when either the best row or the worst row is included as input (yellow), and the results when the best and worst row are both not used (blue).

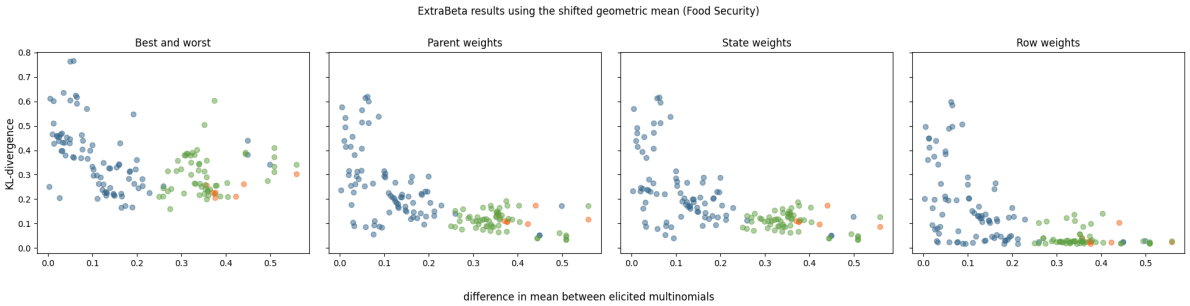


Figure C.4: Results of reconstructing Security CPTs using ExtraBeta (shifted geometric, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results when the good row has *equivalised income* in its best state and for the bad row *equivalised income* is in its worst state (green), and the results for remaining combinations (blue).

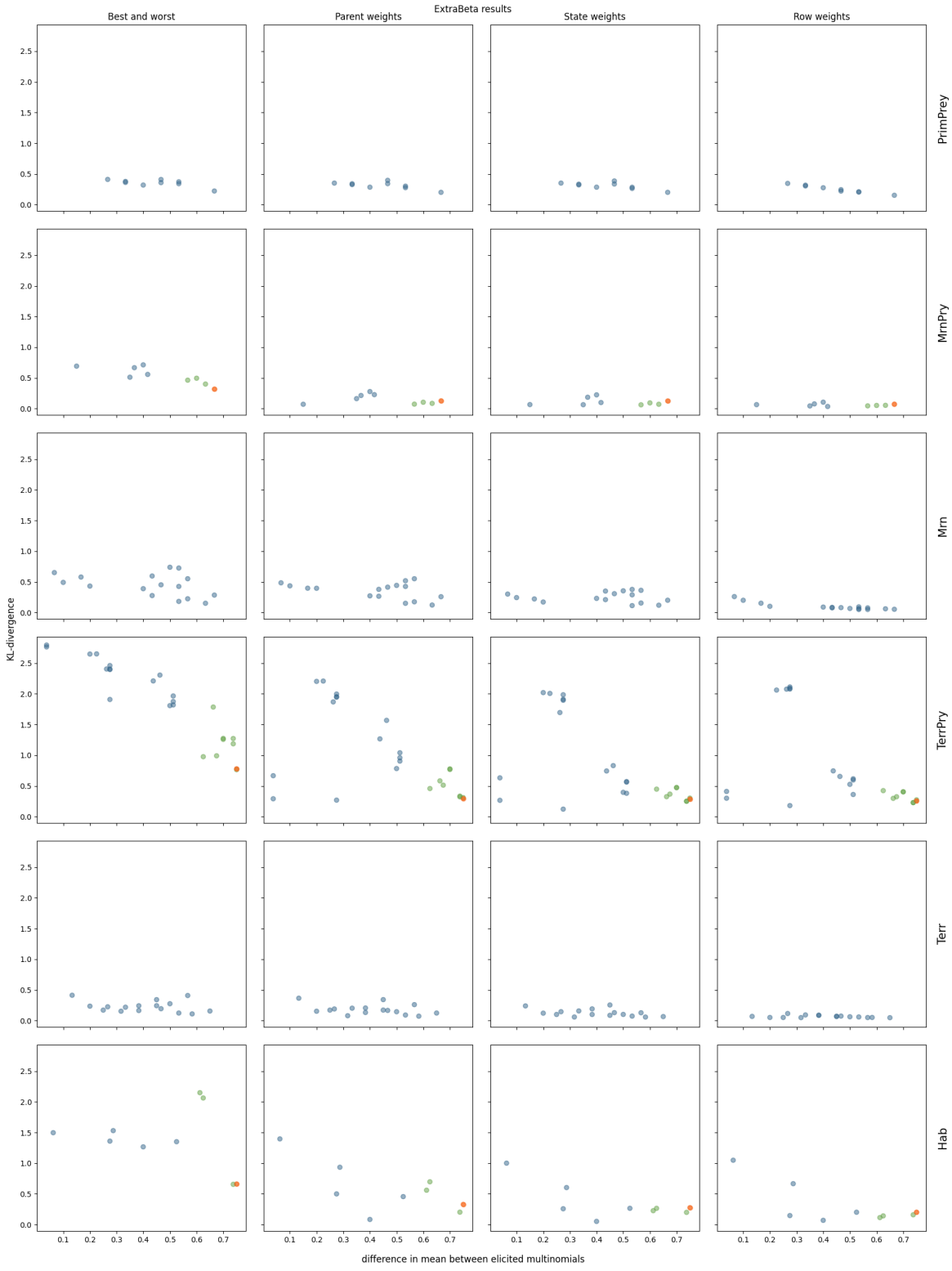


Figure C.5: Results of reconstructing Polar Bears CPTs using ExtraBeta (arithmetic, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results of the dominant parent node fixed to its best and worst state for the good and bad row respectively (green), and the results for remaining combinations (blue).

Figure C.6: Results of reconstructing Polar Bears CPTs using ExtraBeta (arithmetic, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results of the dominant parent node fixed to its best and worst state for the good and bad row respectively (green), and the results for remaining combinations (blue).

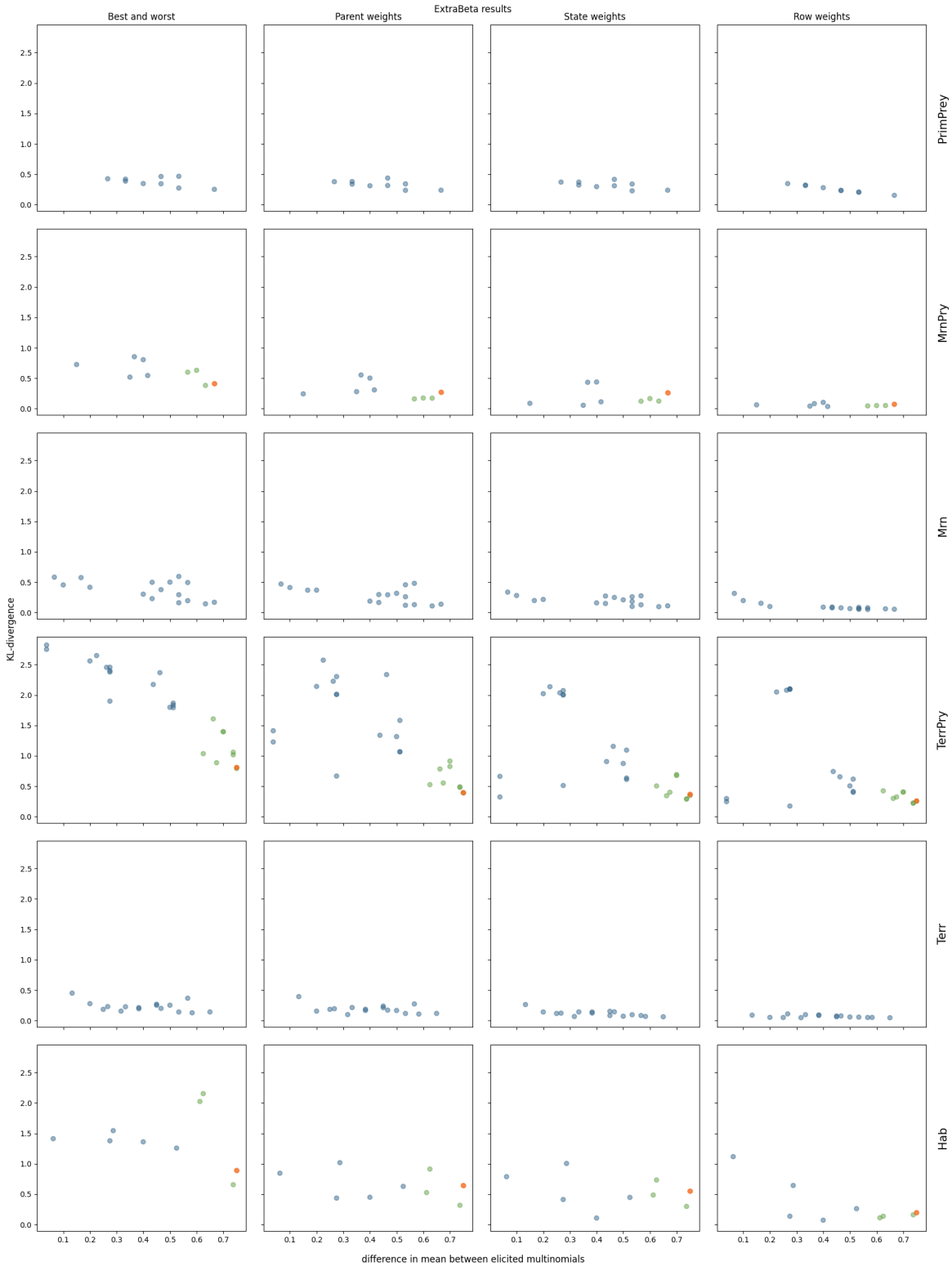


Figure C.7: Results of reconstructing Polar Bears CPTs using ExtraBeta (shifted geometric, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results of the dominant parent node fixed to its best and worst state for the good and bad row respectively (green), and the results for remaining combinations (blue).

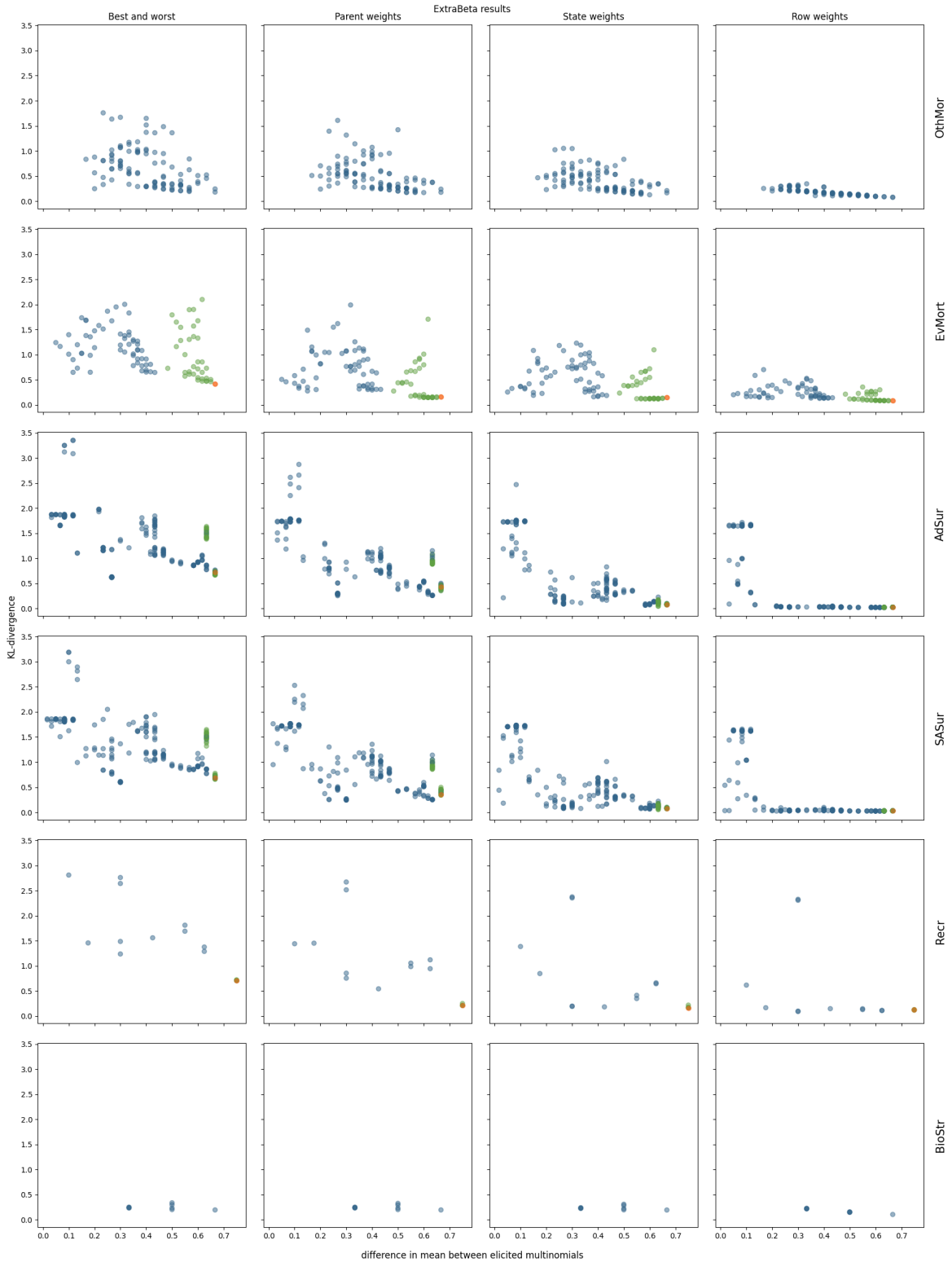
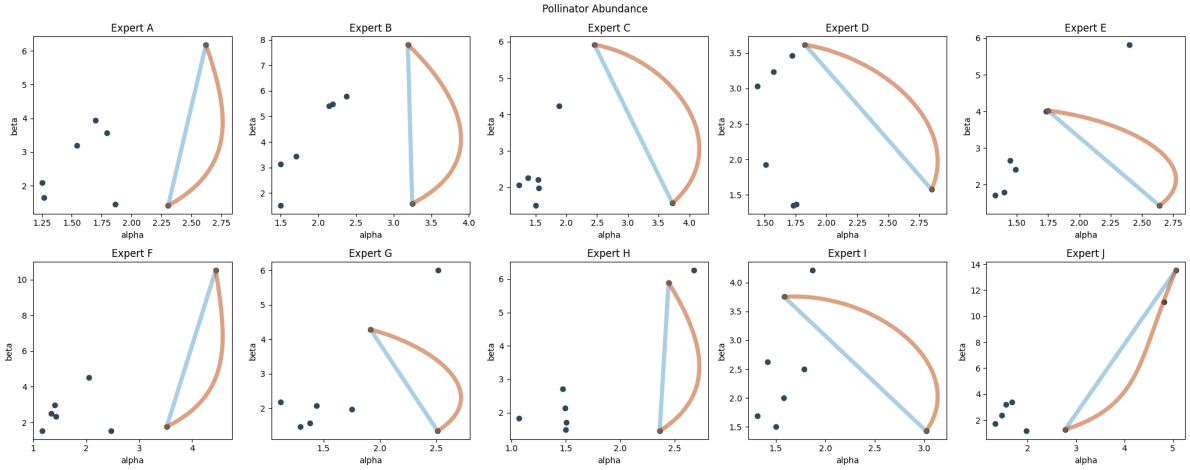
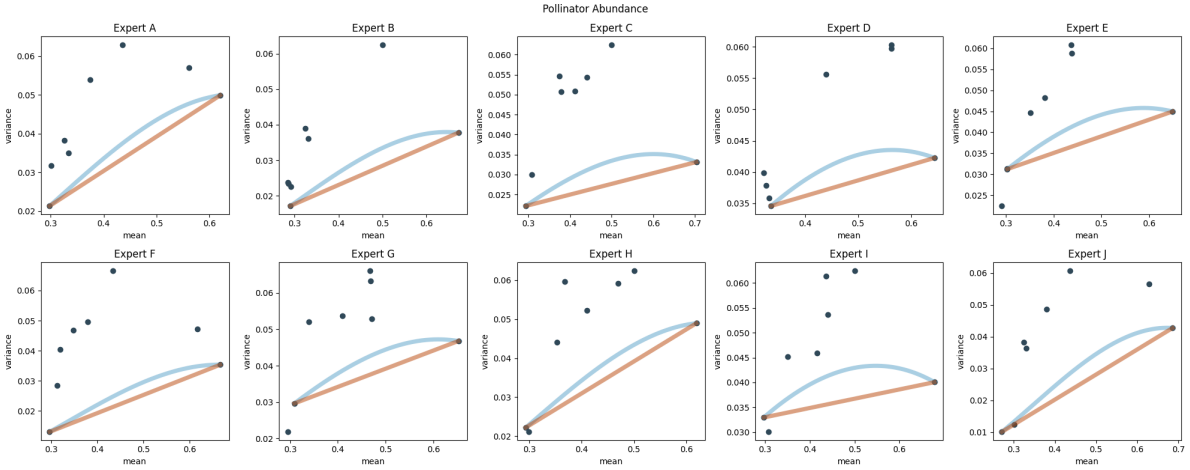


Figure C.8: Results of reconstructing Polar Bears CPTs using ExtraBeta (shifted geometric, α/β) with all potential combinations of good and bad rows as input. Including the InterBeta results (orange), the results of the dominant parent node fixed to its best and worst state for the good and bad row respectively (green), and the results for remaining combinations (blue).

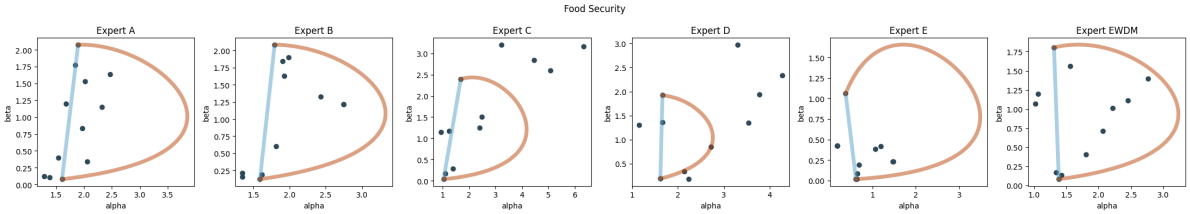


(a) Beta distribution parameters: α, β.

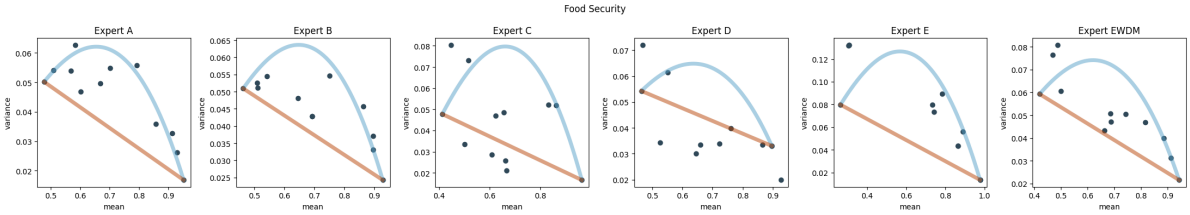


(b) Beta distribution parameters: mean and variance.

Figure C.9: Fitted Beta distribution parameters to true CPTs of the Pollinator Abundance BN, including the interpolation line for α, β (blue) and for the mean and variance (red).

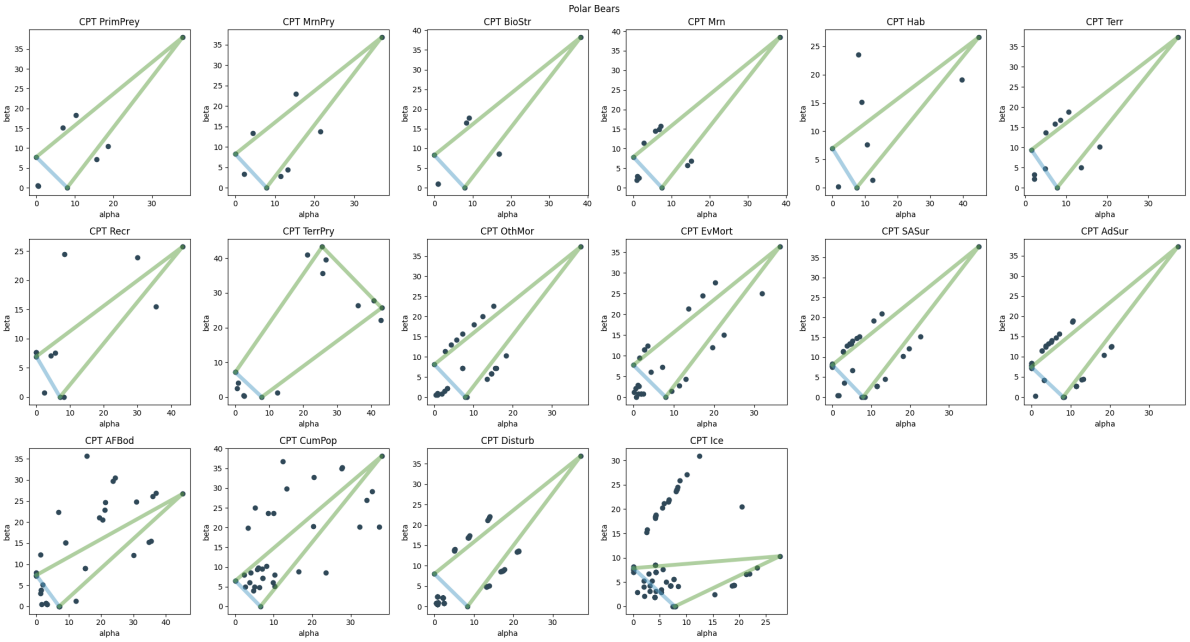


(a) Beta distribution parameters: α, β.

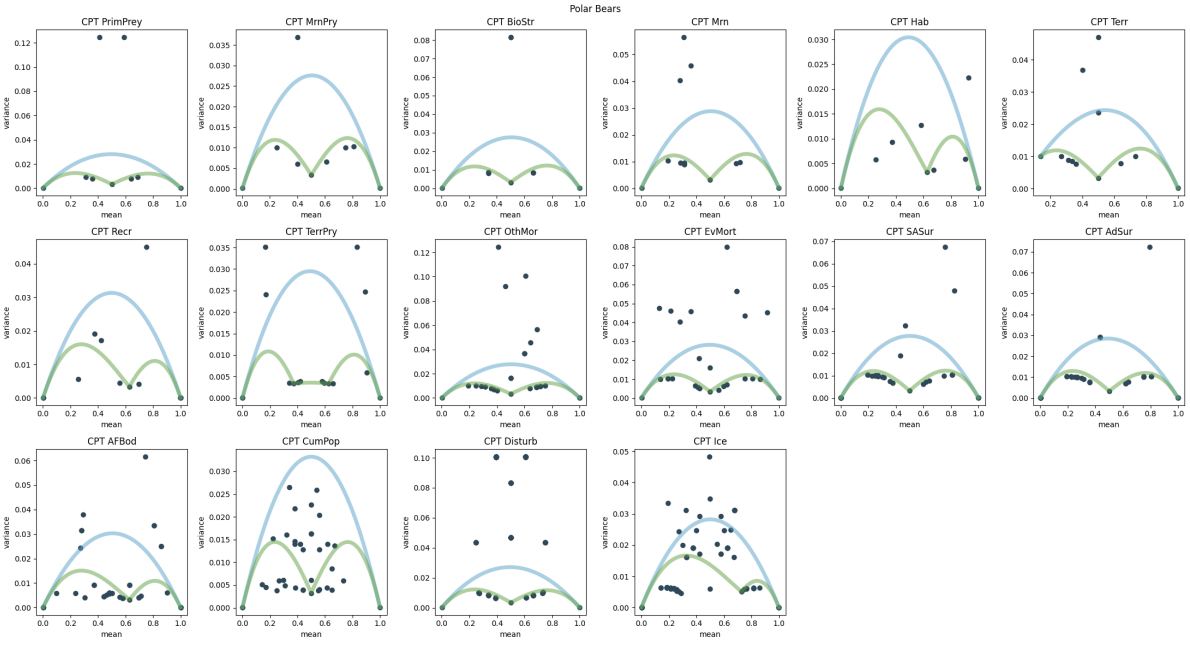


(b) Beta distribution parameters: mean and variance.

Figure C.10: Fitted Beta distribution parameters to true CPTs of the Food Security BN, including the interpolation line for α, β (blue) and for the mean and variance (red).



(a) Beta distribution parameters: α, β.



(b) Beta distribution parameters: mean and variance.

Figure C.11: Fitted Beta distribution parameters to true CPTs of the Polar Bears BN, including the interpolation line for α, β (blue) and for the mean and variance (red).

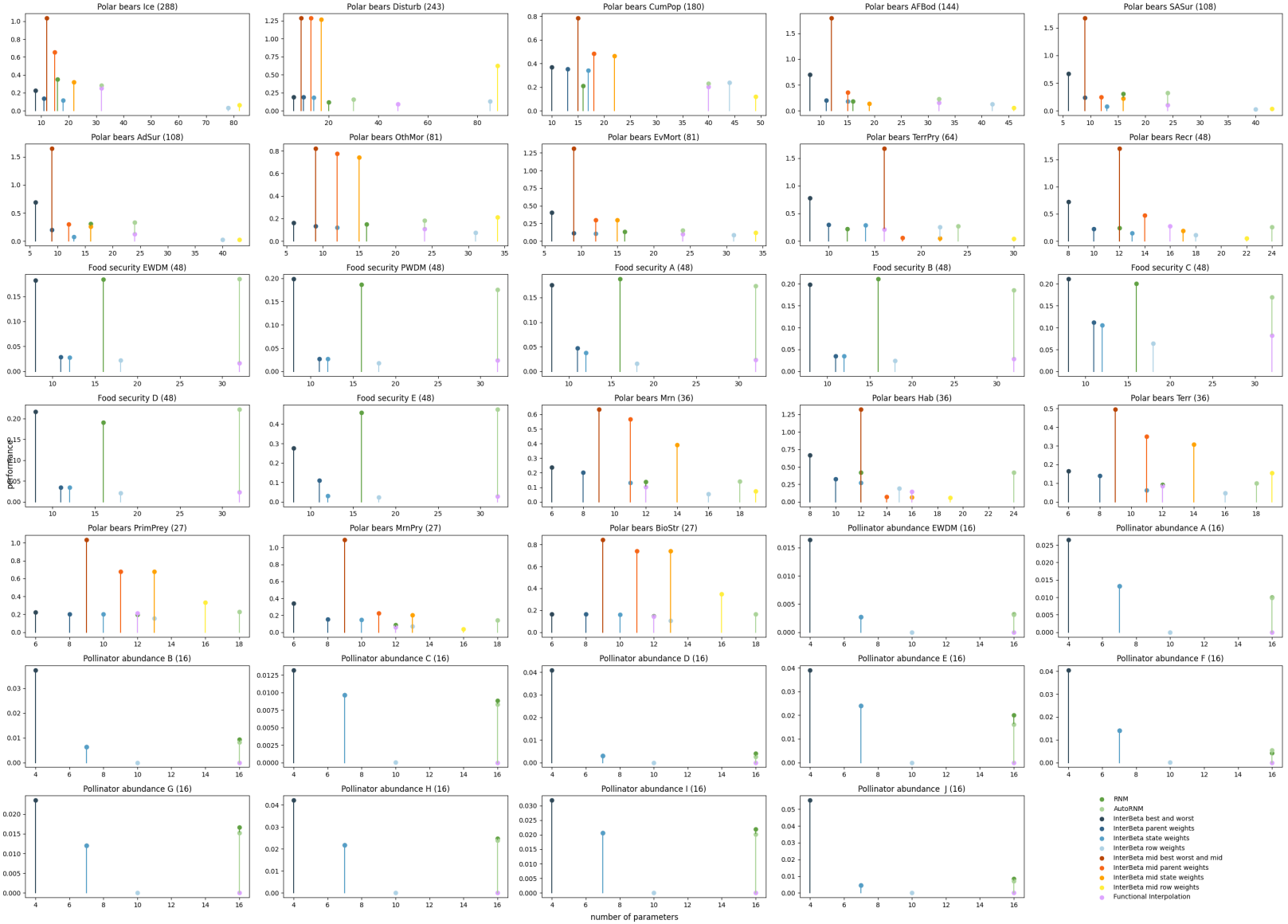


Figure C.12: Comparison of the KL-divergence between elicited CPTs and constructed CPTs by the following methods: InterBeta, RNM, and Functional Interpolation.

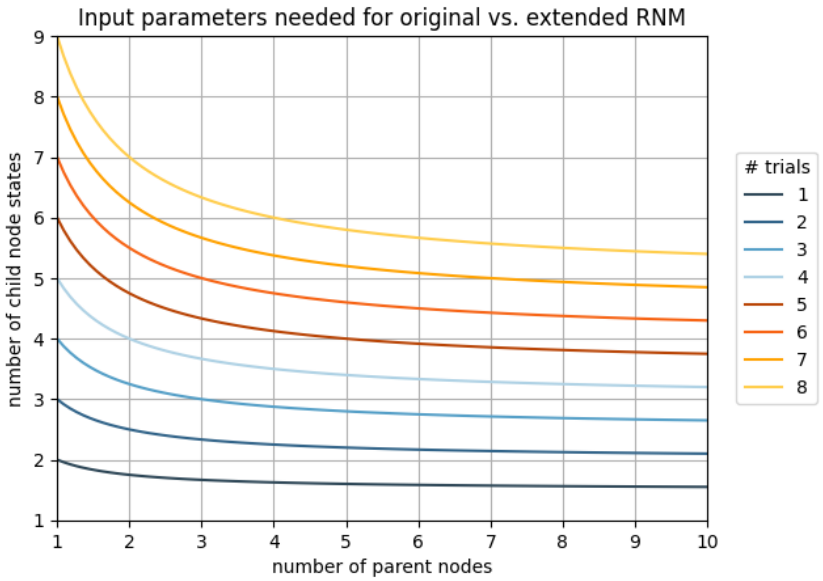


Figure C.13: Number of child node states and parent nodes in a BN such that the original RNM and extended RNM require an equal amount of input parameters.

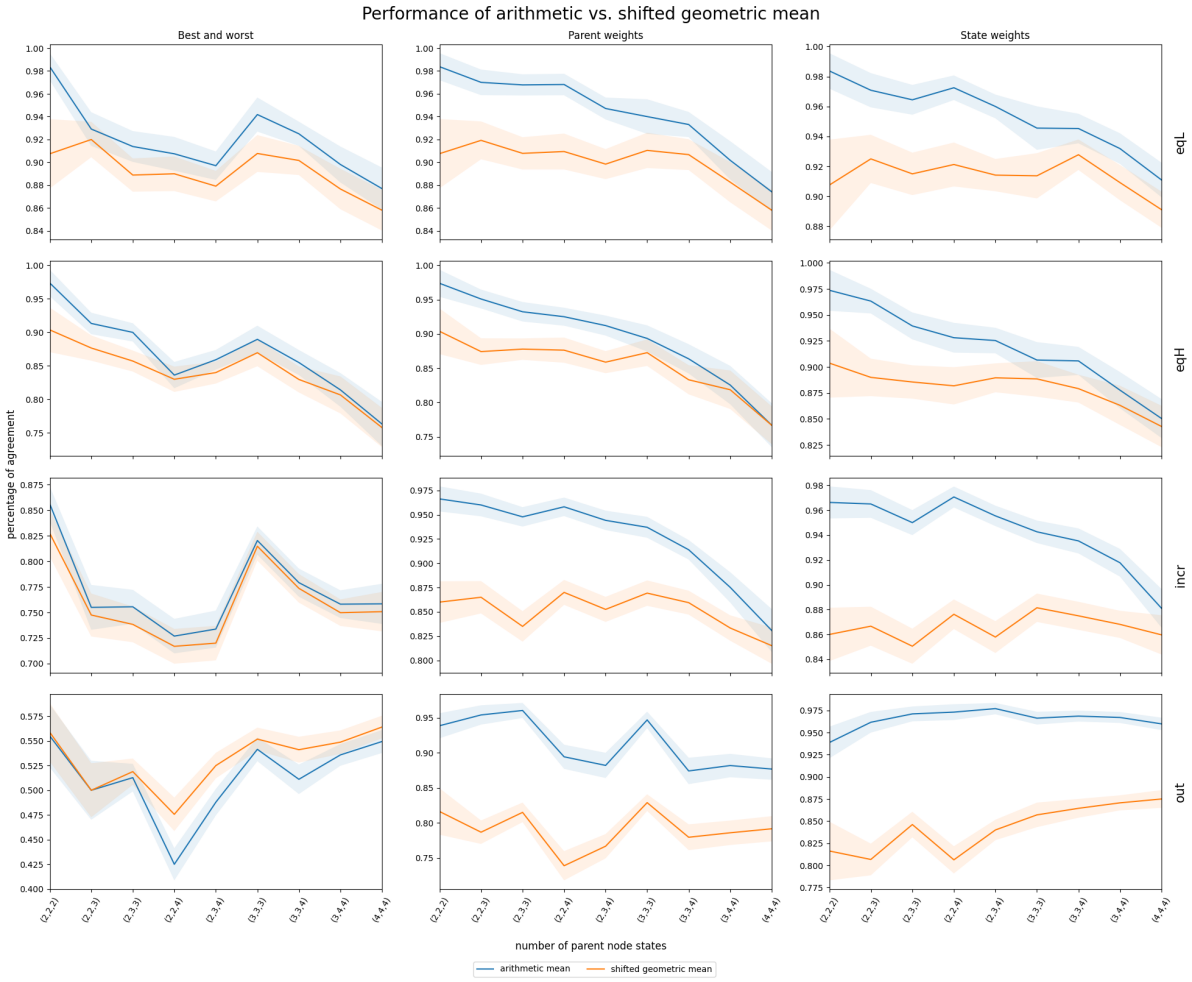


Figure C.14: Mean and 95% confidence interval of InterBeta performance (percentage of agreement), over 100 replications, on simulated data with four different correlation structures. The arithmetic and shifted geometric mean are compared, with α, β as interpolation parameters.

Figure C.15: Mean and 95% confidence interval of InterBeta performance (mean KL-divergence), over 100 replications, on simulated data with four different types of correlation structures. Individually presented for each tested combination of number of parent and child states. The arithmetic mean is used and α, β were used as interpolation parameters.

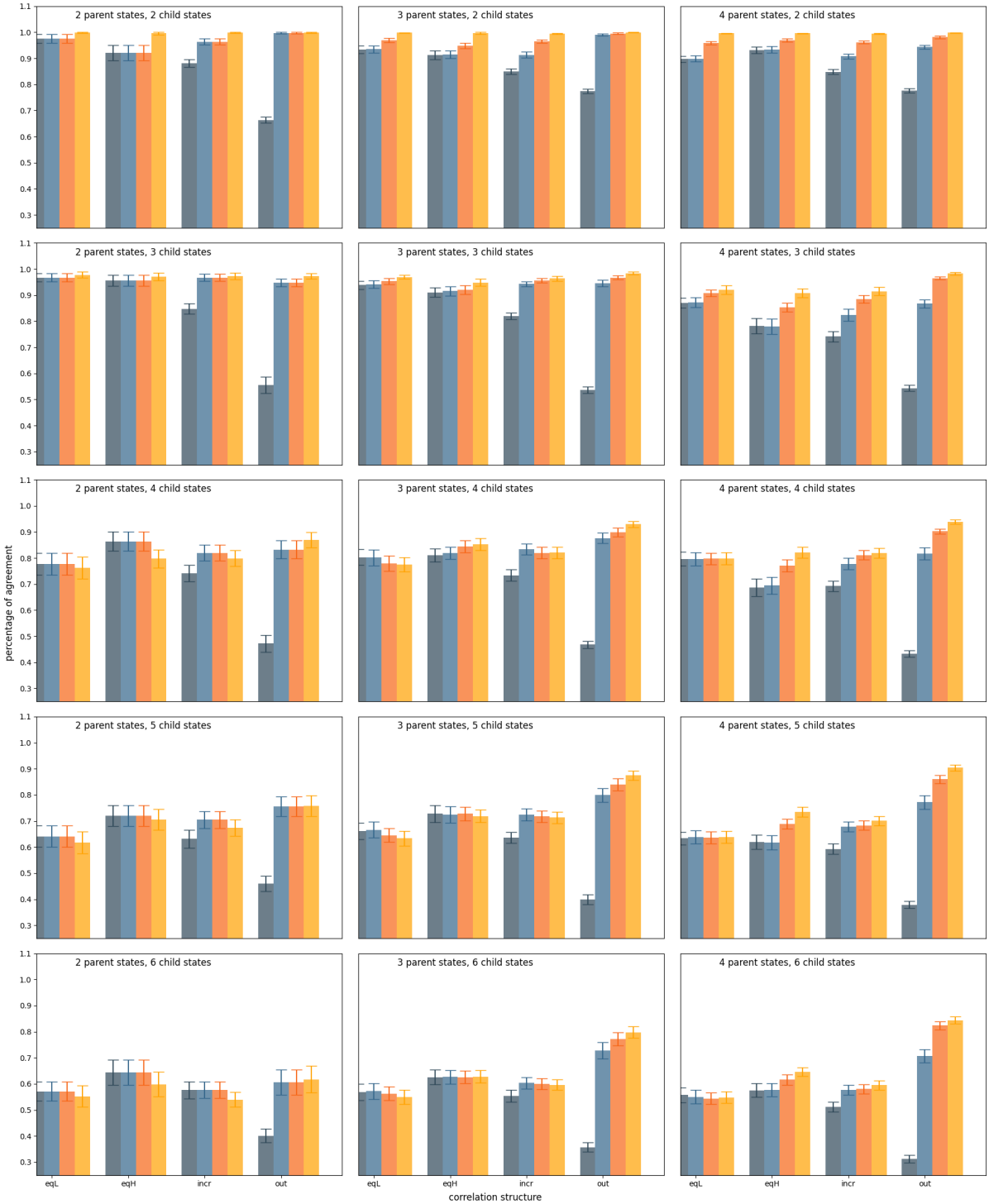


Figure C.16: Mean and 95% confidence interval of InterBeta performance (mean KL-divergence), over 100 replications, on simulated data with four different types of correlation structures. Individually presented for each tested combination of number of parent and child states. The arithmetic mean is used and α, β were used as interpolation parameters.

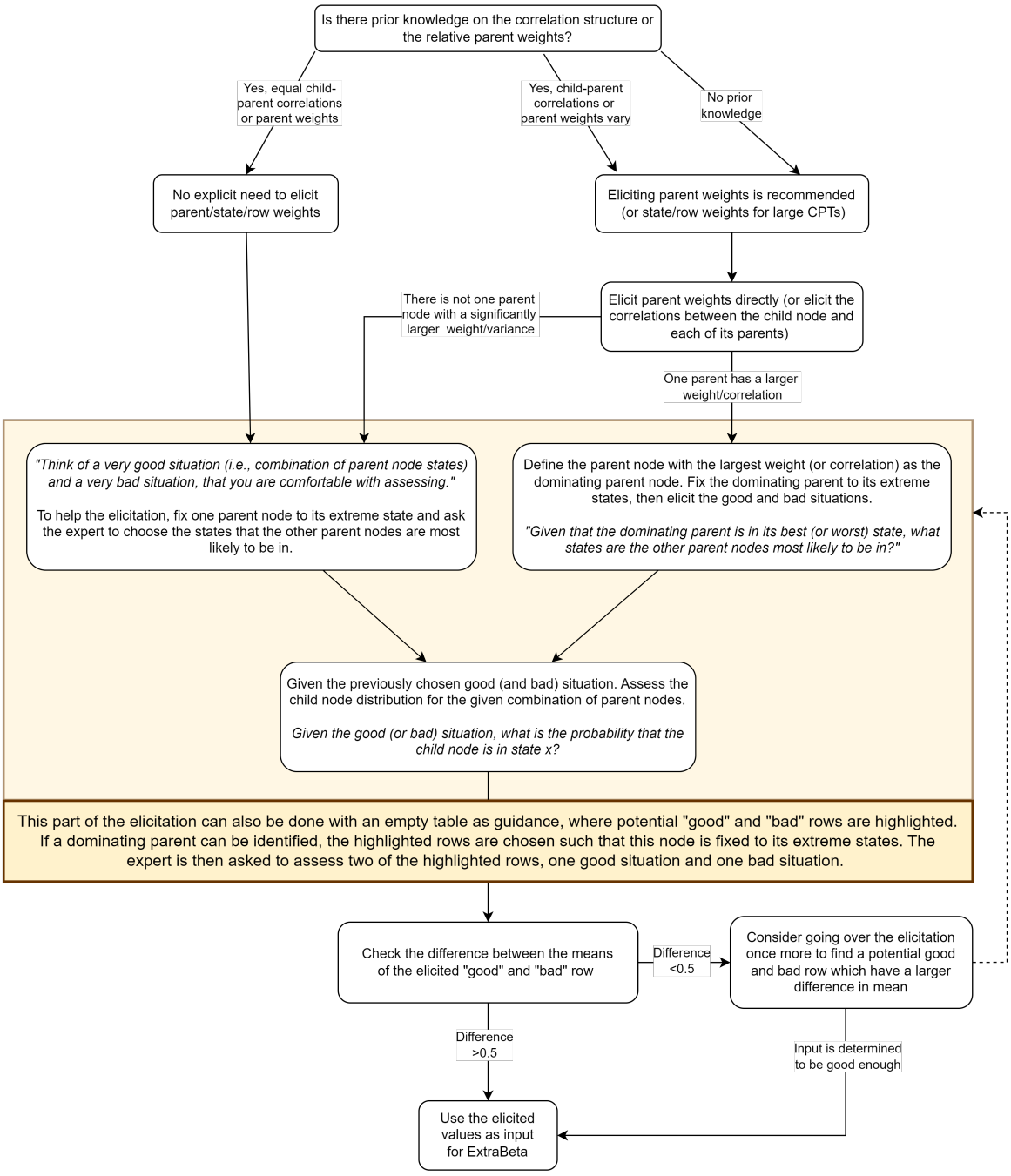


Figure C.17: Proposed elicitation protocol for ExtraBeta.